



UNIMORE

UNIVERSITÀ DEGLI STUDI DI
MODENA E REGGIO EMILIA

DIPARTIMENTO DI SCIENZE FISICHE, INFORMATICHE E MATEMATICHE

Corso di Laurea Magistrale in Informatica

Controllo semaforico adattivo e distribuito per la riduzione delle emissioni e dei tempi di attesa negli incroci urbani

*Adaptive and Distributed Traffic Signal Control for Reducing
Emissions and Waiting Times at Urban Intersections*

RELATORI:

Prof. Giacomo Cabri

Dott. Filippo Muzzini

CANDIDATO:

Simone Candiani

CORRELATRICE:

Prof.ssa Manuela Montangero

“It means your future hasn’t been written yet. No one’s has. Your future is whatever you make it. So make it a good one.”

Dr. Emmett Brown, Back to the Future Part III

Indice

1	Introduzione	3
1.1	Struttura della tesi	5
2	Presentazione della situazione esistente	7
2.1	Algoritmi Evolutivi (GA, NSGA-II, DE)	8
2.1.1	Ottimizzazione multi-obiettivo con GA standard	8
2.1.2	Modello emissivo Vehicle Specific Power (VSP)	9
2.1.3	Ottimizzazione dei tempi semaforici su rete multi-intersezione: GA vs DE	11
2.1.4	Framework bi-livello NSGA-II/MOPSO con adattamen- to real-time	12
2.2	Swarm Intelligence (PSO, ACO)	13
2.2.1	PSO avanzata con dati IoT (IS-APCPSO)	13
2.2.2	DALM con meccanismo look-ahead	15
2.3	Simulated Annealing e Programmazione Matematica	16
2.3.1	Framework integrato con previsione del traffico e Si- mulated Annealing	17
2.3.2	Simulated Bifurcation su reti urbane	18
2.4	Sistemi Fuzzy e Approcci Ibridi	19
2.4.1	Sistema fuzzy multi-agente a cinque moduli	19
2.4.2	Framework ibrido FPA + Type-2 Fuzzy + CSA	21
2.4.3	Ottimizzazione network-level con speed advisory	21
2.5	Approcci Rule-Based Deterministici	23
2.5.1	Simulazione e Design of Experiments	23
2.5.2	Controllo adattivo emission-based	24
2.5.3	SmartLight: scheduling energy-aware	25
2.5.4	Controllo cost-based per piccole città	26
2.5.5	Rule-based eco traffic lights	27

2.6	Approcci basati su Reinforcement Learning	28
2.6.1	DRL-DG: Deep RL a doppio obiettivo	29
2.6.2	Multi-Agent DQN cooperativo	30
2.7	Sintesi critica dello stato dell'arte	32
3	Requisiti e progetto	35
3.1	Il punto di partenza e il contributo della tesi	35
3.2	Requisiti	37
3.3	Limiti del controllore di riferimento	38
3.4	Progetto della funzione di costo	40
3.5	L'Adaptive Mix	43
3.6	Ipotesi del modello e sensori richiesti	45
3.6.1	Dati statici e di stato	45
3.6.2	Dati operativi e sensori	45
3.6.3	Tecnologie sensoristiche realizzabili	47
4	Implementazione	49
4.1	Strumenti di simulazione e modellazione emissiva	49
4.2	Architettura generale e integrazione con SUMO	51
4.3	Identificazione di fasi e flussi	52
4.4	Pre-calcolo statico delle informazioni di rete	54
4.5	Il ciclo di decisione	55
4.6	Il calcolo dei costi	56
4.7	Approssimazioni e scelte tecniche	57
4.8	Generazione delle popolazioni veicolari	58
4.9	Automazione degli esperimenti	59
5	Esperimenti	61
5.1	Setup sperimentale	61
5.2	Risultati su Bologna	64
5.3	Risultati su Manhattan 15×15	65
5.4	Risultati su Manhattan 8×8	66
5.5	Risultati su Masa	67
5.6	Deployment parziale	69

5.7	Scomposizione del tempo di viaggio rispetto al tempo ideale . .	73
6	Conclusioni	77
6.1	Sintesi del lavoro e dei risultati	78
6.2	Sviluppi futuri	80
7	Ringraziamenti	83
	Bibliografia	93

Elenco delle figure

3.1	Flusso decisionale complessivo del controllore semaforico adattivo e distribuito proposto.	47
5.1	Rete stradale di Bologna utilizzata per la valutazione sperimentale nel simulatore SUMO.	63
5.2	Variazione di CO ₂ rispetto al piano fisso, per ciascuna rete e densità.	69
5.3	Variazione di CO ₂ rispetto al piano fisso, con deployment parziale.	72
5.4	Bologna vph 4 470: scomposizione del tempo medio di viaggio in tre componenti.	74

Elenco degli algoritmi

1	Ricostruzione dei flussi associati alle fasi principali del piano semaforico	53
2	Implementazione della regola dello <code>STOP_COST</code> definita nel- l'Eq. (4.1).	56
3	Implementazione della funzione di costo proposta nell'Eq. (3.1).	57
4	Implementazione dell'Adaptive Mix definito nell'Eq. (3.5). . .	57

Elenco delle tabelle

5.1	Caratteristiche strutturali delle reti analizzate.	62
5.2	Bologna — nuovo algoritmo rispetto al controllore di riferimento. Valori percentuali: negativo indica miglioramento. . .	64
5.3	Bologna — nuovo algoritmo rispetto al piano a tempi fissi (Fisso). Valori percentuali: negativo indica miglioramento. . .	64
5.4	Manhattan 15 x 15 — nuovo algoritmo rispetto al controllore di riferimento.	65
5.5	Manhattan 15 x 15 — nuovo algoritmo rispetto al piano a tempi fissi (Fisso).	65
5.6	Manhattan 8 x 8 — nuovo algoritmo rispetto al controllore di riferimento.	66
5.7	Manhattan 8 x 8 — nuovo algoritmo rispetto al piano a tempi fissi (Fisso).	67
5.8	Masa — nuovo algoritmo rispetto al controllore di riferimento.	67
5.9	Masa — nuovo algoritmo rispetto al piano a tempi fissi (Fisso).	68
5.10	Bologna — nuovo algoritmo rispetto al piano a tempi fissi, al variare della frazione di incroci equipaggiati.	70
5.11	Manhattan 15 x 15 — nuovo algoritmo rispetto al piano a tempi fissi, al variare della frazione di incroci equipaggiati. . .	70
5.12	Bologna 4470 — scomposizione del tempo medio di viaggio per veicolo nelle tre componenti.	74

Introduzione

Il costante aumento del traffico veicolare nelle aree urbane avvenuto negli ultimi decenni, in ogni grande città del mondo, ha portato la mobilità sostenibile a diventare una delle tematiche più **sensibili** e **importanti** nell'era della transizione ecologica. Con le continue spinte normative verso la riduzione delle emissioni di anidride carbonica e l'inasprimento dei limiti europei sulle emissioni del settore dei trasporti, si cerca di rendere il più possibile efficiente ogni elemento del sistema di mobilità urbana, compresi i semafori che regolano il flusso negli incroci delle città. Proprio per questo negli ultimi anni sono riemersi con maggiore interesse rispetto al passato gli *algoritmi adattivi per il controllo semaforico*, una soluzione composta da diverse componenti software e hardware che cercano di risolvere alcune delle problematiche esposte.

Che cosa si intende con *algoritmo adattivo*? Come detto precedentemente, si parla di una soluzione composta, che combina la rilevazione del traffico in tempo reale con la gestione dinamica delle fasi semaforiche. Accogliendo questa tecnologia, le città potranno raccogliere e analizzare dati provenienti dai sensori installati sulle strade, ciò permetterà di avere un controllo dinamico del flusso veicolare attraverso decisioni automatiche che ottimizzeranno il consumo di carburante e il tempo di attesa dei conducenti. Tenendo conto di quanto detto in precedenza, viene ora introdotto il problema di ricerca affrontato nella presente tesi, esaminando nel dettaglio le motivazioni e gli obiettivi per cui è nato questo progetto.

Il progetto è stato svolto in ambito universitario, dove abbiamo intrapreso un'ambiziosa iniziativa finalizzata allo sviluppo di un nuovo algoritmo adattivo per il controllo semaforico, sotto la supervisione del professore Giacomo Cabri, della professoressa Manuela Montangero e del ricercatore universitario

Filippo Muzzini Ph.D. dell'Università di Modena e Reggio Emilia.

Nel caso considerato, il lavoro si è concentrato sulla progettazione di un algoritmo originale che integra al suo interno una logica preesistente sviluppata nel gruppo di ricerca, combinandola con dei nuovi componenti pensati per affrontare aspetti del traffico urbano non considerati in precedenza. L'obiettivo del lavoro consisteva nella progettazione di un nuovo algoritmo adattivo, costruito attorno a una formula di costo originale che tiene conto della domanda di traffico in ingresso, della congestione delle strade a valle dell'incrocio e dello stato di saturazione delle corsie locali. All'interno di questa nuova soluzione, l'algoritmo proposto nel progetto preesistente è stato integrato come una delle due componenti di un *Mix Adattivo*: un meccanismo che bilancia dinamicamente la logica basata sulla velocità dei veicoli e la nuova logica basata sull'occupazione delle corsie e sulla congestione, in base allo stato del traffico osservato istantaneamente. Oltre alla progettazione dell'algoritmo, il lavoro ha previsto la validazione delle sue prestazioni su reti urbane realistiche e l'analisi della possibilità di un *deployment parziale* nel quale solo una frazione dei semafori della città viene equipaggiata con la nuova logica. Per raggiungere tali obiettivi, è stata condotta un'attenta analisi delle tecnologie e dei modelli matematici alla base del controllo semaforico adattivo; approfondendo ogni dettaglio per garantire una corretta implementazione, in modo da poter sfruttare l'algoritmo nel vero obiettivo, cioè ridurre le emissioni di CO₂ e i tempi di attesa nelle reti urbane reali, allo scopo di proporre una soluzione realmente applicabile dalle amministrazioni cittadine. L'analisi e lo studio dell'algoritmo saranno affiancati dall'analisi del simulatore di traffico **SUMO** e dei modelli di emissione veicolare (**HBEFA4**) utilizzati per validare i risultati ottenuti su quattro diverse reti stradali, tra cui la mappa di **Bologna**.

1.1 Struttura della tesi

Per quanto riguarda la struttura della tesi, il **Capitolo 2** presenta lo stato dell'arte sul controllo semaforico adattivo, partendo dai sistemi a *tempo fisso* fino ad arrivare agli approcci più recenti basati su *Reinforcement Learning*. Il **Capitolo 3** definisce i requisiti del nuovo algoritmo e ne presenta il progetto, con la formulazione matematica del *Mix Adattivo*, le tre componenti della funzione di costo (*domanda*, *penalità downstream* e *fattore di congestione locale*), il meccanismo di decisione di cambio fase e le ipotesi del modello, inclusi i sensori richiesti per la sua applicabilità reale. Il **Capitolo 4** descrive l'implementazione dell'algoritmo in Python e la sua integrazione con il simulatore **SUMO** tramite l'interfaccia *TraCI*, la generazione delle popolazioni veicolari secondo le distribuzioni del parco circolante italiano (**dati ACI 2022**) e gli script di automazione utilizzati per condurre gli sweep sperimentali. Il **Capitolo 5** presenta i risultati degli esperimenti principali: il confronto tra le *diverse condizioni di densità* e lo *sweep sul deployment parziale* dei semafori smart, che rappresenta il contributo più originale del lavoro. Infine, il **Capitolo 6** contiene le **conclusioni**, dove vengono sintetizzati i contributi della tesi, discussi i risultati alla luce della letteratura, identificati i limiti del modello attuale e proposte le direzioni di sviluppo futuro. In fondo all'elaborato è presente la **bibliografia** con tutti i riferimenti citati nel corso del lavoro.

Presentazione della situazione esistente

In questo capitolo si ricostruisce lo stato dell'arte attuale dei principali metodi di controllo semaforico orientati alla sostenibilità ambientale. Negli ultimi anni, si è avvertita la necessità di ridurre l'impatto del traffico urbano in termini di congestione, consumo energetico ed emissioni. Tutto ciò ha favorito lo sviluppo di numerosi approcci computazionali per l'ottimizzazione dei tempi semaforici.

La letteratura degli ultimi anni propone tecniche tra loro eterogenee: metaeuristiche evolutive, algoritmi di swarm intelligence, metodi di ottimizzazione classica, sistemi fuzzy, strategie deterministiche rule-based e approcci fondati su apprendimento automatico e reinforcement learning. Pur differendo per complessità, ipotesi e requisiti infrastrutturali, tali metodi condividono un obiettivo comune: migliorare il funzionamento del sistema di traffico riducendo al tempo stesso gli effetti ambientali associati alla circolazione veicolare.

Per rendere il confronto più leggibile, i contributi analizzati sono stati organizzati secondo la propria famiglia metodologica di appartenenza. In particolare, vengono esaminati:

- algoritmi evolutivi (Genetic Algorithm, NSGA-II, Differential Evolution);
- tecniche di swarm intelligence (Particle Swarm Optimization, Ant Colony Optimization);
- metodi di ottimizzazione classica come Simulated Annealing e Simulated Bifurcation;
- sistemi fuzzy e approcci ibridi;

- strategie rule-based deterministiche;
- approcci basati su apprendimento automatico e reinforcement learning.

Per ciascun contributo vengono messi in evidenza gli aspetti più significativi per poterli usare in un confronto finale, in particolare la formulazione del problema, la funzione obiettivo, l'algoritmo utilizzato, l'ambiente di simulazione e le metriche di valutazione considerate.

2.1 Algoritmi Evolutivi (GA, NSGA-II, DE)

Gli algoritmi evolutivi rappresentano una classe di metaeuristiche che sono ispirate ai meccanismi della selezione naturale. In questi tipi di approcci, una popolazione di soluzioni candidate evolve progressivamente attraverso l'utilizzo di operatori di selezione, crossover e mutazione, con l'obiettivo di individuare configurazioni sempre più performanti nello spazio delle soluzioni.

Ogni individuo della popolazione codifica tipicamente una configurazione dei tempi delle fasi semaforiche, includendo la durata delle fasi di verde, giallo e rosso per le diverse direzioni di traffico. La qualità di ciascuna configurazione viene valutata attraverso simulazioni microscopiche del traffico, che permettono di stimare e valutare le prestazioni del sistema in termini di efficienza operativa e impatto ambientale.

Di seguito vengono presentati quattro contributi recenti che illustrano la versatilità e l'efficacia di questa famiglia nel contesto del controllo semaforico eco-sostenibile.

2.1.1 Ottimizzazione multi-obiettivo con GA standard

[HacS24] propongono l'applicazione di un Genetic Algorithm (GA) multi-obiettivo per l'ottimizzazione dei tempi semaforici di un'intersezione isolata, con particolare attenzione alla riduzione simultanea del tempo di attesa totale dei veicoli, delle emissioni di CO₂ e del consumo di carburante.

Ogni individuo della popolazione va a codificare una configurazione dei parametri temporali delle fasi semaforiche, attraverso un processo evolutivo

che segue la struttura classica del GA: inizializzazione casuale della popolazione, valutazione della fitness tramite simulazione in SUMO, selezione tramite Tournament Selection, crossover (con probabilità pari a 0.7) e mutazione con bassa probabilità (0.01).

La valutazione delle soluzioni candidate è basata su una funzione di fitness aggregata definita come segue:

$$\text{Fitness} = \frac{1}{total_value} \quad (2.1)$$

In particolare, *total_value* è definita come una misura aggregata che combina tempo di attesa, emissioni di CO₂ e consumo di carburante. Questo permette di trattare il problema come un unico obiettivo composito.

La scelta del Genetic Algorithm viene motivata anche attraverso un confronto qualitativo con altre metaeuristiche, tra cui Particle Swarm Optimization (PSO) e Ant Colony Optimization (ACO). In questo confronto, il GA mostra una buona robustezza nel contesto multi-obiettivo. Va però sottolineato che si tratta di un confronto più argomentativo che sperimentale, e non di una valutazione diretta tra i tre algoritmi.

2.1.2 Modello emissivo Vehicle Specific Power (VSP)

Nel lavoro presentato in [WanZZ⁺24] viene proposto un metodo di ottimizzazione multi-obiettivo per il controllo semaforico di una singola intersezione, dove la componente ambientale è modellata esplicitamente attraverso il *Vehicle Specific Power* (VSP). Il contributo del paper non consiste quindi soltanto nell'introduzione di un modello emissivo più dettagliato, ma anche nella costruzione di un framework che va a considerare congiuntamente ritardo, lunghezza delle code ed emissioni all'interno della procedura di ottimizzazione.

A differenza di molti altri approcci presenti in letteratura, che trattano l'impatto ambientale in modo indiretto attraverso proxy come tempo di attesa, velocità media o numero di fermate, il VSP va a rappresentare la potenza specifica richiesta dal veicolo e cattura l'effetto combinato di velocità e accele-

razione. Il paper richiama inoltre come le diverse condizioni di guida tipiche delle intersezioni semaforiche: decelerazione, arresto, accelerazione e moto uniforme – vadano ad incidere in modo diverso sulle emissioni, motivando così la necessità di adottare un modello emissivo esplicito; in quest’ottica, il VSP risulta particolarmente adatto a descrivere le dinamiche dei flussi in prossimità dell’intersezione.

I valori di VSP vengono classificati in diversi intervalli discreti di potenza specifica, ai quali vengono associati tassi di emissione per i diversi inquinanti. L’emissione del singolo veicolo viene stimata combinando i tassi emissivi con la distribuzione dei valori di VSP presi in esame, mentre le emissioni complessive dell’intersezione analizzata vengono ottenute aggregando i contributi dei diversi rami di accesso e distinguendo tra veicoli in attraversamento e veicoli che sono in attesa; così la valutazione ambientale viene integrata direttamente nella funzione obiettivo del modello.

Dal punto di vista dell’ottimizzazione, viene costruita una formulazione multi-obiettivo che punta a minimizzare simultaneamente ritardo medio, lunghezza media della coda ed emissioni totali, trasformando successivamente i tre rapporti normalizzati in una funzione composita. Per risolvere il problema viene adottato un *improved genetic algorithm*: rispetto a un GA tradizionale, il miglior individuo di ogni generazione viene preservato direttamente come offspring, così da evitare il peggioramento della fitness migliore già raggiunta, mentre le probabilità di crossover e mutazione vengono adattate dinamicamente durante il processo evolutivo.

La parte sperimentale viene condotta in VISSIM su un caso reale di singola intersezione, all’interno di un framework di microsimulazione con rolling optimization ogni 10 minuti. I risultati mostrano che, rispetto allo schema non ottimizzato, il modello proposto riduce il ritardo totale del 21%, la lunghezza complessiva delle code del 23% e le emissioni totali di inquinanti del 15%. Il lavoro evidenzia quindi come l’integrazione di un modello emissivo esplicito basato su VSP all’interno di una procedura di ottimizzazione possa migliorare in modo congiunto sia le prestazioni del traffico sia la valutazione dell’impatto ambientale.

2.1.3 Ottimizzazione dei tempi semaforici su rete multi-intersezione: GA vs DE

In [Fai25] viene affrontato il problema dell'ottimizzazione dei tempi semaforici su una rete composta da 16 intersezioni, per un totale di 64 semafori controllati simultaneamente. L'obiettivo dello studio è confrontare due metaeuristiche evolutive, Genetic Algorithm (GA) e Differential Evolution (DE). Il confronto è stato svolto all'interno di un'applicazione sviluppata dagli autori per la gestione dei tempi di rosso e di verde.

Ogni soluzione candidata codifica i tempi semaforici della rete, inizializzati casualmente entro intervalli prefissati. La formulazione adottata nel paper si basa su una regola di congestione locale: in particolare se una corsia in ingresso raggiunge almeno il 50% del totale dei veicoli presenti nell'intersezione, la situazione viene considerata congestionata e questo comporta l'estensione della durata del verde mediante un fattore di aggiustamento.

La Differential Evolution viene implementata nella forma classica *DE/rand/1*, dove il vettore mutante è generato tramite combinazione vettoriale:

$$v_i = x_{r1} + F \cdot (x_{r2} - x_{r3}) \quad (2.2)$$

dove x_{r1} , x_{r2} e x_{r3} sono individui selezionati casualmente e F è il fattore di scala. Dopo aver concluso la fase di crossover, la selezione viene svolta in modo deterministico: il trial vector sostituisce il vettore corrente solo se produce un valore di fitness non peggiore.

Il confronto finale riportato dagli autori mostra che, nel caso della rete considerata, la DE produce soluzioni migliori del GA sia nell'ottimizzazione dei tempi di verde sia nella minimizzazione dei tempi di rosso. Il lavoro più che costituire uno studio formale di scalabilità, evidenzia come, in questo scenario multi-intersezione, la scelta della metaeuristica possa influire sensibilmente sulla qualità delle soluzioni semaforiche ottenute.

2.1.4 Framework bi-livello NSGA-II/MOPSO con adattamento real-time

In [ShaZWL24] viene proposto un framework avanzato per il controllo semaforico dinamico che affronta uno dei limiti principali dei metodi tradizionali: la difficoltà nel gestire in modo efficace la natura multi-criterio e fortemente variabile del traffico urbano. L'architettura è organizzata su due livelli.

Il primo livello esegue un'ottimizzazione multi-obiettivo offline mediante MOPSO e l'algoritmo ibrido NSGAII-MOPSO. NSGA-II ha il compito di generare soluzioni Pareto-ottimali attraverso ordinamento non dominato e crowding distance, mantenendo diversità lungo il fronte Pareto, mentre MOPSO aggiorna le posizioni delle particelle secondo:

$$v_i(t) = w \cdot v_i(t-1) + c_1 r_1 (pbest_i - x_i) + c_2 r_2 (gbest_i - x_i) \quad (2.3)$$

$$x_i(t) = x_i(t-1) + v_i(t) \quad (2.4)$$

I tre obiettivi sono il ritardo medio $d = \sum_i d_i Q_i / \sum_i Q_i$, il numero medio di fermate $h = \sum_i h_i Q_i / \sum_i Q_i$ e il consumo di carburante $E = f_1 L_s + f_2 d + f_3 h$, dove L_s è la distanza percorsa e f_1, f_2, f_3 sono coefficienti dipendenti dalle caratteristiche del veicolo.

Il secondo livello introduce l'adattamento dinamico: a partire dal timing base ottenuto nel primo livello, gli autori generano in simulazione una corrispondenza tra lunghezze di coda e tempi ottimali di estensione del verde, quindi utilizzano una rete neurale fuzzy ANFIS per apprendere tali regole. Le regole fuzzy sono di tipo Takagi–Sugeno:

$$\text{if } x \text{ is } A_i \text{ and } y \text{ is } B_i \text{ then } f_i = p_i x + q_i y + r_i \quad (2.5)$$

La validazione viene condotta in ambiente SUMO/TraCI e mostra riduzioni del ritardo comprese tra l'11% e il 29%, a seconda dello scenario. Il metodo risulta abbastanza robusto rispetto alle variazioni di flusso e ottiene prestazioni migliori sia del metodo di Webster sia del controllo attuato.

Sintesi della sezione

In sintesi, i lavori esaminati mostrano come gli algoritmi evolutivi si adattino bene a formulazioni multi-obiettivo del controllo semaforico, soprattutto quando è necessario considerare contemporaneamente ritardo, lunghezza delle code, consumo di carburante ed emissioni.

All'interno di questa famiglia emergono due aspetti ricorrenti. Da un lato, l'importanza del modello emissivo adottato, come nel caso dell'integrazione del VSP [WanZZ⁺24]. Dall'altro, il passaggio da intersezioni isolate a scenari multi-intersezione, che rende più rilevante la scelta dell'algoritmo di ottimizzazione, come evidenziato anche nel confronto tra GA e DE [Fai25]. Il framework bi-livello proposto in [ShaZWL24] mostra inoltre una possibile direzione di estensione di questi metodi verso scenari dinamici, combinando ottimizzazione offline e adattamento online.

2.2 Swarm Intelligence (PSO, ACO)

Le tecniche di swarm intelligence rappresentano un'altra classe di metaeuristiche, ispirate al comportamento collettivo di animali sociali come stormi di uccelli o colonie di formiche. A differenza degli algoritmi evolutivi, che operano tramite operatori genetici su una popolazione, gli approcci swarm-based modellano il comportamento di agenti individuali che seguono regole semplici ma che, attraverso l'interazione collettiva, convergono verso soluzioni di buona qualità.

2.2.1 PSO avanzata con dati IoT (IS-APCPSO)

In [HuaZ24] viene analizzato il problema dell'ottimizzazione di una rete stradale urbana in ambiente IoT, con particolare attenzione alla riduzione delle emissioni veicolari e al miglioramento dell'efficienza del traffico; il paper osserva che una parte rilevante dell'inquinamento è concentrata nelle intersezioni semaforizzate, dove le condizioni di attesa e idling vanno a contribuire in modo significativo alle emissioni complessive.

All'interno del paper è stato formulato il problema come un modello di ottimizzazione bi-livello di tipo Stackelberg. Al livello superiore, il sistema di gestione del traffico determina tre variabili decisionali: incremento di capacità dei tratti stradali y_a , durata del ciclo semaforico T_j e rapporto di verde λ_a . Al livello inferiore, il comportamento degli utenti viene descritto mediante un modello di *User Equilibrium*, in cui i flussi andranno a distribuirsi sulla rete minimizzando l'impedenza di percorrenza individuale.

La funzione obiettivo del livello superiore è espressa nel paper nella seguente forma:

$$\min Z(y_a, T_j, \lambda_a) = \sum_{a \in A} t_a(x_a, y_a, \lambda_a) x_a + c_{w1} \sum_{a \in A, i, j} ET_{a,i,j} + c_{w2} \sum_{a \in A, j} Cost_{Total} \quad (2.6)$$

In forma sintetica, la stessa funzione può essere interpretata come:

$$\min Z = \text{impedenza totale del sistema} + c_{w1} \text{ emissioni totali} + c_{w2} \text{ costo totale} \quad (2.7)$$

dove i coefficienti c_{w1} e c_{w2} convertono rispettivamente le emissioni totali e il costo totale nella stessa unità dell'impedenza del sistema; il costo totale include sia il road widening cost sia il costo di implementazione e gestione dell'IoT.

L'algoritmo IS-APCPSO (Immune Selection–Adaptive Promotion Chaotic PSO) va ad estendere la PSO standard con parametri adattivi che regolano l'equilibrio tra esplorazione e sfruttamento, una prima fase di ricerca caotica locale per le particelle migliori e un meccanismo di selezione immunitaria volto a preservare la diversità della popolazione, in tutto ciò l'algoritmo viene utilizzato in forma annidata per risolvere i due livelli del modello.

I risultati infatti evidenziano che le emissioni nelle intersezioni risultano nettamente superiori a quelle sui tratti stradali, e che l'integrazione di road widening, signal timing optimization e IoT consente di ridurre in modo significativo le emissioni complessive rispetto a strategie più tradizionali basate su un solo tipo di intervento.

2.2.2 DALM con meccanismo look-ahead

[LiaWMN24] studiano il problema della temporizzazione semaforica di una singola intersezione nel contesto dell'Internet of Vehicles (IoV), andando a proporre una variante dinamica dell'Ant Colony Optimization con meccanismo *look-ahead* (DALM).

Il sistema di traffico viene descritto come un sistema dinamico, non lineare e tempo-variante, dove le decisioni devono essere aggiornate in modo continuo sulla base dei dati raccolti in tempo reale da sensori e nodi di comunicazione. L'obiettivo è quello di ridurre il tempo di attesa, la lunghezza delle code e le emissioni di CO₂ all'intersezione. Il controllo si basa su un insieme di otto fasi semaforiche non conflittuali. Ogni formica costruisce una sequenza di fasi candidate, selezionandone quattro in modo da garantire che ogni direzione venga servita una sola volta all'interno del ciclo.

La probabilità che una formica k selezioni la fase j al tempo t è:

$$P_j^k(t) = \frac{[\tau_j(t)]^\alpha \cdot [T_{w,j,t}]^\beta \cdot [L_{j,t}]^\gamma}{\sum_{l \in a_k} [\tau_{jl}(t)]^\alpha \cdot [T_{w,jl}]^\beta \cdot [L_{jl}]^\gamma} \quad (2.8)$$

dove $\tau_j(t)$ indica il feromone associato alla fase j , $T_{w,j,t}$ il tempo di attesa e $L_{j,t}$ la lunghezza della coda. I parametri α , β e γ determinano il contributo relativo di ciascuna componente nella selezione della fase.

L'elemento distintivo e rilevante del metodo è il meccanismo di *look-ahead* che inserisce il concetto di *coda potenziale* per anticipare la domanda futura di verde:

$$L_{j,t} = \max \left(L_{j,t}^{(1)}, L_{j,t}^{(2)} \right), \quad L_{j,t}^{(k)} = L_1^{(k)} + \theta \cdot L_2^{(k)} \quad (2.9)$$

dove $L_1^{(k)}$ è la coda corrente, $L_2^{(k)}$ la coda potenziale e $\theta \in [0, 1]$ è un parametro di ponderazione. In questo modo, la scelta della fase non dipenderà solo dalla situazione osservata nell'istante corrente, ma andrà a tenere conto anche dell'evoluzione attesa del traffico nel breve periodo. Le emissioni di CO₂ vengono stimate tramite il modello HBEFA integrato in SUMO.

In conclusione viene mostrato come il DALM riduce in modo consistente il tempo totale di attesa e le emissioni di CO₂ rispetto ai metodi di confronto,

con dei benefici particolarmente evidenti soprattutto negli scenari di traffico medio e alto. L'influenza del meccanismo di look-ahead risulta legata proprio alla capacità di anticipare la formazione delle code attraverso la coda potenziale.

Sintesi della sezione

Gli approcci swarm si collocano nel quadro delle metaeuristiche per l'ottimizzazione del controllo semaforico, con una particolare attenzione alla gestione delle informazioni dinamiche sul traffico. Nel caso di IS-APCPSO [HuaZ24], il problema viene affrontato tramite una formulazione bi-livello che esplicita l'interazione tra controllo e comportamento degli utenti; nel caso di DALM-ACO [LiaWMN24], un aspetto distintivo è l'introduzione di un meccanismo di look-ahead, che permette di anticipare l'evoluzione delle code. In entrambi i lavori, la differenza rispetto a versioni più standard degli algoritmi sta soprattutto nell'integrazione di componenti adattive e nell'utilizzo di informazioni sullo stato dinamico del traffico.

2.3 Simulated Annealing e Programmazione Matematica

Il Simulated Annealing (SA) è una metaeuristica che è ispirata alla ricottura dei metalli; in sostanza a temperatura elevata accetta anche soluzioni peggiorative, raffreddandosi gradualmente verso la convergenza. Formalmente, una nuova soluzione peggiorativa viene accettata con probabilità $P = \exp(-\Delta F/T)$, dove ΔF è la variazione della funzione obiettivo e T la temperatura corrente. Questa tecnica consente di ridurre il rischio di convergere verso minimi locali, rendendola particolarmente adatta a spazi di ricerca complessi.

2.3.1 Framework integrato con previsione del traffico e Simulated Annealing

Attraverso [ZriKK25] gli autori propongono un framework integrato per il controllo semaforico adattivo. Il sistema combina una previsione a breve termine del traffico, l'ottimizzazione dei tempi di verde e un'infrastruttura di comunicazione distribuita (message-oriented middleware) basata su tecnologie come MQTT o Kafka.

Il Simulated Annealing (SA) viene utilizzato nel modulo di ottimizzazione, per determinare la durata ottimale delle fasi verdi a partire dai flussi di traffico previsti. L'obiettivo principale è minimizzare il tempo medio di attesa dei veicoli:

$$\min F = \sum_{i=1}^N w_i(t) \quad (2.10)$$

dove $w_i(t)$ rappresenta il tempo medio di attesa sull'approccio i al tempo t . Nel paper, quella quantità viene poi approssimata in funzione dei flussi previsti e della distribuzione dei tempi di verde, così da ottenere una formulazione compatibile con l'ottimizzazione in tempo quasi reale.

Il modulo di previsione riesce a fornire stime a breve termine dei flussi veicolari sulla base di dati storici, misure in tempo reale e informazioni contestuali, facendo riferimento a modelli di deep learning come LSTM e GNN. È quindi importante notare come il modello predittivo non prende direttamente decisioni di controllo semaforico: il controllo rimane affidato al SA, mentre la componente di machine learning fornisce esclusivamente l'input previsionale necessario all'ottimizzazione.

Le emissioni di CO₂ e il consumo di carburante non rientrano direttamente nella funzione obiettivo primaria, ma vengono utilizzati come riferimenti di valutazione dell'efficacia del sistema nel simulatore SUMO/TraCI. Il confronto tra schema fisso e schema adattivo mostra una riduzione del 21.6% nel tempo medio di attesa, del 12.9% nelle emissioni di CO₂ e del 15.6% nel consumo di carburante. Più che isolare il contributo di una singola tecnica, la ricerca ha evidenziato l'efficacia del framework integrato nel migliorare contemporaneamente le prestazioni del traffico e l'impatto ambientale.

2.3.2 Simulated Bifurcation su reti urbane

In [ZhaLY⁺25] viene affrontato il problema dell'ottimizzazione globale dei sistemi semaforici, in particolare su una rete urbana reale, utilizzando dei dati di traffico e topologia stradale derivati da un sottoinsieme della rete di Pechino. Il modello considera intersezioni a quattro vie con schema a otto fasi e riconduce il problema a una formulazione di tipo *spin glass*, adatta a descrivere sistemi complessi con interazioni multi-spin.

La funzione obiettivo è espressa tramite un Hamiltoniano globale

$$H[x(t), x(t-1)] = H_q[x(t)] + H_d[x(t), x(t-1), t],$$

dove il termine H_q rappresenta la varianza dei flussi di traffico tra le diverse direzioni e il termine H_d penalizza i cambi di fase, così da evitare switching troppo frequenti e favorire una maggiore stabilità del sistema. Nell'estensione del metodo, gli autori introducono inoltre un termine aggiuntivo per vincolare l'evoluzione allo spazio degli stati ammissibili.

Per risolvere il problema, viene adottata una versione estesa della Simulated Bifurcation (SB), che andrà ad essere combinata con *self-consistent field theory*. A differenza del Simulated Annealing (SA), che esplora lo spazio delle soluzioni tramite perturbazioni stocastiche, la SB adotta un approccio diverso: riformula il problema combinatorio come un sistema di equazioni dinamiche continue e ne segue l'evoluzione in modo deterministico, sfruttando i fenomeni di biforcazione del sistema. Gli esperimenti che sono svolti su dati reali di Pechino mostrano che la SB ottiene valori dell'Hamiltoniano mediamente più bassi e una convergenza più stabile rispetto al SA.

Nel complesso, i risultati suggeriscono che approcci ispirati alla fisica statistica possono costituire una valida alternativa alle metaeuristiche classiche per l'ottimizzazione semaforica su reti urbane complesse, in particolare nei casi in cui il problema risulti fortemente non lineare e difficile da affrontare con metodi più tradizionali.

Sintesi della sezione

Analizzando i contributi presi in riferimento, questi mostrano due possibili sviluppi dei metodi di ottimizzazione classica nel controllo semaforico. Da un lato, il framework presentato in [ZriKK25] riesce a integrare la previsione del traffico e Simulated Annealing, mantenendo il controllo semaforico affidato a una procedura di ottimizzazione esplicita. Dall'altra parte, il lavoro di [ZhaLY⁺25] considera il problema su una rete urbana reale e propone una formulazione matematica pensata per il coordinamento globale tra più intersezioni. In entrambi i casi, l'interesse del contributo non va a risiedere solo nella qualità della soluzione, ma anche nel tentativo di estendere questi metodi verso scenari più dinamici o più ampi.

2.4 Sistemi Fuzzy e Approcci Ibridi

I sistemi di controllo semaforico basati su logica fuzzy rappresentano un'alternativa comprensibile e computazionalmente leggera rispetto ad altri approcci di ottimizzazione globale o ai metodi basati su apprendimento automatico. Questi sistemi vanno ad operare mediante regole linguistiche (IF-THEN) e funzioni di appartenenza che modellano l'incertezza e la gradualità delle variabili di traffico, producendo decisioni di controllo trasparenti e facilmente comprensibili.

Sono inoltre presenti approcci ibridi che estendono i sistemi fuzzy combinandoli con metaeuristiche per ottimizzare automaticamente parametri e struttura del controllore, o con tecniche di coordinamento per affrontare scenari multi-intersezione.

2.4.1 Sistema fuzzy multi-agente a cinque moduli

Nel lavoro presentato in [ChaK25] viene proposto un sistema di controllo semaforico basato su logica fuzzy Mamdani e architettura multi-agente, progettato per la gestione decentralizzata di reti urbane con più intersezioni. Ogni

intersezione è gestita da un agente autonomo composto da cinque moduli funzionali:

1. **PECM** (Prioritize Emergency Car Module): rileva veicoli di emergenza e assegna loro priorità;
2. **HTEM** (Heavy Traffic Evaluation Module): valuta il livello di congestione sulla base di code e tempi di attesa;
3. **CWTM** (Calculating Waiting Time Module): garantisce equità tra i flussi, evitando starvation;
4. **CDVM** (Calculating Distance of Vehicles Module): stima la distanza dei veicoli dall'intersezione per migliorare la regolazione dinamica del verde;
5. **ETDM** (Extension Time Decision Module): determina l'estensione del verde combinando le informazioni dei moduli precedenti.

Il modulo CDVM rappresenta il contributo originale principale, poiché consente di tenere conto della prossimità dei veicoli all'intersezione e di adattare in modo più accurato la durata del verde. Per limitare la crescita combinatoria della base di regole, gli autori hanno optato per una riduzione che porta il numero complessivo di regole da 818 a 210.

Dalle simulazioni in SUMO emerge una riduzione del tempo medio di attesa rispetto alla versione priva del modulo CDVM, che raggiunge il 48.65% nello scenario con nove intersezioni. Il paper evidenzia inoltre che la versione collaborativa del sistema multi-agente ottiene risultati migliori di quella non collaborativa su tutti gli indicatori, sebbene il test statistico (paired T-test) non rilevi una differenza significativa; un altro risultato particolarmente interessante è l'osservazione di un fenomeno emergente di *green wave* anche in assenza di comunicazione diretta tra gli agenti, suggerendo che semplici regole locali possono generare forme di coordinamento globale.

2.4.2 Framework ibrido FPA + Type-2 Fuzzy + CSA

[KorA24] propongono un sistema di controllo semaforico ibrido che integra tre tecniche eterogenee, ciascuna dedicata a un sotto-problema specifico.

Il **Flower Pollination Algorithm (FPA)**, un algoritmo bio-ispirato che combina ricerca globale basata su Lévy flights con ricerca locale stocastica, viene utilizzato nel modulo di phase optimization per determinare il *phase plan* e la lunghezza ottimale del ciclo semaforico. Uno degli aspetti più distintivi del lavoro è infatti l'ottimizzazione congiunta di phase plan, phase sequence e signal timing.

La **logica fuzzy di tipo 2** viene utilizzata nel modulo di signal timing optimization per decidere, sulla base del numero di veicoli e della lunghezza della coda, se è opportuno aumentare o diminuire il tempo di verde. A differenza della fuzzy di tipo 1, le funzioni di appartenenza in questo caso sono tridimensionali e modellano esplicitamente l'incertezza associata alle misure di traffico tramite il *footprint of uncertainty*, che richiede un successivo passaggio di type-reduction in fase di defuzzificazione.

Il **Crow Search Algorithm (CSA)** ottimizza i centri e i lati delle funzioni di appartenenza del sistema fuzzy in funzione delle diverse condizioni di traffico, così da poter adattare il controllore a scenari operativi differenti.

La funzione obiettivo consiste nella minimizzazione del ritardo medio per veicolo, calcolato secondo la formulazione dell'Highway Capacity Manual (HCM 2000). La valutazione viene condotta su 36 scenari sperimentali, ottenuti combinando nove condizioni di traffico e quattro geometrie di intersezione, ciascuno simulato con 20 seed casuali. I risultati permettono di mostrare che il framework ibrido ottiene circa il 32% di miglioramento rispetto al controllo a tempo fisso e prestazioni migliori del 5% e del 6% rispetto ai sistemi FPA-TSC e Type-1 FL-TSC, rispettivamente.

2.4.3 Ottimizzazione network-level con speed advisory

Nell'articolo presentato in [YuaZ24] viene proposto un metodo di controllo connesso tra i segnali semaforici e la velocità dei veicoli in ambiente di *connected vehicles*, applicato a un'arteria reale composta da 11 intersezioni

nella città di Tsukuba (Giappone). Il contributo del lavoro consiste nell'integrare il coordinamento dei segnali con il controllo della velocità, superando gli approcci tradizionali focalizzati sulla singola intersezione o sul solo signal control.

Il problema è formulato su due livelli. A livello locale, l'obiettivo è minimizzare il ritardo totale

$$\min D(g) = \sum_{m=1}^N D_m(g),$$

dove

$$D_m(g) = T_m(g) - A_m(g)$$

rappresenta la differenza tra l'istante reale di uscita dalla linea d'arresto e quello che si avrebbe in condizioni di flusso libero, con vincolo $g \geq 5$ s. A livello di rete, il metodo ottimizza il coordinamento lungo l'arteria massimizzando la *GreenWave bandwidth*

$$\max b,$$

soggetta ai vincoli fisici che legano banda verde, tempi di rosso e offset semaforici.

Un elemento distintivo del lavoro è il controllo adattivo della velocità dei veicoli connessi. A livello di singola intersezione, la velocità consigliata viene calcolata a partire dalla distanza dal semaforo, dalla lunghezza stimata della coda e dalla velocità della shockwave:

$$v = \frac{L_i - l_i}{g_{ij} + l_i/v_i} \quad (2.11)$$

dove L_i è la distanza dall'intersezione, l_i la lunghezza della coda, v_i la velocità dell'onda di congestione e g_{ij} l'istante di inizio del verde all'intersezione i . Per il caso di intersezioni multiple, il paper estende questa logica costruendo un insieme di velocità ammissibili che consenta al veicolo di attraversare consecutivamente più intersezioni senza arresti non necessari.

La valutazione in VISSIM con dati reali mostra riduzioni di ritardo, fermate, consumi ed emissioni di CO₂, con benefici particolarmente evidenti in

condizioni di traffico medio. Il paper riesce a mostrare che la combinazione di controllo semaforico e controllo della velocità ottiene risultati migliori rispetto all'uso separato delle due strategie, al prezzo però di una gestione centralizzata e di una modellazione dettagliata dell'arteria e dei veicoli connessi.

Sintesi della sezione

Nel complesso, i sistemi fuzzy e gli approcci ibridi risultano abbastanza efficaci nel mantenere una struttura di controllo coerente, anche quando gli scenari diventano più complessi.

Ad esempio, il sistema multi-agente presentato in [ChaK25] mostra come semplici regole locali possano portare a forme di coordinamento tra più intersezioni. Allo stesso modo, il framework descritto in [KorA24] evidenzia come la combinazione tra metaeuristiche e logica fuzzy possa essere sfruttata per ottimizzare congiuntamente phase plan, phase sequence e temporizzazione semaforica.

Infine, il lavoro in [YuaZ24] estende l'analisi a una scala di rete, includendo anche il coordinamento tra semafori e il controllo della velocità in scenari con veicoli connessi.

2.5 Approcci Rule-Based Deterministici

A differenza delle metaeuristiche e degli approcci basati su apprendimento automatico presentati nelle sezioni precedenti, questa sezione va a considerare metodi deterministici e rule-based che non impiegano tecniche di AI/ML. Questi approcci oltre a risultare immediatamente applicabili anche su infrastrutture tradizionali, riescono a offrire una piena trasparenza del processo decisionale.

2.5.1 Simulazione e Design of Experiments

[MonGF⁺18] propongono un approccio deterministico e comprensibile che combina microsimulazione del traffico e Design of Experiments (DoE) per

analizzare sistematicamente l'impatto delle configurazioni semaforiche su un'arteria urbana reale composta da 14 intersezioni semaforizzate.

Il metodo considera tre fattori sperimentali con tre livelli ciascuno: la strategia di sincronizzazione (current, parallel, staggered), il tempo di ciclo ($\{100, 120, 140\}$ s) e il limite di velocità ($\{40, 50, 60\}$ km/h). L'obiettivo è quello di minimizzare simultaneamente tre variabili di risposta: il tempo medio nel sistema, le emissioni di CO₂ e il consumo di carburante.

Il disegno fattoriale completo 3^3 , replicato due volte, genera 27 trattamenti e 54 esecuzioni complessive. La valutazione viene condotta in PTV VISSIM e analizzata tramite ANOVA e grafici degli effetti principali. I risultati mostrano che la configurazione migliore corrisponde alla sincronizzazione *staggered*, a un ciclo di 100 s e a un limite di velocità di 60 km/h, con un'elevata capacità esplicativa del modello statistico.

Il miglioramento osservato viene attribuito soprattutto alla riduzione degli arresti alle intersezioni e del numero di eventi di accelerazione, che si traducono in minori emissioni e minore consumo di carburante. Il principale limite dell'approccio risiede nella necessità di competenze specialistiche per costruire e calibrare il modello di simulazione, oltre che nell'uso di software dedicato come PTV VISSIM.

2.5.2 Controllo adattivo emission-based

In [LerU21] viene proposto un algoritmo adattivo per intersezioni isolate che utilizza direttamente le emissioni di CO come criterio di regolazione del controllo, anziché metriche surrogate. L'incremento di emissione dovuto al controllo semaforico è modellato come la somma, su tutte le modalità di guida, della differenza tra l'emissione con semaforo e quella senza:

$$E = \sum_{j=1}^M (E_j^A - E_j^B) \quad (2.12)$$

dove E_j^A e E_j^B sono le emissioni nella modalità j (cruising, decelerazione, accelerazione, idle) rispettivamente in presenza e in assenza del semaforo, ciascuna ottenuta come prodotto del tasso di emissione EF_j per il tem-

po trascorso nella modalità, determinato geometricamente dai diagrammi tempo-spazio basati sulla teoria delle shockwave.

La politica di controllo valuta la derivata dell'emissione totale rispetto al tempo di rosso:

$$\frac{dE}{dR} = \frac{dE_{CR}}{dR} + \frac{dE_{dec}}{dR} + \frac{dE_{acc}}{dR} + \frac{dE_{idle}}{dR} \quad (2.13)$$

Il segno di tale derivata determina se aumentare o diminuire lo split semaforico. La lunghezza del ciclo viene adattata separatamente sulla base delle code residue.

La validazione in AIMSUN viene svolta su sei scenari, con rapporto di flusso critico compreso tra 0.5 e 1.0. Questo mostra che il metodo permette di ridurre in media le emissioni di CO del 7.67% rispetto al controllo fisso di Webster. I benefici su ritardo e tempo di arresto risultano particolarmente evidenti negli scenari più congestionati, con miglioramenti massimi pari a circa il 33% nel ritardo e al 36% nel tempo di arresto.

2.5.3 SmartLight: scheduling energy-aware

In [YouBR23] gli autori introducono SmartLight, un algoritmo di scheduling semaforico esplicitamente orientato all'efficienza energetica. I veicoli vengono raggruppati in platoon sulla base della prossimità spaziale e della velocità, e per ciascun platoon viene calcolato un *index factor*:

$$E = \sum_{i=1}^{N_{PV}} mV_i.S \times mV_i.W \quad (2.14)$$

che sintetizza l'impatto complessivo del flusso in termini di velocità e peso relativo dei veicoli in movimento. Valori elevati indicano flussi per i quali l'arresto comporterebbe un maggiore spreco energetico; tali flussi ricevono quindi maggiore priorità nella schedulazione.

Il consumo istantaneo di carburante è stimato mediante l'Instantaneous

Fuel Consumption Model (IFCM), che utilizza la forza di trazione

$$R_t = D_f + I_f + G_f \quad (2.15)$$

e un consumo dipendente dal segno di R_t . Le emissioni di CO_2 sono stimate come funzione lineare del carburante consumato.

La selezione delle fasi combina diverse informazioni sul traffico reale, tra cui la presenza di veicoli prioritari, il waiting delay, l'index factor e la densità dei flussi. La valutazione in SUMO+NS-2 mostra come SmartLight riduce il consumo di carburante di circa il 20% rispetto a ITLC e del 30% rispetto a ETLA. Il miglioramento energetico è accompagnato da una lieve riduzione del throughput rispetto a ITLC (circa 5%), ma da prestazioni comunque superiori a ETLA.

2.5.4 Controllo cost-based per piccole città

Nel lavoro presentato in [SanRMD24] viene analizzato un algoritmo di controllo semaforico smart esplicitamente orientato alla riduzione delle emissioni di CO_2 nel contesto delle piccole città, un ambito ancora poco esplorato.

Il sistema non fa uso di comunicazioni V2V/V2I né di tecniche di apprendimento automatico, ma si basa su un algoritmo attivo che determina il cambio di fase in funzione del costo associato ai flussi concorrenti.

Per ciascun flusso, il costo istantaneo è definito come:

$$Cost_f = \sum_{l \in f} \sum_{i \in l} (J + K v_i^2) \quad (2.16)$$

dove J rappresenta una costante associata alla presenza dei veicoli, mentre il termine $K v_i^2$ descrive il contributo dovuto alla loro energia cinetica. L'idea di fondo è che arrestare un flusso con veicoli in movimento comporti un costo maggiore, poiché l'energia persa dovrà essere recuperata successivamente bruciando carburante. L'algoritmo avvia il cambio di fase quando il costo del flusso bloccato supera il costo di arrestare il flusso attivo, con vincoli

di sicurezza che impongono a ciascun flusso un tempo minimo di 10 s e un tempo massimo di 180 s.

La stima delle emissioni viene effettuata nel simulatore tramite il parametro di *Vehicle Specific Power* (VSP):

$$\text{VSP} = v(1.1a + 9.81s + 0.132) + 0.000302v^3 \quad (2.17)$$

che esprime la potenza istantanea per unità di massa del veicolo a partire da velocità, accelerazione e pendenza stradale. Infatti a partire dal VSP, il simulatore stima il consumo di carburante e le relative emissioni di CO₂ applicando coefficienti specifici per tipologia di veicolo e di carburante.

La valutazione su un'intersezione reale a Castelo Branco, in Portogallo, mostra che l'algoritmo intelligente riduce le emissioni di CO₂ del 32–40% a seconda della densità di traffico, con degli incrementi della velocità media compresi tra il 60% e il 101% e riduzioni del tempo di attesa tra il 53% e il 95%; quindi viene evidenziato, che anche senza infrastrutture V2X o modelli di apprendimento, strategie cost-based progettate direttamente sull'obiettivo emissivo possono produrre benefici rilevanti in scenari urbani di piccola scala.

2.5.5 Rule-based eco traffic lights

In [CabCMM24] viene esteso un precedente algoritmo cost-based per la riduzione delle emissioni, inizialmente limitato a una singola intersezione a tre vie. L'algoritmo di partenza prevedeva già due ottimizzazioni: la prima era l'aumento del peso cinetico per i veicoli del flusso in movimento (*Enhancement 1*) e la seconda era *safeguard phase skipping* quando non vi sono più veicoli da arrestare nel flusso da bloccare (*Enhancement 2*). I contributi originali introdotti in [CabCMM24] sono invece la generalizzazione a intersezioni con un numero arbitrario di rami (N-way) e l'*early termination* del ciclo, che consente di interromperlo anticipatamente quando i flussi rimanenti hanno costo nullo.

Nel caso delle intersezioni N-way, il controllore adotta una logica a cicli: in ogni ciclo ciascun traffic flow deve ricevere il verde esattamente una volta, e il verde viene assegnato al flusso con costo più elevato tra quelli che non lo

hanno ancora ricevuto nel ciclo corrente. Questa scelta consente di mantenere una forma di fairness pur lasciando dinamica la sequenza delle fasi.

La validazione in SUMO considera sia mappe sintetiche, comprendenti intersezioni singole e griglie di tipo Manhattan, sia una mappa reale dell'area urbana di Modena. La popolazione veicolare è stata costruita utilizzando diverse tipologie di veicoli secondo i rapporti di adozione riportati dall'ACI, mentre le emissioni di CO₂ sono stimate tramite le funzioni di simulazione di SUMO.

Nei vari esperimenti si osserva che la riduzione delle emissioni è significativa anche con un singolo semaforo intelligente e cresce progressivamente all'aumentare del numero di semafori controllati. Dai dati risulta che, anche senza un coordinamento esplicito tra i semafori, il sistema riesce comunque a ridurre emissioni, tempi di attesa e tempi di percorrenza, mantenendo buone prestazioni anche in scenari urbani più complessi.

Sintesi della sezione

Gli approcci rule-based esaminati mostrano che il controllo semaforico eco-orientato può essere affrontato anche senza ricorrere a tecniche di apprendimento o a procedure di ottimizzazione particolarmente complesse. In questa famiglia si ritrovano sia metodi fondati su simulazione e analisi sperimentale [MonGF⁺18], sia strategie adattive che modellano direttamente il fenomeno emissivo [LerU21], sia algoritmi di scheduling e switching basati su criteri energetici espliciti [YouBR23, SanRMD24, CabCMM24]. Il tratto comune è la trasparenza del processo decisionale, unita alla possibilità di operare anche in contesti infrastrutturali relativamente semplici.

2.6 Approcci basati su Reinforcement Learning

Il Reinforcement Learning (RL) affronta il controllo semaforico come un processo decisionale sequenziale in cui un agente apprende una politica di controllo tramite interazione con l'ambiente, senza richiedere un modello espli-

cito del sistema. A ogni passo temporale, l'agente va a osservare lo stato dell'intersezione, seleziona un'azione (la fase semaforica da attivare) e riceve un reward che guida l'apprendimento.

Rispetto alle metaeuristiche, il reinforcement learning è in grado di adattarsi meglio alle condizioni dinamiche del traffico. Questo ovviamente però comporta alcuni compromessi, tra cui una minore interpretabilità, una fase di addestramento più costosa dal punto di vista computazionale e una validazione più difficile.

2.6.1 DRL-DG: Deep RL a doppio obiettivo

[ZhaCHZ24] introducono DRL-DG (Deep Reinforcement Learning with Dual Goals), un sistema progettato per ottimizzare simultaneamente efficienza del traffico e riduzione delle emissioni di CO₂.

Lo stato dell'intersezione è rappresentato tramite una griglia binaria 8×10 a discretizzazione non uniforme: le celle vicine all'intersezione misurano 7 m, mentre quelle più lontane arrivano fino a 180 m. Lo spazio delle azioni consiste nella selezione di una delle quattro fasi ammissibili: $\mathcal{A} = \{EWG, EWLG, NSG, NSLG\}$. Il verde minimo è fissato a 10 s ed è esteso di 10 s ogni volta che l'agente rileleziona la stessa fase, fino a una durata massima di 60 s; inoltre, dopo sei selezioni consecutive della medesima fase il sistema forza il passaggio alle altre.

L'agente utilizza un'architettura Double DQN (DDQN) con rete ibrida: due strati convoluzionali (100 e 10 filtri, kernel 1×3 , stride 2), uno strato LSTM (2 livelli, 80 unità, dropout 0.2) per catturare le dipendenze temporali, e cinque strati fully connected da 400 neuroni. La funzione target è:

$$y(s, a) = r + \gamma Q\left(s', \arg \max_{a'} Q(s', a'; w_\theta); w^t\right) \quad (2.18)$$

Il contributo principale è la funzione di reward a doppio obiettivo:

$$R^{(t)} = W_{TE} \cdot R_{TE}^{(t)} + W_{CE} \cdot R_{CE}^{(t)} \quad (2.19)$$

con $R_{TE}^{(t)} = -(CWT^{(t+1)} - CWT^{(t)})$ e $R_{CE}^{(t)} = -(PE^{(t+1)} - PE^{(t)})$, dove CWT

è il tempo di attesa cumulativo e PE le emissioni cumulative. I pesi sono determinati automaticamente tramite l'*entropy weight method*:

$$H_j = -\frac{1}{\ln m} \sum_{i=1}^m P_{ij} \ln P_{ij}, \quad w_j = \frac{1 - H_j}{\sum_{k=1}^n (1 - H_k)} \quad (2.20)$$

in modo da bilanciare dinamicamente traffico ed emissioni riducendo il ricorso a una pesatura soggettiva.

Le emissioni sono calcolate tramite il modello di emissione di SUMO, basato sulla potenza istantanea del motore:

$$PE = \frac{P_{Roll} + P_{Air} + P_{Accel} + P_{Grad}}{\eta_{gearbox}} \quad (2.21)$$

Su uno scenario simulato costruito a partire da un'intersezione reale di Changsha (Cina), con traffico modellato tramite distribuzione di Weibull, DRL-DG viene confrontato con controllo a tempo fisso (FTSC), controllo attuato (ASC) e una variante DRL a singolo obiettivo (DRL-SG). I risultati medi riportati mostrano che DRL-DG ottiene il valore più basso di emissioni di CO₂, pari a 94.97 g/veicolo, rispetto a 313.59 g/veicolo di FTSC, 200.84 g/veicolo di ASC e 163.64 g/veicolo di DRL-SG. Sul piano dell'efficienza, DRL-DG raggiunge un tempo medio di attesa di 15.68 s/veicolo, molto vicino al valore di DRL-SG, pari a 14.79 s/veicolo, e nettamente inferiore rispetto ai 95.27 s/veicolo di FTSC e ai 53.61 s/veicolo di ASC. Questo suggerisce che l'introduzione dell'obiettivo ambientale comporta solo un lieve compromesso rispetto al controllore DRL ottimizzato esclusivamente sul traffico, mantenendo comunque prestazioni molto superiori ai controllori tradizionali.

2.6.2 Multi-Agent DQN cooperativo

In [KolKBA23] gli autori estendono il problema del controllo semaforico a una rete di 6 intersezioni mediante un approccio di *Multi-Agent Reinforcement Learning* (MARL) cooperativo basato su *Deep Q-Learning*. Il metodo adotta una logica cooperativa a parametri condivisi, in cui un'unica rete neu-

rale governa tutti gli agenti senza comunicazione esplicita tra le intersezioni: ciascun agente riceve in ingresso lo stato delle corsie e l'identificativo del proprio incrocio e ottiene un reward locale, mentre la cooperazione emerge indirettamente attraverso l'ambiente condiviso.

Lo stato include l'*halting number* percentuale di tutte le corsie in ingresso della rete, cioè il numero di veicoli con velocità inferiore a 0.1 m/s, e l'identificativo dell'intersezione corrente in forma binaria. Lo spazio delle azioni è $\mathcal{A} = \{NSG, EWG\}$, con fasi di 30 s separate da un giallo di 3 s.

La DQN utilizzata adotta 2 strati nascosti da 256 neuroni con attivazione ReLU ed è addestrata con Adam ($\alpha = 0.0003$, $\gamma = 0.95$, batch size 512, memoria di esperienza 15 000). La funzione di reward non è espressa nel paper tramite una formula chiusa, ma viene descritta in termini della deviazione standard σ dell'*halting number* delle quattro strade in ingresso. L'obiettivo dell'agente è minimizzare tale deviazione standard, così da bilanciare il traffico tra le diverse direzioni. Il reward massimo, pari a 1, si ottiene quando $\sigma = 0$, cioè quando tutte le corsie in ingresso risultano uniformemente caricate. Il paper specifica inoltre che, per $\sigma < 0.1$, il reward assume valori nell'intervallo $[0, 1]$, mentre per $\sigma \geq 0.1$ assume valori in $[-1, 0]$.

Su 10 000 episodi di test, il controller DQN-MARL ottiene:

- tempo di attesa medio: 236.28 s (−24% rispetto al ciclo fisso, −19% rispetto al controllo attuato, −15% rispetto a PG-SA);
- emissioni di CO₂: 1143.75 kg (−15% rispetto al ciclo fisso, −11% rispetto all'attuato, −7% rispetto a PG-SA);
- consumo di carburante: 503.21 L, contro i 597.04 L del ciclo fisso.

Il confronto con PG-SA, cioè il controller del precedente lavoro degli autori addestrato su singola intersezione con la stessa logica di reward, mostra quindi un ulteriore miglioramento sia nelle metriche classiche sia in quelle ambientali, evidenziando il vantaggio del passaggio a un approccio multi-agente su rete. Il paper sottolinea però anche alcuni vincoli applicativi, tra cui la necessità di infrastrutture di rilevamento, come *loop detectors* o traffic cameras, e il supporto a meccanismi di comunicazione V2X.

Sintesi della sezione

Riassumendo, i lavori basati su reinforcement learning affrontano il controllo semaforico come un problema decisionale sequenziale in cui la politica viene appresa dall'interazione con l'ambiente. Il contributo di [ZhaCHZ24] mostra come l'obiettivo ambientale possa essere inserito direttamente nella reward senza rinunciare alla qualità del controllo del traffico; il lavoro di [KolKBA23] estende invece l'analisi a uno scenario multi-intersezione, evidenziando il ruolo della cooperazione indiretta tra agenti. In entrambi i casi, i risultati confermano il potenziale adattivo del paradigma, ma anche la necessità di validazione su scenari più ampi e realistici.

2.7 Sintesi critica dello stato dell'arte

I diciotto lavori analizzati mostrano che il controllo semaforico eco-sostenibile viene oggi affrontato attraverso paradigmi metodologici differenti, accomunati dall'obiettivo di migliorare le prestazioni del traffico riducendo al tempo stesso l'impatto ambientale della circolazione veicolare. Nel loro insieme, questi contributi delineano un panorama articolato, in cui convivono metodi più semplici e interpretabili, come quelli *rule-based* o fondati su simulazione e *design of experiments*, e approcci più adattivi e *data-driven*, in particolare quelli basati su *deep reinforcement learning* e *multi-agent reinforcement learning*.

Confronto tra paradigmi metodologici. Le metaeuristiche — in particolare algoritmi evolutivi e tecniche di *swarm intelligence* — trattano il controllo semaforico soprattutto come un problema di ottimizzazione e risultano adatte quando l'obiettivo è bilanciare più metriche, come ritardo, lunghezza delle code, consumo di carburante ed emissioni. Il loro limite principale è che la qualità dei risultati dipende in misura significativa dal simulatore adottato, dal modello emissivo integrato e dalla possibilità di eseguire valutazioni ripetute spesso costose dal punto di vista computazionale. Gli approcci basati su *reinforcement learning*, al contrario, offrono una mag-

giore capacità di adattamento a condizioni di traffico variabili, ma pongono criticità più marcate in termini di interpretabilità, addestramento, trasferibilità e validazione. In posizione intermedia si collocano i sistemi fuzzy e gli approcci *rule-based*, che rinunciano in parte alla flessibilità dei metodi più sofisticati ma mantengono un vantaggio rilevante sul piano della trasparenza e della fattibilità applicativa.

Realismo ambientale e scala del problema. Un primo elemento di differenziazione tra i lavori riguarda il modo in cui viene rappresentato l'impatto ambientale. In parte della letteratura tale aspetto è trattato indirettamente, attraverso metriche surrogate come tempo di attesa, velocità media o numero di fermate, assumendo che una maggiore fluidità del traffico si traduca automaticamente in minori emissioni. Altri contributi adottano invece modelli emissivi espliciti, come il *Vehicle Specific Power* o modelli di emissione integrati nei simulatori, mettendo in evidenza che accelerazioni, decelerazioni e fasi di arresto possono incidere in modo rilevante anche quando le metriche di traffico migliorano. Da questo punto di vista, i lavori metodologicamente più solidi sono quelli che separano chiaramente la valutazione della mobilità da quella ambientale e incorporano nella funzione obiettivo una stima emissiva esplicita.

Un secondo aspetto riguarda la scala del problema affrontato. Molti contributi considerano intersezioni isolate, utili per verificare il comportamento di un algoritmo in un contesto controllato, ma solo in parte rappresentativi della complessità urbana reale. Quando il problema viene esteso a reti multi-intersezione, emergono nuove esigenze: coordinamento tra nodi, gestione di flussi non locali, maggiore dimensionalità dello spazio decisionale e costi computazionali più elevati. Il passaggio a scenari *network-level* non costituisce quindi soltanto un'estensione quantitativa, ma introduce un cambiamento qualitativo nella natura stessa del problema.

Indicazioni che emergono dal confronto. Nel complesso, nessuna famiglia di metodi può essere considerata universalmente superiore. Le metaeuristiche risultano efficaci quando il problema è formulato come ottimizzazione

multi-obiettivo; il *reinforcement learning* appare promettente nei contesti dinamici, ma richiede un processo di sviluppo e validazione più oneroso; i sistemi fuzzy e gli approcci *rule-based* offrono invece maggiore semplicità, interpretabilità e minori requisiti infrastrutturali. Per un contesto applicativo realistico, questi ultimi aspetti assumono un peso non trascurabile, soprattutto quando l'obiettivo non è soltanto ottenere buone prestazioni in simulazione, ma anche progettare strategie comprensibili, controllabili e potenzialmente trasferibili.

Requisiti e progetto

In questo capitolo vengono definiti il punto di partenza e il contributo della tesi, nonché i requisiti che il nuovo algoritmo di controllo semaforico deve soddisfare. L'analisi prende avvio dai limiti del controllore di riferimento e prosegue con la descrizione della soluzione proposta.

Il capitolo è articolato in sei sezioni: la prima presenta il punto di partenza e il contributo della tesi; la seconda definisce i requisiti; la terza analizza i limiti del controllore di riferimento; la quarta descrive la nuova funzione di costo; la quinta introduce l'*Adaptive Mix*; la sesta illustra le ipotesi del modello e le tecnologie sensoristiche necessarie per una possibile applicazione reale.

3.1 Il punto di partenza e il contributo della tesi

Il quadro tracciato nel capitolo 2 mette in luce un terreno scoperto. Le famiglie più sofisticate, dalle metaeuristiche multi-obiettivo al reinforcement learning, ottengono buoni risultati in simulazione, ma a un prezzo che ne limita l'adozione reale: valutazioni computazionalmente onerose, fasi di addestramento su dati storici, coordinamento centralizzato o infrastrutture di comunicazione veicolo-infrastruttura. Gli approcci rule-based deterministici si trovano all'opposto: rinunciano a parte della flessibilità, ma in cambio offrono trasparenza, decentralizzazione e applicabilità immediata sulle infrastrutture esistenti.

In quest'ultima famiglia si colloca il controllore di riferimento sviluppato all'interno del gruppo di ricerca [CabCMM24], su cui si basa questa tesi. Si tratta di un controllore cost-based e decentralizzato, che decide la fase da servire confrontando i costi dei flussi calcolati su misure locali e mantenendo,

attraverso un meccanismo di rotazione, una forma di equità ciclica fra le fasi principali.

Pur condividendone le proprietà che lo rendono interessante, il controllore presenta due limiti che ne attenuano l'efficacia sulle reti urbane reali. Il primo riguarda ciò che il controllore vede del traffico: la decisione si basa sui veicoli presenti sulla corsia in ingresso, e nient'altro. Tutto ciò che sta arrivando dalle strade a monte non entra nel calcolo, e i plotoni in avvicinamento non vengono anticipati. Il secondo riguarda ciò che il controllore ignora dell'altro lato dell'incrocio: lo stato delle strade in uscita non è osservato, e così il verde può essere assegnato anche verso destinazioni già sature, alimentando fenomeni di spillback. Il **contributo della tesi** consiste nel superare entrambi i limiti senza rinunciare alle proprietà che rendono questa classe di controllori realisticamente applicabile. A questo scopo si introducono due elementi. Il primo è una nuova funzione di costo, che integra la domanda dei veicoli in avvicinamento dalle strade a monte, una penalità verso le destinazioni dove sono già presenti code, e un fattore di saturazione locale per le corsie prossime alla congestione. Il secondo è il meccanismo di Adaptive Mix, che a livello di singolo semaforo e sulla base dell'occupazione osservata bilancia in modo dinamico la logica del controllore di riferimento e quella della nuova formula. Il progetto di questi due elementi è oggetto del Capitolo 3.

3.2 Requisiti

Il nuovo algoritmo deve soddisfare alcuni requisiti che derivano **dall’obiettivo generale della tesi** (ovvero ridurre le emissioni di CO₂ e i tempi di attesa nelle reti urbane) e dal **contesto applicativo** (essere realmente utilizzabile dalle amministrazioni cittadine senza richiedere infrastrutture di nuova generazione).

I requisiti principali sono i seguenti:

Decentralizzazione: ogni semaforo deve poter decidere in maniera autonoma quando cambiare fase, senza dover comunicare con altri semafori e senza dover essere coordinato da un’autorità centrale. Questo requisito è importante perché un sistema decentralizzato è più resiliente, infatti un guasto di un singolo semaforo non si propaga, inoltre è più economico da installare e più facile da estendere in modo incrementale (si può equipaggiare prima una parte della città e poi il resto).

Interpretabilità: l’algoritmo deve essere basato su regole esplicite scritte in modo trasparente, e non su modelli appresi da dati come quelli usati nel *Reinforcement Learning* o nelle *reti neurali profonde*. Le motivazioni sono diverse: da un lato, gli algoritmi interpretabili sono più facili da verificare e da accettare per le pubbliche amministrazioni; dall’altro, l’interpretabilità permette di tarare manualmente i parametri e di comprendere perché il sistema prende una determinata decisione, evitando l’effetto *black box*.

Uso esclusivo di dati locali: il metodo deve basarsi solo su informazioni disponibili a livello locale, come lo stato delle corsie in ingresso, l’occupazione delle strade a valle e a monte e la velocità dei veicoli in transito. Non deve assumere la presenza di sistemi di comunicazione veicolo-infrastruttura (V2X), GPS individuali o tracciamento dei veicoli. Tutti i dati richiesti devono essere ottenibili da sensori passivi installati a bordo strada.

Conservazione dell'infrastruttura semaforica esistente: bisogna mantenere invariati il numero e il tipo delle fasi del piano semaforico originale, come l'ordine ciclico e la durata dei segnali di giallo e rosso completo, andando a sostituire solo la logica di durata e transizioni delle fasi semaforiche; questo consente un *deployment* immediato sui semafori esistenti senza dover riprogettare gli impianti.

Deployment immediato: non deve essere necessaria alcuna fase di addestramento su dati storici. I parametri vanno fissati a priori sulla base di considerazioni qualitative interpretabili, in modo che il sistema sia pronto a funzionare dal primo giorno di installazione.

3.3 Limiti del controllore di riferimento

Il modello di controllo assunto come riferimento (l'algoritmo [CabCMM24]) soddisfa già diversi dei requisiti elencati nella sezione precedente: è decentralizzato, *rule-based*, basato su dati locali, e non richiede infrastrutture aggiuntive. Tuttavia, presenta due limiti strutturali che l'algoritmo esposto in questa tesi intende superare.

Limite 1: cieco rispetto al traffico in avvicinamento. Il sistema valuta solo i veicoli che si trovano già sulla propria corsia in ingresso al semaforo. Un plotone in arrivo dalla strada a monte non viene considerato nella formula del costo finché i veicoli non hanno raggiunto la corsia controllata e si sono fermati in coda. Ciò impone un comportamento puramente reattivo e impedisce qualsiasi forma di anticipazione.

Limite 2: cieco rispetto alla congestione downstream. Non viene considerato lo stato delle strade in uscita dall'incrocio. Può quindi dare il verde a un flusso anche quando la strada a valle verso cui i veicoli sono diretti è già satura, contribuendo a peggiorare il fenomeno dello *spillback* (la coda che si propaga all'indietro fino a bloccare l'incrocio stesso). In una rete urbana reale con asimmetrie e colli di bottiglia, questo limite può vanificare

l'effetto di una gestione adattiva.

In sintesi, l'obiettivo del progetto è introdurre due nuovi meccanismi che superino entrambi i limiti, senza rinunciare alle proprietà desiderabili del controllore di riferimento.

3.4 Progetto della funzione di costo

La prima componente del progetto è una nuova funzione di costo. La sua struttura è radicalmente diversa da quella del controllore di riferimento: invece di sommare i contributi di ogni singolo veicolo, il costo di un flusso è ora la somma, sulle sue corsie in ingresso, del prodotto di tre componenti che caratterizzano aspetti diversi della situazione del traffico:

$$Cost_f = \sum_{l \in f} [demand_l \cdot ds_penalty_l \cdot cong_factor_l] \quad (3.1)$$

La notazione è la stessa del Capitolo 2: f è il flusso (un insieme di corsie che ricevono il verde insieme) e l una sua corsia in ingresso. La componente $demand_l$ misura quanta domanda di traffico c'è al momento sulla corsia, mettendo insieme i veicoli fermi e quelli che stanno arrivando dalle strade a monte. La componente $ds_penalty_l$ tiene conto di quanto è piena la strada in cui i veicoli dovrebbero andare. Il fattore $cong_factor_l$ infine si attiva quando la corsia stessa sta saturando.

Scelta del prodotto invece della somma. È importante chiarire fin da subito che le tre componenti non sono tre condizioni che una corsia deve soddisfare per ottenere il verde: il verde viene comunque assegnato, ciclo dopo ciclo, a tutti i flussi che hanno domanda. Quello che la funzione di costo determina è l'*ordine di priorità* con cui i flussi vengono serviti, in questo senso la struttura della formula è pensata per stabilire quali flussi vanno serviti *prima* degli altri.

Delle tre componenti, solo $demand_l$ può annullare il costo: se su una corsia non ci sono né veicoli fermi né veicoli in avvicinamento, il suo costo è nullo e non c'è motivo di darle la precedenza. Le altre due componenti, invece, non azzerano mai il costo: $ds_penalty_l$ parte da 1 quando la destinazione è libera, e $cong_factor_l$ vale 1 quando la corsia è sotto la soglia di saturazione. Una corsia con domanda normale e destinazione libera riceve quindi un costo pari alla sua sola domanda, ed entra regolarmente nella rotazione del verde come in qualunque controllore a costo.

La scelta di moltiplicare le tre componenti, anziché sommarle, consente a ciascun fattore di modulare il contributo degli altri. Una domanda elevata e una forte saturazione locale tendono ad aumentare il costo del flusso, mentre un'elevata occupazione delle destinazioni a valle lo riduce attraverso la penalità downstream, scoraggiando l'assegnazione del verde verso strade già congestionate. Di conseguenza, il costo assume valori elevati soprattutto quando una corsia presenta molta domanda e una condizione locale critica, ma dispone ancora di sufficiente capacità a valle. Con una somma, invece, i contributi agirebbero in modo indipendente e lineare, rendendo meno marcata la distinzione tra un flusso effettivamente prioritario e uno caratterizzato da una sola componente elevata. Il prodotto permette quindi di far emergere con maggiore evidenza le situazioni in cui è opportuno intervenire, senza favorire flussi che rischierebbero di trasferire traffico verso destinazioni già sature. I flussi ordinari continuano comunque a essere serviti in funzione della domanda.

Componente di domanda. La prima componente si calcola come

$$demand_l = J (stopped_l + \alpha approaching_l) \quad (3.2)$$

dove $stopped_l$ è il numero di veicoli attualmente fermi sulla corsia l (veicoli con velocità sotto una soglia minima) e $approaching_l$ è il conteggio dei veicoli presenti sugli archi a monte che confluiscono nell'incrocio. Il termine $\alpha approaching_l$ con $\alpha = 0.3$ rappresenta una forma di anticipazione: tiene conto del traffico in avvicinamento prima ancora che si accodi all'incrocio. Il valore di α è volutamente inferiore a 1 perché i veicoli ancora distanti dal semaforo, per cui il controllore non è ancora certo che raggiungano la corsia l (potrebbero svoltare prima o rallentare a causa del traffico), e pesarli quanto i veicoli già fermi sarebbe sovrastimare il loro contributo. Questa componente è la risposta al primo limite identificato nella sezione precedente: il semaforo non è più cieco rispetto al traffico in avvicinamento.

Componente di penalità a valle. La seconda componente si calcola come

$$ds_penalty_l = \frac{1}{(1 + \beta * occ_down_l)} \quad (3.3)$$

con $\beta = 2.0$, dove occ_down_l è l'occupazione massima tra le strade in uscita verso cui possono dirigersi i veicoli presenti sulla corsia l . L'idea di base è che se la strada in cui si manderebbe il traffico è già piena, dare il verde a quella corsia non risolve il problema ma lo sposta solo di pochi metri, con il rischio di creare *spillback*. Con i parametri scelti, la penalità vale 1 quando la destinazione è vuota e tende a $1/3$ quando è completamente piena, scoraggiando il flusso senza vietarlo del tutto. Questa componente risponde al secondo limite individuato: il semaforo non è più cieco rispetto allo stato a valle.

Componente di fattore di congestione. La terza componente si calcola come

$$cong_factor_l = \begin{cases} 1 & \text{se } occ_l \leq \theta \\ 1 + \delta (occ_l - \theta) & \text{se } occ_l > \theta \end{cases} \quad (3.4)$$

dove occ_l è l'occupazione della corsia l , $\theta = 0.6$ è la soglia oltre la quale il meccanismo si attiva e $\delta = 3.0$ ne regola la pendenza. Al di sotto della soglia il fattore vale 1 ed è quindi neutro nel prodotto, lasciando il costo determinato dalle altre due componenti; al contrario, il fattore cresce linearmente con l'occupazione: una corsia che è prossima alla saturazione vede il proprio costo amplificato in misura proporzionale a quanto eccede θ , così da indurre il semaforo a servirla prima che raggiunga la saturazione completa.

Penalità di switch (STOP_COST).

Oltre alla funzione di costo, il progetto prevede un meccanismo separato, denominato *STOP_COST*. Tale meccanismo non interviene direttamente nel calcolo del costo dei flussi, ma opera a un livello successivo, modificando la regola utilizzata per decidere il cambio di fase. In particolare, quando viene valutata la possibilità di revocare il verde al flusso attualmente servito, al relativo costo viene aggiunta una penalità pari a 500 punti per ogni veicolo che,

in quel momento, risulta effettivamente in movimento sulle corsie associate al flusso.

L'effetto è che un plotone in transito non viene spezzato solo perché un altro flusso ha intanto accumulato un costo di poco superiore. Perché lo switch avvenga, il vantaggio dell'alternativa deve essere abbastanza grande da ripagare il fatto di fermare veicoli che stavano già scorrendo. Vale la pena distinguere questo meccanismo dal K_i asimmetrico del controllore di riferimento, perché a prima vista sembrano fare la stessa cosa. Il K_i asimmetrico vive dentro la formula del costo e gonfia il peso di tutti i veicoli del flusso verde, in ogni istante. Lo STOP_COST invece non tocca la formula: interviene soltanto nel confronto finale tra i costi, nel momento in cui il semaforo sta per cambiare. Con la scelta di tenere la penalità fuori dal costo, si riesce a separare due domande che altrimenti finirebbero mescolate: quanto conta servire un certo flusso, e quanto conviene interrompere un flusso che sta già funzionando. In sintesi, la prima è una valutazione dello stato, la seconda riguarda la stabilità del controllo.

3.5 L'Adaptive Mix

La formula introdotta nella sezione precedente affronta i limiti del controllore di riferimento, ma da sola non costituisce ancora la soluzione completa proposta in questa tesi. Per poterne capire la ragione occorre una riflessione sui regimi di traffico in cui un controllore semaforico effettivamente si trova a operare.

Le reti urbane possono presentarsi in due regimi molto diversi tra loro. Il primo è quello delle griglie regolari e fluide, tipiche di alcune città di impianto moderno, come alcune zone di Manhattan: qui il traffico scorre in modo relativamente simmetrico. Il secondo è quello delle reti urbane reali, come la rete di Bologna, caratterizzate da asimmetrie, colli di bottiglia e zone che si saturano con facilità: qui la priorità si sposta sul gestire le code prima che degenerino e sul tenere conto di quanto sia congestionata ciascuna destinazione.

Nessuna funzione di costo a parametri fissi è ottima in entrambi i regimi. Il controllore di riferimento, il cui costo cresce quadraticamente con la velocità dei veicoli, è per costruzione orientato a non interrompere i flussi veloci, e si rivela quindi adatto al primo regime. La formula proposta in questo lavoro, con la sua attenzione alla domanda di traffico, alla penalità verso le destinazioni congestionate e alla saturazione locale delle corsie in ingresso, è invece adatta al secondo. Una soluzione monolitica, che adottasse sempre l'una o sempre l'altra, sarebbe inevitabilmente sub-ottimale in almeno uno dei due regimi.

L'idea progettuale alla base dell'*Adaptive Mix* consiste nel non scegliere a priori quale logica adottare, ma nel lasciare che ogni semaforo decida autonomamente in base allo stato del proprio incrocio. Il costo di ciascun flusso diventa così una combinazione convessa dei due costi:

$$Cost_f = (1 - w) \cdot Cost_f^{\text{rif}} + w \cdot Cost_f^{\text{nuova formula}} \quad (3.5)$$

dove $Cost_f^{\text{rif}}$ e $Cost_f^{\text{nuova formula}}$ sono i costi del flusso f calcolati rispettivamente con la formula del controllore di riferimento e con la formula proposta. Il peso $w \in [0, 1]$ è la media istantanea delle occupazioni di tutte le corsie in ingresso dell'incrocio:

$$w = \frac{1}{|L|} \sum_{l \in L} occ_l \quad (3.6)$$

dove L è l'insieme delle corsie in ingresso del semaforo e occ_l l'occupazione della corsia l .

Comportamento ottenuto. Quando l'incrocio è poco occupato, w resta basso e il comportamento del semaforo assomiglia a quello del controllore di riferimento, che protegge chi si muove. Quando invece l'incrocio si riempie, w cresce e prevale progressivamente la nuova formula, e il semaforo inizia a gestire le code, ad anticipare il traffico in arrivo e a tenere conto dello spazio disponibile a valle. La transizione tra i due regimi è graduale e automatica.

Locale e automatico. L'aspetto progettualmente più interessante dell'A-

daptive Mix è che il bilanciamento tra le due strategie avviene a livello di singolo semaforo, senza alcuna comunicazione con gli altri. Nella stessa rete possono coesistere incroci che si comportano quasi come il controllore di riferimento (perché poco trafficati) e incroci che usano prevalentemente la nuova formula (perché sono più carichi). Tutto questo emerge spontaneamente dal valore locale dell'occupazione, senza alcun parametro globale né alcuna forma di coordinamento.

3.6 Ipotesi del modello e sensori richiesti

Per essere applicabile in un contesto reale, il nuovo algoritmo richiede una serie di ipotesi sulla disponibilità di dati e sull'infrastruttura sensoristica installata su ogni semaforo. Queste ipotesi vanno dichiarate esplicitamente per capire cosa serve nel mondo reale per replicare i risultati ottenuti in simulazione.

3.6.1 Dati statici e di stato

L'algoritmo si basa su informazioni che normalmente sono già disponibili, o nella centralina semaforica o nei dati di progetto della rete stradale, senza richiedere sensori dedicati. In particolare, considera il piano semaforico, con fasi, durate e collegamenti, la configurazione delle corsie controllate, cioè lunghezza e numero di corsie per ogni edge, gli edge posti a monte e a valle dell'incrocio e, infine, lo stato corrente delle luci e della fase attiva.

3.6.2 Dati operativi e sensori

Le misure operative che richiedono sensori installati sull'infrastruttura stradale sono cinque e vengono descritte di seguito. Tutte vanno raccolte con frequenza di almeno una misura al secondo per ogni semaforo.

Misura 1 – Conteggio veicoli per ogni corsia in ingresso. Numero totale di veicoli presenti sulla corsia in un dato istante. Serve a calcolare l'occupazione locale e a determinare il peso w del Mix Adattivo.

Misura 2 – Conteggio veicoli fermi per ogni corsia in ingresso.

Distinzione tra veicoli in movimento e veicoli fermi (sotto soglia di velocità). Entra nel termine $stopped_l$ della componente di domanda.

Misura 3 – Conteggio veicoli per ogni edge a valle. Serve a calcolare l'occupazione downstream usata nella componente di penalità.

Misura 4 – Conteggio veicoli per ogni edge a monte. Serve a stimare i veicoli in avvicinamento ($approaching_l$).

Misura 5 – Velocità istantanea per veicolo. Necessaria per il calcolo del costo del controllore di riferimento, che entra come una delle due componenti del Mix Adattivo.

3.6.3 Tecnologie sensoristiche realizzabili

Tutte le grandezze considerate possono essere rilevate con tecnologie già disponibili sul mercato, come spire induttive multiple, sensori radar a microonde, sistemi di videocamere con *computer vision* o sensori LiDAR. La scelta della soluzione sensoristica più adatta non rientra negli obiettivi di questa tesi. Tuttavia, è utile evidenziare che le misure richieste non presuppongono l'introduzione di infrastrutture completamente nuove o particolarmente invasive. In diversi contesti urbani italiani, infatti, tecnologie di questo tipo sono già presenti, almeno in parte, per attività di monitoraggio del traffico, gestione della viabilità o alimentazione dei pannelli a messaggio variabile.

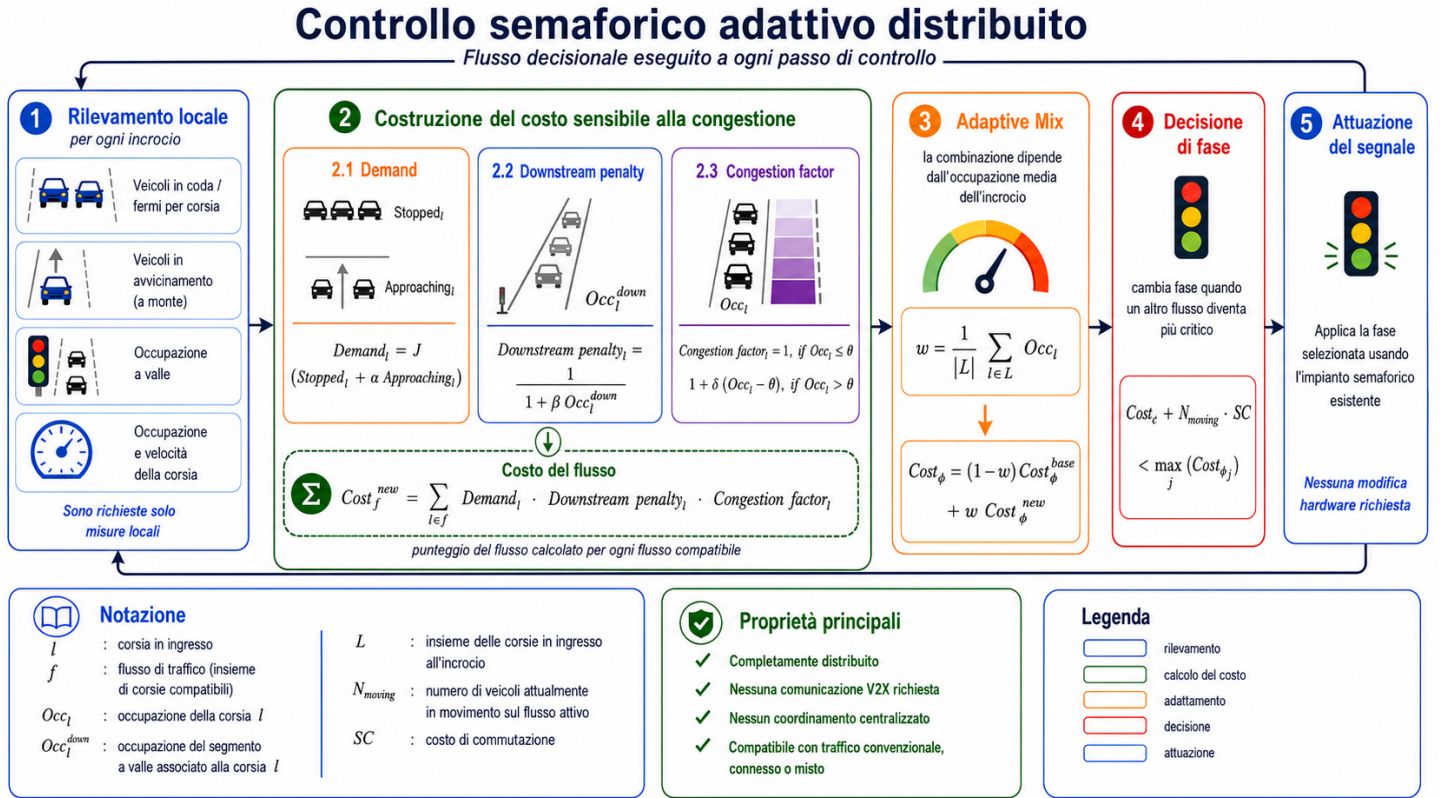


Figura 3.1: Flusso decisionale complessivo del controllore semaforico adattivo e distribuito proposto.

Implementazione

Dopo aver fissato i requisiti e definito il progetto della soluzione nel Capitolo 3, in questo capitolo viene descritta la realizzazione del progetto in un sistema funzionante. L'algoritmo è implementato in Python e si interfaccia con il simulatore SUMO tramite l'interfaccia TraCI. Il testo procede dall'architettura complessiva e dall'integrazione con il simulatore fino al dettaglio della logica: come ogni semaforo ricostruisce la struttura del proprio incrocio, quali informazioni conviene calcolare una sola volta, quali operazioni vengono eseguite a ogni passo di simulazione e come la funzione di costo del capitolo precedente si traduce in codice. Il capitolo si conclude con la descrizione delle approssimazioni adottate, della procedura di generazione delle popolazioni veicolari basata sui dati ACI 2022 e degli script di automazione utilizzati per condurre la campagna sperimentale.

4.1 Strumenti di simulazione e modellazione emissiva

Prima di esporre l'architettura complessiva, è utile introdurre brevemente gli strumenti software che sono stati impiegati nella parte sperimentale di questa tesi. Come emerge dall'analisi del capitolo 2, la quasi totalità dei lavori analizzati valuta i propri risultati tramite microsimulazione del traffico, affidandosi a simulatori come VISSIM, AIMSUN o SUMO; la scelta dello strumento di simulazione e quella del modello emissivo integrato incidono in modo significativo sull'attendibilità dei risultati; per questa ragione, entrambe devono essere dichiarate esplicitamente.

- **SUMO (Simulation of Urban MObility)** è un simulatore di traffico microscopico [LópBBW⁺18], open-source e multimodale, sviluppato dal Deutsches Zentrum für Luft- und Raumfahrt (DLR). A differenza dei

modelli macroscopici, che descrivono il traffico in termini aggregati di flusso e densità, un simulatore microscopico modella il comportamento di ogni singolo veicolo in termini di posizione, velocità, accelerazione e cambi di corsia, secondo specifici modelli di *car-following* e *lane-changing*. Questa granularità è essenziale, poiché la funzione di costo proposta opera proprio su grandezze a livello di singolo veicolo e di singola corsia, come il numero di veicoli fermi, la velocità istantanea e l'occupazione.

- **TraCI (Traffic Control Interface)** è l'interfaccia che consente a un programma esterno di interagire con una simulazione SUMO in esecuzione, avanzando la simulazione passo per passo. A ogni passo, TraCI permette sia di leggere lo stato della rete — numero di veicoli presenti su una corsia, veicoli fermi, velocità e occupazione degli archi — sia di intervenire sul controllo, ad esempio modificando la fase di un semaforo. Questo meccanismo rende possibile l'implementazione di una logica di controllo adattiva esterna al simulatore: l'algoritmo proposto, descritto nel Capitolo 3, è un programma Python che a ogni secondo di simulazione legge lo stato di ciascun incrocio tramite TraCI e decide se mantenere o cambiare la fase corrente.
- **HBEFA (Handbook Emission Factors for Road Transport)** è un database di fattori di emissione per il trasporto su strada [INF19], ampiamente utilizzato in ambito europeo, che associa a ciascuna classe di veicolo i tassi di emissione dei principali inquinanti in funzione delle condizioni di guida. SUMO integra i fattori HBEFA, nella versione 4 (HBEFA4), all'interno dei propri modelli di emissione, stimando il consumo e le emissioni di ogni veicolo a partire dalla sua dinamica istantanea (velocità e accelerazione).

4.2 Architettura generale e integrazione con SUMO

La scelta architetturale di fondo deriva direttamente dal requisito di decentralizzazione, infatti ogni semaforo della rete è governato da una propria istanza della classe, e questa conosce esclusivamente il proprio incrocio: le fasi del suo piano semaforico, le corsie in ingresso, le strade a monte e a valle. Come già anticipato non esiste una memoria condivisa tra i controllori né alcun canale di comunicazione tra di essi. Ogni decisione nasce dallo stato locale.

L'orchestrazione della simulazione è affidata a uno script separato, `RUNNER.PY`, che carica la popolazione veicolare, avvia il processo SUMO, apre la connessione TraCI e, a ogni passo temporale, invoca il metodo decisionale di ciascun controllore. Allo stesso script è affidata anche la raccolta dei dati. Al termine della simulazione, il runner interroga SUMO per ottenere, per ogni veicolo, la distanza percorsa, il tempo di percorrenza, il tempo di attesa, la velocità media e le emissioni di CO₂; tali valori vengono quindi salvati in un file CSV, che costituisce l'output sperimentale.

SUMO avanza per passi discreti della durata di un secondo. A ogni passo, il runner invoca il metodo `performStep()` su ogni controllore, ed è all'interno di tale metodo che si concentra l'intera logica dell'algoritmo adattivo: la lettura dello stato dell'incrocio, il calcolo dei costi dei flussi e la decisione se mantenere o cambiare la fase corrente.

Il sistema consente di eseguire la stessa popolazione di veicoli con tre logiche di controllo: il piano a tempi fissi originale della rete, il controllore di riferimento e il nuovo algoritmo. La logica attiva è selezionata da un parametro passato al runner. Questo dettaglio implementativo è ciò che rende possibile il confronto a parità di popolazione descritto nel Capitolo 5: il medesimo insieme di veicoli, con gli stessi tempi di partenza e gli stessi percorsi, attraversa la rete sotto le tre logiche, e le differenze nei risultati sono imputabili esclusivamente all'algoritmo e non alla variabilità della domanda.

4.3 Identificazione di fasi e flussi

Prima di poter calcolare un costo, ogni controllore deve ricostruire la struttura del proprio incrocio. Questa operazione viene eseguita una sola volta, nel costruttore.

Fasi. Un piano semaforico in SUMO è costituito da una sequenza ciclica di fasi. Ciascuna fase è caratterizzata da una durata e da una stringa di stato che indica, per ogni collegamento controllato, se in quell'istante presenta luce verde, gialla o rossa. Non tutte le fasi rappresentano una decisione di controllo: le configurazioni di giallo e di rosso completo costituiscono infatti fasi di transizione, la cui durata è stabilita dalle norme di sicurezza e non deve essere modificata. Il controllore interviene quindi esclusivamente sulle fasi principali, sufficientemente lunghe da rappresentare un'effettiva scelta di controllo, distinguendole dalle fasi di transizione.

Per ciascuna fase principale viene inoltre registrato l'indice della fase precedente. Questa informazione consente di inserire i cambi di fase nel punto corretto del ciclo, senza forzare o eliminare le transizioni intermedie.

Flussi. Per ogni fase principale viene successivamente ricostruito il relativo flusso, definito come l'insieme degli archi che ricevono contemporaneamente il verde, delle corsie di ingresso da cui provengono i veicoli e delle corsie in uscita verso cui sono diretti. Su tale struttura viene applicata la funzione di costo definita nel Capitolo 3: il costo complessivo del flusso è ottenuto sommando i contributi associati alle sue corsie di ingresso.

Il nucleo della procedura di ricostruzione è riportato nell'Algoritmo seguente.

Algoritmo 1 Ricostruzione dei flussi associati alle fasi principali del piano semaforico

```

1 for phaseIndex in self.__mainPhases:
2     movingEdges, in_lanes, out_lanes = [], [], []
3     phaseString = self.__program.phases[phaseIndex].state.lower()
4     for j, link in enumerate(controlled_links):
5         if len(link) != 0:
6             edge = link[0][0][: -2]
7             if phaseString[j] == 'g' and edge not in
movingEdges:
8                 movingEdges.append(edge)
9                 if phaseString[j] == 'g':
10                     in_lane = link[0][0]
11                     if in_lane not in in_lanes:
12                         in_lanes.append(in_lane)
13                     if len(link[0]) >= 2 and link[0][1]:
14                         out_lanes.append(link[0][1])
15     self.__flows.append(Flow(phaseIndex, movingEdges,
in_lanes, out_lanes))

```

Poiché il piano semaforico non cambia durante una simulazione, questa analisi viene effettuata una sola volta alla creazione del controllore e il risultato resta valido per l'intera durata dell'esperimento.

4.4 Pre-calcolo statico delle informazioni di rete

Una prima versione dell'implementazione interrogava SUMO a ogni passo anche per informazioni che non variano nel corso dell'esperimento, come la lunghezza di una corsia o il numero di corsie di un arco. Su reti grandi e con molti semafori questo comportava un numero elevato di chiamate TraCI ridondanti, che costituiscono la principale fonte di lentezza in questo tipo di simulazioni: il tempo non si concentra nel calcolo del costo, ma nelle richieste ripetute al simulatore di informazioni già note in partenza.

Tre tipi di informazione sono di natura statica: la capacità di ogni corsia in ingresso, stimata dalla sua lunghezza; la capacità complessiva di ogni arco in uscita, ricavata dalla lunghezza e dal numero di corsie; e l'elenco degli archi entranti nel nodo a monte di ciascuna corsia, necessario per stimare i veicoli in avvicinamento. Nessuna di queste grandezze dipende dallo stato del traffico: dipendono unicamente dalla geometria della rete.

La soluzione adottata è stata quella di spostare questi calcoli nel costruttore, così da essere eseguiti una sola volta e memorizzati in dizionari indicizzati per corsia o per arco; durante la simulazione il metodo decisionale legge tali valori dai dizionari invece che interrogare nuovamente SUMO, così da lasciare nel ciclo soltanto le chiamate strettamente necessarie, quelle che leggono lo stato dinamico del traffico: numero di veicoli, veicoli fermi, velocità. Va sottolineato un punto: questa ottimizzazione non altera in alcun modo i risultati numerici.

4.5 Il ciclo di decisione

Il metodo `performStep()`, invocato a ogni passo di simulazione, contiene la logica decisionale. Tre meccanismi agiscono in cascata, con diversi ruoli: due controlli di sicurezza che delimitano la finestra temporale entro cui la decisione di costo può operare, e la regola principale che, all'interno di questa finestra, decide se mantenere o cambiare la fase corrente.

Il primo controllo è il verde minimo. Nei primi secondi di una fase principale il controllore non valuta alcun cambio: ogni flusso che ottiene il verde lo mantiene per un tempo minimo, evitando commutazioni troppo rapide che risulterebbero inefficienti per il flusso del traffico e, in un deployment reale, problematiche dal punto di vista della sicurezza. Solo trascorso questo intervallo la logica di costo entra in funzione.

Il secondo controllo agisce all'estremo opposto: se una fase è attiva oltre un tempo massimo, il cambio viene forzato indipendentemente dai costi. Senza questo limite un flusso particolarmente carico potrebbe mantenere l'incrocio a tempo indeterminato, impedendo agli altri flussi di essere serviti e violando il requisito di equità ciclica.

Tra questi due estremi opera la regola principale. Il controllore calcola i costi di tutti i flussi, come descritto nella Sezione 4.6, e applica la regola dello `STOP_COST`, formalizzata nell'Eq. (4.1). La fase viene cambiata soltanto se il costo del flusso attualmente servito, incrementato della penalità associata ai veicoli ancora in movimento, è strettamente inferiore al massimo dei costi dei flussi alternativi.

$$Cost_{corrente} + n_{moving} \cdot STOP_COST < \max_f Cost_f \quad (4.1)$$

dove n_{moving} è il numero di veicoli in movimento sul flusso corrente.

La distanza tra la regola enunciata in astratto nel Capitolo 3 e queste poche righe è volutamente minima. È una conseguenza diretta del requisito di interpretabilità, che impone che la logica sia leggibile e verificabile riga per riga.

Algoritmo 2 Implementazione della regola dello STOP_COST definita nell'Eq. (4.1).

```
1 currentCost = currentFlow.cost if currentFlow else 0
2 maxAlternativeCost = max([f.cost for f in self.flows])
3 n_moving = 0
4 if currentFlow:
5     for edge in currentFlow.edges:
6         n_total = traci.edge.getLastStepVehicleNumber(edge)
7         n_halt = traci.edge.getLastStepHaltingNumber(edge)
8         n_moving += (n_total - n_halt)
9 if currentCost + n_moving * STOP_COST < maxAlternativeCost:
10     self.switchTrafficLight()
```

4.6 Il calcolo dei costi

Il metodo `calcFlowCosts()` implementa la funzione di costo proposta nell'Eq. (3.1) e il bilanciamento adattivo definito nell'Eq. (3.5). Per ogni flusso vengono calcolati separatamente il costo del controllore di riferimento e quello della formula proposta; i due valori vengono successivamente combinati sulla base dell'occupazione locale dell'incrocio.

Il peso che governa la combinazione è w , utilizzato nell'Eq. (3.5) e calcolato come media istantanea delle occupazioni di tutte le corsie in ingresso dell'incrocio. Il valore viene ricalcolato all'inizio di ogni valutazione, poiché varia insieme allo stato del traffico e determina, per ciascun semaforo e in ciascun istante, il peso assegnato alle due strategie.

Il costo della formula proposta viene calcolato sommando, sulle corsie in ingresso del flusso, il prodotto delle tre componenti definite nel capitolo precedente. La domanda dell'Eq. (3.2) combina i veicoli fermi e una frazione α di quelli in avvicinamento dai tratti a monte. La penalità a valle è definita nell'Eq. (3.3) e dipende dall'occupazione delle destinazioni. Il fattore di congestione dell'Eq. (3.4) resta pari a 1 finché l'occupazione locale non supera la soglia θ e, oltre tale soglia, cresce linearmente con la differenza. Il prodotto delle tre componenti viene infine sommato sulle corsie del flusso,

come previsto dalla funzione di costo dell'Eq. (3.1).

Algoritmo 3 Implementazione della funzione di costo proposta nell'Eq. (3.1).

```
1 downstream_penalty = 1.0 / (1.0 + BETA * ds_occ_max)
2 demand = J * (stopped + ALPHA * approaching)
3 congestion_factor = 1.0
4 if local_occ > THRESHOLD:
5     congestion_factor = 1.0 + DELTA * (local_occ -
        THRESHOLD)
6 new_formula_cost += demand * downstream_penalty *
    congestion_factor
```

Ottenuti i due costi, quello finale del flusso è la loro combinazione pesata da w : Anche in questo caso una sola riga realizza il bilanciamento adattivo

Algoritmo 4 Implementazione dell'Adaptive Mix definito nell'Eq. (3.5).

```
1 flow.cost = (1.0 - w) * formula_cost_rif + w *
    new_formula_cost
```

locale tra le due strategie, senza alcun parametro globale e senza coordinamento tra semafori: ogni incrocio determina autonomamente quale peso attribuire a ciascuna delle due strategie in base alla propria occupazione corrente.

4.7 Approssimazioni e scelte tecniche

All'interno dell'implementazione vengono introdotte alcune approssimazioni che però vanno dichiarate esplicitamente, poiché incidono sul comportamento del sistema pur non alterandone la logica di fondo.

La prima riguarda la **velocità** nel costo del controllore di riferimento, che dipende dal quadrato della velocità dei veicoli. Un valore anomalo, dovuto a un transitorio della simulazione, produrrebbe un costo sproporzionato. Per

evitarlo, la velocità che entra nel termine quadratico è limitata superiormente a una soglia. Nei casi ordinari il limite non interviene; serve unicamente a rendere il calcolo robusto rispetto a valori estremi che non corrispondono a situazioni di traffico reali.

La seconda riguarda la **capacità delle corsie**, stimata dividendo la lunghezza per lo spazio medio occupato da un veicolo in coda. Si tratta di un'approssimazione, poiché la capacità reale dipende anche dalla composizione del traffico e dallo stile di guida. Su corsie molto corte questo rapporto può degenerare, perciò la capacità è comunque vincolata a un valore minimo. L'approssimazione è accettabile perché la funzione di costo non utilizza mai la capacità come valore assoluto, ma soltanto per calcolare occupazioni relative, cioè rapporti compresi tra zero e uno.

La terza riguarda la gestione delle **eccezioni**. Le chiamate a TraCI possono occasionalmente fallire, in genere perché un elemento della rete si trova in uno stato transitorio e momentaneamente non è interrogabile. Per evitare che il fallimento di una singola lettura interrompa la simulazione o perfino alteri in modo sensibile il costo del flusso, il codice è programmato in modo da intercettare questi casi e utilizza dei valori conservativi. Questa soluzione privilegia la robustezza dell'esecuzione durante campagne sperimentali prolungate rispetto alla precisione della singola misura istantanea; la scelta è giustificata dalla durata complessiva degli esperimenti.

4.8 Generazione delle popolazioni veicolari

Per procedere con la fase sperimentale, ogni configurazione richiede una popolazione, ovvero l'insieme dei veicoli che entrano nella rete, con i rispettivi tempi di partenza, percorsi e parametri.

Le tipologie di veicolo presenti (autovetture, motocicli, veicoli commerciali leggeri, mezzi pesanti) e le relative classi di alimentazione ed emissione sono campionate riproducendo la composizione del parco circolante italiano secondo i dati ACI 2022. Questo aspetto è rilevante, poiché implica che le emissioni di CO₂ calcolate mediante il modello HBEFA4 si riferiscono a

una flotta realistica e non a una composizione arbitraria, rendendo i risultati rappresentativi del contesto italiano a cui la tesi si rivolge.

Per ogni veicolo i parametri fisici (lunghezza, massa, accelerazione, decelerazione, velocità massima) e quelli comportamentali del conducente sono poi campionati da distribuzioni realistiche. Per il profilo del guidatore si introduce una variabile latente di aggressività, da cui dipendono il tempo di reazione e il fattore di rispetto del limite di velocità, mentre l'imperfezione di guida è campionata indipendentemente. Questa scelta riproduce un fenomeno realistico: gli stili di guida non variano in modo casuale, ma in modo coerente con il temperamento del guidatore.

Un ultimo punto è di natura metodologica più che implementativa. Una volta generata, una popolazione viene salvata su file e riutilizzata identica per le tre logiche di controllo, così le differenze che emergono sono misurate sulla medesima sequenza di veicoli e di conseguenza la variabilità dovuta alla casualità della domanda viene eliminata e ciò che resta è attribuibile esclusivamente alla logica di controllo.

4.9 Automazione degli esperimenti

La campagna sperimentale richiede un numero elevato di simulazioni: per ogni rete, per ogni densità e per ogni ripetizione, devono essere eseguite tre simulazioni, una per logica di controllo. Una gestione manuale sarebbe impraticabile e soggetta a errori, perciò è stato sviluppato uno script di automazione.

Lo script genera le popolazioni mancanti, lancia le simulazioni necessarie, raccoglie i file CSV prodotti e li archivia in una struttura di cartelle organizzata per rete, densità e logica di controllo. Inoltre è incrementale: prima di lanciare una simulazione verifica se il risultato corrispondente è già presente e, in tal caso, la salta. Questa proprietà non è secondaria: data la durata di una campagna completa, poterla interrompere e riprendere senza ripetere il lavoro già svolto è ciò che la rende effettivamente eseguibile.

Per sfruttare le macchine multicore lo script esegue più simulazioni in parallelo. Il grado di parallelismo è configurabile e va tarato sul numero di

core e sulla memoria disponibile: un valore troppo basso spreca risorse, un valore troppo alto satura la macchina e rallenta tutte le esecuzioni simultaneamente. Gli avvii delle simulazioni parallele sono distanziati nel tempo e gli eventuali errori di ciascuna esecuzione vengono registrati separatamente, in modo da individuare e ripetere automaticamente le sole simulazioni eventualmente fallite anziché accorgersene a posteriori.

L'effetto complessivo è la riproducibilità. A partire dalle stesse popolazioni e dagli stessi script, i risultati presentati nel Capitolo 5 possono essere rigenerati senza interventi manuali, eliminando alla radice la categoria di errori che deriva dall'eseguire a mano la stessa procedura su centinaia di simulazioni.

Esperimenti

In questo capitolo vengono presentati i risultati sperimentali ottenuti applicando il nuovo algoritmo alle quattro reti considerate nella campagna sperimentale.

La prima parte del capitolo è dedicata al setup sperimentale, quindi alla descrizione delle reti utilizzate, delle diverse densità di traffico, della metodologia di confronto e delle metriche considerate. Successivamente vengono analizzati i risultati ottenuti su ciascuna rete, confrontando il nuovo algoritmo sia con il controllore di riferimento sia con il piano a tempi fissi.

A partire dai risultati raccolti viene inoltre discusso il compromesso tra riduzione delle emissioni e variazione dei tempi di percorrenza. Infine, il capitolo considera anche uno scenario di *deployment* incrementale, nel quale l'algoritmo viene attivato solo su una parte degli incroci della rete.

5.1 Setup sperimentale

Reti analizzate. Gli esperimenti sono stati condotti su quattro reti, scelte per coprire situazioni di traffico molto diverse tra loro. La mappa di **Bologna** è una porzione di rete urbana reale, con strade asimmetriche e incroci eterogenei. Le reti **Manhattan 8×8** e **Manhattan 15×15** sono griglie regolari sintetiche, la prima di dimensione contenuta e la seconda più estesa, utilizzate per valutare il comportamento dell'algoritmo in condizioni topologicamente ordinate e per studiarne la scalabilità. La rete **Masa** è di piccole dimensioni; nello specifico è una porzione della città di Modena, impiegata come ulteriore caso di validazione. Le caratteristiche strutturali delle quattro reti (numero di incroci semaforizzati, numero di archi) sono riassunte nella Tabella 5.1.

Tabella 5.1: Caratteristiche strutturali delle reti analizzate.

Rete	Incroci semaforici	Archi
Bologna	228	10281
Manhattan 8×8	60	224
Manhattan 15×15	225	840
Masa	6	937

Confronto a parità di popolazione. Come anticipato nel Capitolo 4, ogni popolazione di veicoli, una volta generata, è stata riutilizzata identica per le tre logiche di controllo: il piano a tempi fissi originale della rete (indicato come *Fisso*), il controllore di riferimento e il nuovo algoritmo proposto. Poiché i tre algoritmi operano sul medesimo insieme di veicoli, con gli stessi tempi di partenza e gli stessi percorsi, le differenze osservate sono attribuibili esclusivamente alla logica di controllo e non alla variabilità della domanda.

Ripetizioni e significatività statistica. Per ogni combinazione di rete e densità sono state generate tra sette e dieci popolazioni indipendenti, ciascuna con un campionamento casuale diverso. Per ciascuna configurazione si riportano le variazioni percentuali medie delle tre metriche. Per le emissioni di CO₂ viene inoltre indicato l'intervallo di confidenza al 95%, calcolato mediante la distribuzione t di Student.

Metriche. Le tre metriche analizzate sono le **emissioni totali di CO₂** (somma sui veicoli validi, in grammi), il **tempo di attesa medio** per veicolo e il **tempo di percorrenza medio per veicolo**. Sono considerati validi i veicoli che hanno effettivamente percorso una distanza non nulla e prodotto emissioni positive, escludendo così i veicoli che non sono mai entrati in circolazione. L'obiettivo primario della tesi è la riduzione della CO₂; le metriche temporali sono riportate per valutare l'effetto complessivo dell'algoritmo sul servizio offerto agli utenti.



Figura 5.1: Rete stradale di Bologna utilizzata per la valutazione sperimentale nel simulatore SUMO.

5.2 Risultati su Bologna

Bologna è la rete su cui si concentra l'analisi principale, essendo l'unica rete urbana reale tra quelle considerate. La Tabella 5.2 riporta il confronto tra il nuovo algoritmo e il controllore di riferimento, la Tabella 5.3 il confronto tra il nuovo algoritmo e il piano a tempi fissi.

Tabella 5.2: Bologna — nuovo algoritmo rispetto al controllore di riferimento. Valori percentuali: negativo indica miglioramento.

VPH	CO ₂ %	IC 95%	Attesa %	Percorr. %
2235	−0.74	[−2.55; +1.08]	−1.97	−0.93
3350	−4.64	[−8.04; −1.24]	−13.32	−12.75
4470	−6.84	[−9.91; −3.77]	−12.49	−21.06
5585	−5.23	[−12.93; +2.48]	−6.87	−12.60
6700	−2.70	[−7.17; +1.76]	−2.26	−7.23

Tabella 5.3: Bologna — nuovo algoritmo rispetto al piano a tempi fissi (Fisso). Valori percentuali: negativo indica miglioramento.

VPH	CO ₂ %	IC 95%	Attesa %	Percorr. %
2235	+0.32	[−3.25; +3.89]	+55.91	+28.03
3350	−1.49	[−4.09; +1.12]	+23.02	+16.69
4470	−3.19	[−7.95; +1.57]	+16.39	+10.72
5585	−1.08	[−6.16; +4.00]	+22.14	+20.55
6700	+2.87	[−0.56; +6.30]	+25.65	+32.12

Rispetto all'algoritmo di riferimento (Tabella 5.2), il nuovo algoritmo migliora le emissioni di CO₂, con un picco di −6.84% a 4470, accompagnato da una sensibile riduzione anche dei tempi di percorrenza (−21.06%); al contrario, rispetto al piano a tempi fissi (Tabella 5.3), il guadagno sulla CO₂ è più contenuto e raggiunge il massimo (−3.19%), con intervalli di confidenza che includono lo zero a tutte le densità.

Nel confronto con il piano a tempi fissi, le metriche temporali mostrano invece un peggioramento consistente: il fenomeno è riconducibile al comportamento dell'algoritmo sui numerosi incroci secondari della rete, dove la

bassa occupazione delle corsie in ingresso riduce il peso della componente di contesto e la funzione di costo si appiattisce sulla sola componente locale, generando commutazioni guidate da differenze minime tra le fasi.

5.3 Risultati su Manhattan 15×15

La rete Manhattan 15×15 rappresenta la versione più estesa della topologia a griglia regolare e consente di valutare la scalabilità dell’algoritmo su una rete con un numero maggiore di incroci e percorsi.

Tabella 5.4: Manhattan 15 x 15 — nuovo algoritmo rispetto al controllore di riferimento.

VPH	CO₂ %	IC 95%	Attesa %	Percorr. %
5585	+0.17	[−0.02; +0.35]	+3.39	+0.16
6700	+0.15	[−0.65; +0.96]	+2.51	+0.26
16800	−0.63	[−1.04; −0.21]	−11.07	−3.16
21000	−1.45	[−2.69; −0.21]	−15.66	−5.33
25200	−2.95	[−5.19; −0.70]	−16.30	−7.91

Tabella 5.5: Manhattan 15 x 15 — nuovo algoritmo rispetto al piano a tempi fissi (Fisso).

VPH	CO₂ %	IC 95%	Attesa %	Percorr. %
5585	−24.48	[−24.66; −24.30]	−91.46	−42.11
6700	−22.88	[−23.57; −22.18]	−88.95	−40.69
16800	−12.93	[−13.42; −12.43]	−67.51	−29.21
21000	−11.08	[−11.40; −10.76]	−58.12	−25.05
25200	−13.19	[−15.22; −11.16]	−50.27	−26.66

I risultati in Tabella 5.4 mostrano che il nuovo algoritmo ha un comportamento quasi identico a basso carico. Alle densità di 5585 e 6700 vph, infatti, le variazioni delle emissioni di CO₂ restano inferiori allo 0.3%, come già visto anche nella griglia 8 × 8. Quando la densità aumenta, invece, il vantaggio diventa più evidente. A 25200 vph, la riduzione di CO₂ arriva al −2.95%, con una diminuzione dei tempi di attesa pari al −16.30%. Questo

andamento dipende dal fatto che, quando il traffico aumenta, le corsie iniziano a riempirsi di più e la componente di contesto incide maggiormente sulla scelta della fase. In queste condizioni, i meccanismi pensati per limitare lo spillback e gestire la saturazione locale riescono quindi a dare un contributo più concreto rispetto al solo controllore adattivo. Osservando la Tabella 5.5 per il confronto con il piano a tempi fissi, si nota invece un miglioramento netto in tutte le densità. Le riduzioni di CO_2 risultano comprese tra l'11.08% e il 24.48%, mentre i tempi di attesa diminuiscono di almeno il 50.27% in tutte le configurazioni.

5.4 Risultati su Manhattan 8×8

La rete Manhattan 8×8 è una griglia regolare di dimensione contenuta. La sua simmetria la rende un caso utile per osservare il comportamento dell'algoritmo in assenza delle asimmetrie tipiche delle reti reali.

Tabella 5.6: Manhattan 8×8 — nuovo algoritmo rispetto al controllore di riferimento.

VPH	CO_2 %	IC 95%	Attesa %	Percorr. %
2235	+0.08	$[-0.12; +0.29]$	+2.02	+0.08
3350	+0.24	$[+0.04; +0.44]$	+2.15	+0.17
4470	+0.09	$[-0.23; +0.41]$	+1.14	+0.10
5585	+0.18	$[-0.11; +0.46]$	-0.35	-0.12
6700	+0.10	$[-0.26; +0.46]$	-1.25	-0.33
8940	-0.10	$[-0.21; +0.01]$	-3.82	-1.14

Rispetto al precedente controllore, i risultati riportati in Tabella 5.6 mostrano differenze molto contenute per tutte le densità considerate. Le variazioni di CO_2 restano infatti inferiori allo 0.3% in valore assoluto, mentre i tempi risultano sostanzialmente invariati. Questo comportamento può essere spiegato dalla struttura regolare della griglia: la simmetria della rete e l'uniformità degli incroci fanno sì che l'occupazione delle corsie in ingresso rimanga generalmente bassa. Di conseguenza, la componente di contesto incide poco sulla

Tabella 5.7: Manhattan 8 x 8 — nuovo algoritmo rispetto al piano a tempi fissi (Fisso).

VPH	CO ₂ %	IC 95%	Attesa %	Percorr. %
2235	-25.43	[-25.77; -25.08]	-93.37	-44.13
3350	-22.12	[-22.33; -21.91]	-87.99	-40.99
4470	-19.49	[-19.76; -19.21]	-81.94	-37.94
5585	-17.42	[-17.92; -16.91]	-76.52	-35.43
6700	-15.49	[-15.83; -15.16]	-71.18	-32.84
8940	-12.95	[-13.22; -12.68]	-62.06	-29.04

scelta delle fasi e la funzione di costo del nuovo algoritmo tende ad avvicinarsi a quella del controllore di riferimento. Il confronto con il piano a tempi fissi, riportato in Tabella 5.7, evidenzia invece un miglioramento più marcato. In questo caso il nuovo algoritmo ottiene riduzioni di CO₂ comprese tra il -25.43% a basso carico e il -12.95% in condizioni differenti, insieme a una riduzione dei tempi di attesa superiore al 60% in tutte le densità analizzate.

5.5 Risultati su Masa

La rete Masa, di piccole dimensioni, costituisce un ulteriore caso di validazione su una topologia diversa dalle precedenti.

Tabella 5.8: Masa — nuovo algoritmo rispetto al controllore di riferimento.

VPH	CO ₂ %	IC 95%	Attesa %	Percorr. %
457	-0.22	[-0.40; -0.04]	-25.71	-0.61
686	+0.11	[-0.05; +0.27]	-14.31	-0.36
915	-0.08	[-0.20; +0.05]	-7.50	-0.10
1372	-0.06	[-0.24; +0.11]	-7.84	-0.26
2745	+0.39	[-0.55; +1.33]	+2.93	+1.08

Rispetto al controllore di riferimento, riportato in Tabella 5.8, il nuovo algoritmo produce risultati molto simili in tutte le densità testate. Le variazioni di CO₂ rimangono infatti inferiori allo 0.4% in valore assoluto. Questo dipende anche dalla dimensione ridotta della rete: le corsie in ingresso non

Tabella 5.9: Masa — nuovo algoritmo rispetto al piano a tempi fissi (Fisso).

VPH	CO ₂ %	IC 95%	Attesa %	Percorr. %
457	-3.50	[-4.29; -2.71]	-83.03	-8.89
686	-3.33	[-3.87; -2.80]	-83.88	-9.08
915	-3.40	[-3.68; -3.12]	-83.32	-9.10
1372	-3.53	[-3.81; -3.25]	-81.06	-9.80
2745	-9.77	[-14.12; -5.42]	-72.11	-24.75

arrivano a livelli di occupazione particolarmente alti e, di conseguenza, la componente di contesto non pesa abbastanza da modificare in modo evidente il comportamento rispetto al controllore adattivo. Il confronto con il piano a tempi fissi, mostrato in Tabella 5.9, è invece più chiaro. In questo caso il nuovo algoritmo riduce la CO₂ in modo stabile, attorno al -3.4%, per le densità fino a 1372 vph, con un miglioramento che cresce fino al -9.77% a 2745 vph. Anche i tempi di attesa diminuiscono in modo marcato, con riduzioni superiori al 70% in tutte le densità analizzate.

Per sintetizzare in un'unica vista i risultati a copertura completa, la Figura 5.2 riporta la variazione di CO₂ rispetto al piano fisso per tutte e quattro le reti, in funzione della densità di traffico. Le barre verticali indicano l'intervallo di confidenza al 95%.

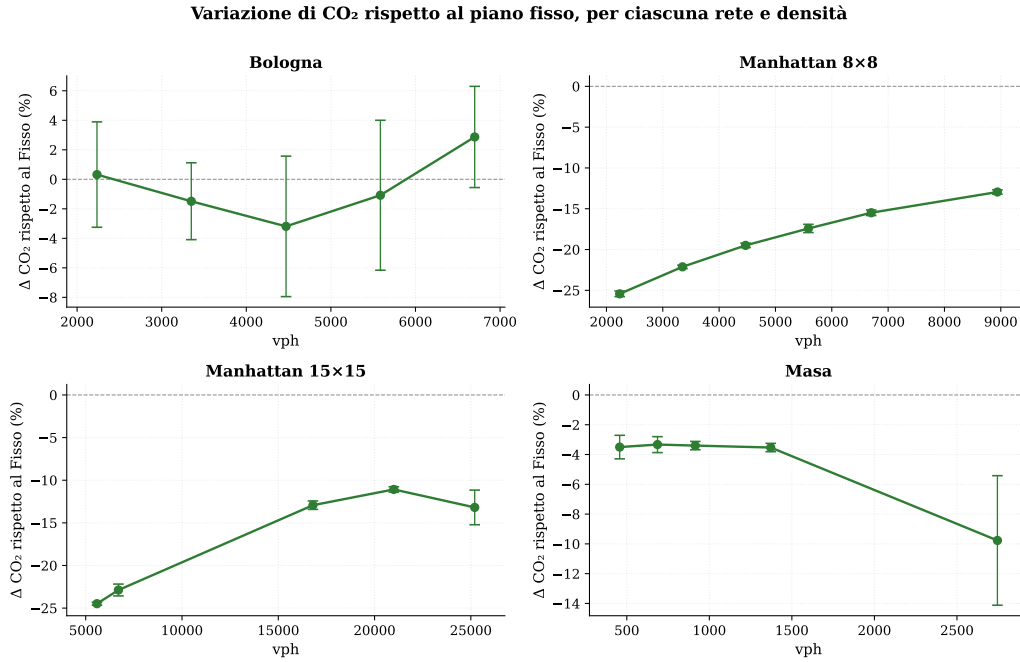


Figura 5.2: Variazione di CO₂ rispetto al piano fisso, per ciascuna rete e densità.

5.6 Deployment parziale

Uno degli aspetti pratici più rilevanti per un'amministrazione cittadina è la possibilità di equipaggiare con il nuovo algoritmo solo una parte degli incroci, anziché l'intera rete in una sola volta. Il requisito di decentralizzazione fissato nel Capitolo 3 rende questo scenario tecnicamente possibile, dato che ogni semaforo decide in modo indipendente.

In questa sezione ne viene valutata l'efficacia.

Configurazione dell'esperimento. L'esperimento è stato condotto sulle due reti su cui la copertura completa produce risultati di natura opposta: la rete urbana reale di Bologna, dove il vantaggio rispetto al piano fisso è modesto, e la griglia regolare Manhattan 15×15 , dove il vantaggio è ampio e sistematico. Per ciascuna rete si è scelta una densità di riferimento (vph 4470 per Bologna, vph 16800 per Manhattan). Una frazione crescente degli

incroci (10%, 25%, 50%) è stata equipaggiata con il nuovo algoritmo, selezionando ogni volta gli incroci a maggiore traffico (i *nodi critici* della rete), mentre i restanti incroci hanno mantenuto il piano a tempi fissi originale. La riga 100% corrisponde al risultato a piena copertura discusso nelle Sezioni 5.2 e 5.3 e serve da riferimento.

Tabella 5.10: Bologna — nuovo algoritmo rispetto al piano a tempi fissi, al variare della frazione di incroci equipaggiati.

% incroci	CO ₂ %	IC 95%	Attesa %	Percorr. %
10%	-6.40	[-10.48; -2.33]	-9.54	-12.73
25%	-8.17	[-10.90; -5.43]	-6.58	-12.99
50%	-7.99	[-14.39; -1.58]	-3.27	-0.20
100%	-3.19	[-7.95; +1.57]	+16.39	+10.72

Tabella 5.11: Manhattan 15 x 15 — nuovo algoritmo rispetto al piano a tempi fissi, al variare della frazione di incroci equipaggiati.

% incroci	CO ₂ %	IC 95%	Attesa %	Percorr. %
10%	-0.57	[-1.49; +0.35]	-8.56	-2.58
25%	-3.16	[-4.03; -2.28]	-23.85	-9.09
50%	-7.07	[-7.73; -6.40]	-44.75	-18.36
100%	-12.93	[-13.42; -12.43]	-67.51	-29.21

I risultati mettono in luce due comportamenti opposti, ciascuno coerente con la topologia della rete su cui è osservato. Su Bologna (Tabella 5.10) l'efficacia del deployment presenta un andamento non monotono e raggiunge il massimo al 25% di copertura (-8.17% di CO_2 , con intervallo di confidenza nettamente separato dallo zero), per poi ridiscendere fino al -3.19% della copertura completa. Equipaggiare oltre il quarto degli incroci a maggiore traffico non porta beneficio aggiuntivo e, anzi, lo erode. Il dato più rilevante è il comportamento delle metriche temporali: con una copertura del 10% o del 25% esse migliorano simultaneamente alla CO_2 (attesa -6.58% e percorrenza -12.99% al 25%), mentre nella copertura completa si invertono e peggiorano sensibilmente ($+16.4\%$ e $+10.7\%$). La spiegazione è coerente con quanto già discusso nella Sezione 5.2: sui nodi secondari poco trafficati la componente di contesto del nuovo algoritmo non si attiva e la funzione di costo si appiattisce sulla sola componente locale, generando commutazioni che peggiorano il quadro complessivo. Equipaggiare un numero ridotto di incroci, selezionati tra quelli a maggiore traffico, sembra limitare tali commutazioni ed è associato a un miglioramento simultaneo delle tre metriche nella stessa direzione.

Su Manhattan 15×15 (Tabella 5.11) il comportamento è più regolare. Il miglioramento aumenta man mano che cresce la percentuale di incroci controllati dall'algoritmo: si passa dal -0.57% con il 10% di copertura, valore molto vicino al piano fisso, fino al -12.93% con la copertura completa.

Lo stesso andamento si vede anche sui tempi. I tempi di attesa, infatti, passano da una riduzione del -8.56% al -67.51% . Questo risultato è coerente con la struttura della griglia: gli incroci sono abbastanza simili tra loro e ciascuno contribuisce in modo comparabile al risultato finale. Di conseguenza, se si riduce il numero di nodi equipaggiati, si perde anche una parte quasi proporzionale del beneficio.

La differenza tra le due reti chiarisce un principio operativo di carattere generale: la strategia di deployment ottimale non è universale, ma dipende dalla geometria della rete. Sulle reti regolari, dove la distribuzione del traffi-

co è omogenea, la copertura completa è preferibile e ogni incrocio aggiuntivo contribuisce al risultato. Sulle reti urbane reali, caratterizzate da pochi nodi critici e molti nodi secondari, conviene invece concentrare l'equipaggiamento sui soli nodi a maggiore traffico, ottenendo il massimo beneficio con la minima copertura. Su Bologna, in particolare, equipaggiare solo il 25% degli incroci a maggiore traffico produce simultaneamente una riduzione di CO₂ circa due volte e mezzo superiore a quella ottenibile con la copertura completa e un miglioramento netto dei tempi di attesa e di percorrenza, che con la copertura completa risultano invece peggiorati.

La Figura 5.3 riporta in forma grafica i risultati delle Tabelle 5.10 e 5.11, evidenziando il contrasto tra l'andamento osservato sulle due reti.

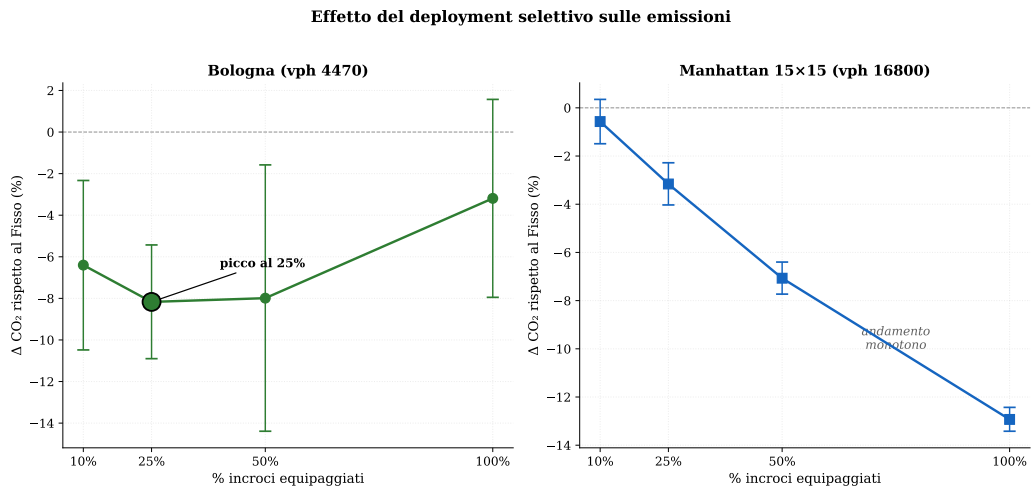


Figura 5.3: Variazione di CO₂ rispetto al piano fisso, con deployment parziale.

5.7 Scomposizione del tempo di viaggio rispetto al tempo ideale

Per inquadrare meglio il significato dei tempi misurati nelle simulazioni, conviene confrontarli con il tempo *ideale* che ciascun veicolo impiegherebbe a percorrere la propria route in assenza di semafori e di altri veicoli. Il tempo ideale di ciascun veicolo è stato calcolato sommando, lungo gli edge della sua route, il rapporto tra lunghezza dell'edge e velocità libera dell'edge (campi `length` e `maxSpeed` del file di rete `bologna_net.xml`). I file di popolazione `vp_bologna_vph4470_runN.yaml` forniscono il *routeID* assegnato a ciascuno dei 4 469 veicoli, mentre `bologna_rou.xml` fornisce la sequenza esatta di edge per ognuna delle 214 route definite.

Il tempo di viaggio reale, misurato in simulazione, si scompone così in tre componenti:

- **Tempo ideale:** il tempo che il veicolo impiegherebbe a velocità libera lungo la propria route.
- **Tempo perso ai semafori (wait):** il tempo passato fermi al rosso.
- **Tempo perso in coda (delay residuo):** il tempo passato a velocità ridotta per traffico, accodamenti e micro-decelerazioni; cioè il tempo perso rispetto alla velocità libera che non sia spiegato dal wait.

La Tabella 5.12 riporta la scomposizione per il punto di progetto di Bologna (4470), confrontando il piano a tempi fissi con le due configurazioni del nuovo algoritmo.

Tabella 5.12: Bologna 4 470 — scomposizione del tempo medio di viaggio per veicolo nelle tre componenti.

Algoritmo	Tempo ideale	Wait	Tempo in coda	Tempo totale
Fisso	2' 27"	10' 35"	6' 02"	19' 04"
Algoritmo adattivo	2' 27"	12' 19"	6' 22"	21' 08"
Algoritmo adattivo top-25%	2' 27"	9' 35"	4' 12"	16' 14"

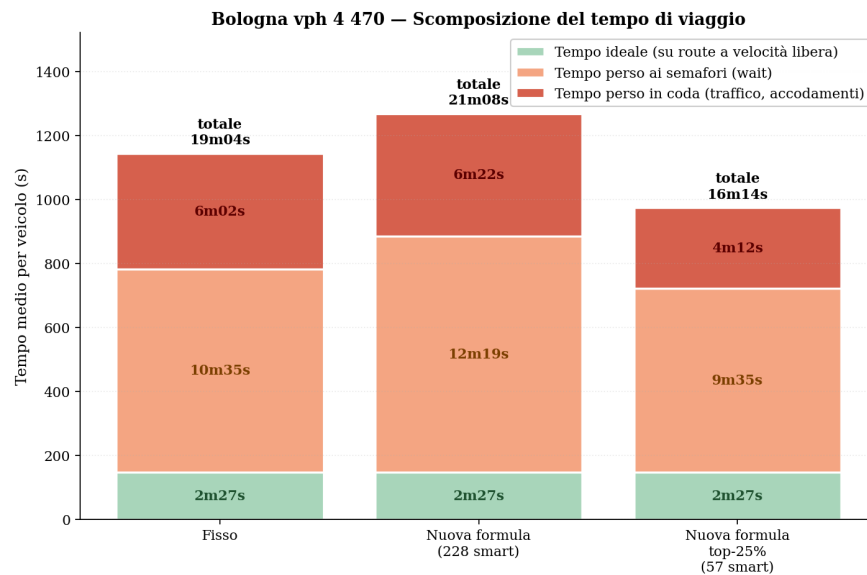


Figura 5.4: Bologna vph 4470: scomposizione del tempo medio di viaggio in tre componenti.

Il tempo ideale è identico per tutti i casi (2m 27s): è una proprietà delle route assegnate ai veicoli, che non dipende dall'algoritmo di controllo. Nei tre scenari testati corrisponde solo a circa il 13% del tempo totale; il restante 87% è "tempo perso" rispetto al cammino ideale, distribuito tra wait ai semafori (9–12 minuti) e delay residuo per traffico (4–6 minuti).

Confrontando le tre configurazioni, il piano a tempi fissi accumula un delay residuo di 6m 02s, che rappresenta il costo strutturale della congestione

della rete al punto di progetto, dovuto alle interazioni tra veicoli indipendentemente dalla logica del controllo semaforico. Sotto la versione standard del nuovo algoritmo il wait sale a 12m 19s (1m 44s in più rispetto al Fisso) e il delay residuo cresce leggermente a 6m 22s. Il bilancio sul tempo totale è negativo: il veicolo medio impiega 2 minuti in più rispetto al Fisso. Il fenomeno è coerente con quanto già discusso nella Sezione 5.2: applicando la formula adattiva anche sui 170 nodi secondari, dove il peso della componente di contesto è ridotto, le commutazioni decise da micro-differenze tra le fasi fermano i veicoli senza beneficio compensativo.

Sotto la variante top-25%, entrambe le componenti del delay si riducono simultaneamente. Il wait scende a 9m 35s (1m 00s in meno del Fisso) e il delay residuo crolla da 6m 02s a 4m 12s. Il tempo totale di viaggio si abbassa da 19m 04s a 16m 14s, con un risparmio di 2 minuti e 50 secondi per veicolo.

Conclusioni

Il lavoro svolto in questa tesi nasce da un problema concreto: nelle città, la gestione del traffico agli incroci continua a incidere in modo significativo sia sulle emissioni di CO₂, sia sui tempi di attesa degli utenti. I semafori sono quindi un elemento apparentemente semplice, ma ancora molto rilevante, su cui è possibile intervenire senza dover modificare in modo radicale l'infrastruttura urbana esistente.

Per questo motivo la scelta è ricaduta su controllori adattivi rule-based, decentralizzati e applicabili alle infrastrutture già presenti. Non si tratta degli approcci più sofisticati in assoluto, ma proprio questa caratteristica li rende interessanti in un'ottica applicativa: sono interpretabili, non richiedono una fase di addestramento su dati storici, non dipendono da un coordinamento centralizzato e non assumono la presenza di comunicazioni veicolo-infrastruttura. In altre parole, rappresentano una soluzione maggiormente compatibile con un possibile trasferimento in un contesto urbano reale.

L'obiettivo principale era intervenire su due limiti del controllore di riferimento sviluppato nel gruppo di ricerca. Il primo riguardava il traffico in avvicinamento. Il controllore, infatti, basava le proprie decisioni sui veicoli già fermi sulla corsia locale, senza considerare i plotoni in arrivo dalle strade a monte. Il secondo limite riguardava invece la congestione a valle: in alcune situazioni il verde poteva essere assegnato verso destinazioni già molto occupate, aumentando il rischio di spillback.

6.1 Sintesi del lavoro e dei risultati

Per affrontare questi aspetti è stata introdotta una nuova funzione di costo. L'obiettivo non era sostituire completamente il controllore di partenza, ma integrarlo con informazioni precedentemente non considerate. La formula tiene quindi conto della domanda di traffico in avvicinamento, dell'occupazione delle destinazioni a valle e della saturazione delle corsie in ingresso.

A questa funzione è stato affiancato l'*Adaptive Mix*, che rappresenta uno degli elementi più importanti del lavoro. Il suo scopo è evitare una scelta rigida tra due comportamenti opposti: da un lato il controllore di riferimento, più orientato a non interrompere i flussi veloci; dall'altro la nuova logica, più attenta alla gestione delle code e dello spillback. Il peso assegnato alle due logiche cambia automaticamente in base all'occupazione osservata localmente.

Questo permette di ottenere un comportamento più flessibile. In una stessa rete possono esserci incroci che continuano a comportarsi quasi come il controllore originale e altri che, trovandosi in condizioni più critiche, danno maggiore importanza alla nuova funzione di costo. Tutto questo avviene senza coordinamento centrale e senza scambio di informazioni tra semafori.

La validazione è stata condotta a parità di popolazione veicolare su quattro reti: la rete urbana reale di Bologna, le griglie regolari Manhattan 8×8 e 15×15 e la rete Masa. Le emissioni sono state stimate tramite un modello realistico, utilizzando popolazioni veicolari rappresentative del parco circolante italiano.

Nel complesso, i risultati sono coerenti con il comportamento atteso. Il nuovo algoritmo porta benefici soprattutto nei casi in cui il contesto della rete diventa più importante. Questo avviene sulla rete di Bologna lungo tutto l'intervallo di carico considerato e, sulle griglie regolari, in modo sempre più evidente all'aumentare della densità del traffico. Nei casi a basso carico, invece, il comportamento tende naturalmente ad avvicinarsi a quello del controllore di riferimento. L'occupazione rimane bassa, le code sono meno critiche e l'*Adaptive Mix* assegna quindi un peso ridotto alla nuova logica.

Il confronto con il piano a tempi fissi conferma ulteriormente questa ten-

denza. Sulle griglie regolari i miglioramenti sono ampi e sistematici, mentre sulla rete reale di Bologna risultano più contenuti. Questo non è sorprendente: una rete urbana reale è meno uniforme, presenta incroci con ruoli molto diversi e non sempre reagisce come una griglia regolare, dove il contributo dei singoli nodi è più omogeneo.

Il risultato più interessante riguarda però il deployment parziale. La possibilità di equipaggiare solo una parte degli incroci ha mostrato che il beneficio non dipende soltanto dalla percentuale di copertura, ma anche dalla struttura della rete. Sulle griglie regolari il miglioramento cresce in modo abbastanza regolare all'aumentare del numero di nodi controllati. Sulla rete urbana reale, invece, il massimo beneficio si ottiene equipaggiando solo gli incroci a maggiore traffico; estendere il controllo a tutti i nodi non porta necessariamente a un risultato migliore.

Questo aspetto è particolarmente rilevante dal punto di vista applicativo. In una rete reale, caratterizzata da pochi incroci critici e da molti nodi secondari, può essere più efficace intervenire in modo selettivo. Non serve necessariamente controllare tutto. Concentrarsi sui nodi più importanti può permettere di ottenere un miglioramento significativo delle emissioni e, allo stesso tempo, contenere l'impatto infrastrutturale dell'intervento.

Sono emersi anche alcuni limiti. In particolare, la copertura completa sulle reti reali può peggiorare alcune metriche temporali. Questo accade perché, sugli incroci secondari e poco trafficati, l'algoritmo può arrivare a commutare sulla base di differenze molto piccole tra le fasi. In questi casi la maggiore adattività non produce sempre un vantaggio, ma può introdurre una variabilità non necessaria. Proprio per questo, nei contesti urbani reali, il deployment selettivo risulta una scelta più convincente.

Va inoltre ricordato che tutta la validazione è stata svolta in simulazione. Gli strumenti utilizzati permettono di ottenere risultati realistici e coerenti con il contesto italiano, ma una conferma definitiva richiederebbe una sperimentazione su strada. Questo passaggio sarebbe necessario per valutare anche aspetti che in simulazione possono essere solo parzialmente rappresentati.

6.2 Sviluppi futuri

Il lavoro lascia aperte diverse direzioni di sviluppo. Una prima direzione riguarda la taratura dei parametri del modello, che in questa tesi sono stati fissati a priori su base qualitativa. Sarebbe interessante studiare una procedura di taratura automatica, prestando però attenzione al compromesso principale: migliorare le prestazioni senza perdere la semplicità e l'applicabilità immediata dell'approccio.

Una seconda direzione riguarda il deployment parziale. I risultati ottenuti mostrano che la scelta dei nodi da equipaggiare è determinante, soprattutto nelle reti reali. Un possibile sviluppo sarebbe quindi definire un criterio automatico per individuare gli incroci più critici, considerando non solo il volume di traffico, ma anche la posizione del nodo e il suo ruolo nella struttura complessiva della rete.

Un altro possibile passo avanti consiste nell'introdurre forme leggere di coordinamento tra semafori vicini. Non un coordinamento centralizzato, che andrebbe contro l'impostazione del lavoro, ma uno scambio limitato di informazioni locali. Questo potrebbe aiutare a gestire meglio alcune situazioni di traffico senza rinunciare alla natura decentralizzata del controllore.

Restano poi due aspetti più applicativi: la validazione su altre reti urbane reali e una sperimentazione pilota su strada. Entrambi sarebbero utili per capire quanto i risultati ottenuti siano generalizzabili e quali adattamenti sarebbero necessari in un contesto operativo.

Infine, il modello potrebbe essere esteso includendo categorie di utenti che in questa fase non sono state considerate, come il trasporto pubblico, i veicoli di emergenza e gli attraversamenti pedonali. Sono elementi centrali nella gestione reale della mobilità urbana e permetterebbero di rendere il controllore più completo.

In conclusione, questa tesi mostra che un controllore semaforico decentralizzato può essere reso più sensibile sia al traffico in arrivo sia alla congestione a valle, senza perdere le caratteristiche che ne rendono realistica l'adozione. I risultati indicano anche un aspetto importante: in una rete urbana reale non sempre la soluzione migliore consiste nel controllare tutti gli incroci. Una

strategia selettiva, concentrata sui nodi più rilevanti, può offrire un compromesso più efficace tra beneficio ambientale, tempi di percorrenza e costo di implementazione.

Ringraziamenti

Direi che, arrivati alla fine di questa avventura, il tono formale possa finalmente venire meno.

Sono stati due anni piuttosto intensi. Lo studio e l'università sono stati compagni costanti e, forse, finiranno anche per mancarmi. Tuttavia, la cosa più importante è che, durante questo percorso, ho iniziato a riscoprire me stesso.

Anche se non sarà particolarmente elegante, credo di dover ringraziare anche me stesso. Partendo da tutto il lavoro che ho dovuto intraprendere su me stesso, e che ancora oggi non è affatto conclusa, i viaggi tra Parma e Modena, le ore trascorse sui materiali degli esami e tutto il tempo dedicato alla tesi sono stati ripagati, se mi hanno permesso di arrivare alla conclusione di questo capitolo.

La mia famiglia

Vorrei iniziare dalle persone che mi sono state accanto da sempre: la mia famiglia, forse la parte più difficile di questi ringraziamenti. Sapete bene quanto faccia fatica a dimostrare il mio affetto, ma questa volta voglio provarci.

Mamma, Fiorenza. Sei una persona che mette il cuore in ciò che fa e cerca sempre di dare tutto quello che può. Riesci a trasmettere agli altri la passione per le cose che ami, ma ciò che mi ha sempre colpito di più è che non chiedi mai nulla in cambio, nemmeno un grazie. Sei davvero una persona da ammirare e non posso che essere orgoglioso di te. Spero di essere riuscito a diventare, almeno in parte, come te.

Babbo, Giovanni. Molti dicono che ci assomigliamo sotto diversi aspetti e io non posso che esserne fiero. Come la mamma, quando tieni a qualcosa

cerchi di dare tutto te stesso. La passione che metti nei tuoi interessi e la fermezza con cui difendi i tuoi ideali sono qualità alle quali chiunque dovrebbe aspirare. Certo, per farti cambiare idea bisogna essere davvero convincenti, ma, se sono riuscito a diventare anche solo un po' simile a te, non posso che ritenermi fortunato.

Mio fratello, Marco. La passione per i videogiochi, per il mondo dell'informatica e, sì, anche per la cucina è nata in gran parte grazie a te. Alcuni di questi interessi li condividiamo ancora oggi, mentre altri, come la cucina, li ho scoperti inizialmente quasi per necessità. Hai sempre avuto un carattere tutto tuo e, molte volte, mi sono sentito io il fratello maggiore: questo lo sai anche tu, ahah. Una cosa, però, non posso negarla: quando ho bisogno ci sei, anche soltanto per ascoltarmi quando parto per la tangente con uno dei miei discorsi.

Grazie a tutti e tre per avermi accompagnato fino a questo momento, attraverso tutte le fasi che ho vissuto: raccogliendo i miei pezzi nei periodi peggiori e mettendomi una mano sulla spalla in quelli migliori, semplicemente per ricordarmi che ci siete. So di non essere la persona più semplice del mondo, ma sappiate che ho sempre apprezzato tutto ciò che avete fatto per me.

I nonni, Bruno e Piero. Non ho avuto la fortuna di trascorrere con entrambi la stessa quantità di tempo e molte delle cose che conosco sul nonno Piero mi sono arrivate attraverso i racconti degli altri. Per quanto possa sembrare contraddittorio, avevate caratteri molto diversi e, allo stesso tempo, diversi aspetti in comune: entrambi silenziosi, ma capaci, quando parlavate, di lasciare un segno profondo. La determinazione e la convinzione con cui difendevate ciò che vi stava a cuore sono qualità che oggi considero rare. Grazie, quindi, per tutto ciò che, direttamente o attraverso i ricordi di chi vi ha conosciuto, siete riusciti a trasmettermi.

Le nonne, Rosa e Anna. Dire che siete due persone molto diverse sarebbe perfino riduttivo: ognuna di voi possiede caratteristiche davvero uniche. D'altra parte, dopo aver condiviso per tanti anni la vita con due uomini dal carattere così forte, non potevate certamente essere da meno. Avete sempre dimostrato passione e fermezza verso le persone e le cose che amate. Certo, entrambe avete una testa davvero dura e farvi cambiare idea può essere complicato, ma persino questo è diventato per me una fonte d'ispirazione. Grazie a entrambe.

Gli zii e le zie: Gabriele, Enrico, Enrica e Stefania. Avete cercato di esserci e di sostenere non soltanto me, ma tutta la famiglia, sia nella vita di tutti i giorni sia nei momenti più difficili. Siete persone molto diverse tra voi, ma condividete l'attenzione che dedicate alla famiglia e la passione che mettete in ciò che fate. A modo vostro, siete stati tutti una parte importante della mia vita fino a oggi. Grazie davvero.

Le cugine, Greta e Virginia. Con te, Gre, sono praticamente cresciuto e forse è anche per questo che siamo simili sotto molti aspetti. Vedere la determinazione e la costanza che metti in quello che fai è sorprendente, e non posso che essere fiero della persona che stai diventando. Tu, Virgi, hai invece un carattere molto definito: anche se spesso sei silenziosa, quando credi in qualcosa sembra che niente possa fermarti. Sono davvero curioso di vedere dove ti porterà il futuro. Grazie a entrambe per l'affetto e per esserci sempre state.

Anita, Bianco, Luna e Yuri. Ognuno di loro mi ha regalato, a modo proprio, un affetto incondizionato. Lo sanno tutti, però, che il rapporto con **Luna** è stato quello più profondo: sei stata l'unica ad accompagnarmi lungo tutto questo percorso e, anche senza poter usare le parole, sei sempre rimasta al mio fianco. Grazie: sei stata un'ottima compagna di viaggio.

Gli amici

Dopo la famiglia, è arrivato il momento degli amici.

Non sono mai stato una persona particolarmente estroversa e, probabilmente, nemmeno sempre semplice da capire. A prescindere da chi sia qui oggi a festeggiare insieme a me, però, vorrei ringraziare tutti: sia le persone che ho perso lungo il cammino sia quelle che, invece, sono rimaste. In un modo o nell'altro, ognuno di voi ha fatto parte di questi anni e ha contribuito a renderli ciò che sono stati.

A differenza della tesi precedente, questa volta dovrò dilungarmi un po' di più. Voglio provare ad aprirmi maggiormente, cosa che allora avevo cercato di evitare, forse perché ho sempre avuto un po' di timore per il modo in cui i rapporti possono cambiare o finire. Arrivati a questo punto, però, credo sia giusto provarci.

Il gruppo di Discord e della laurea triennale

Comincio dagli amici che accompagnano ormai da non so più quanti anni molte delle mie giornate e delle mie serate.

Seba, alias BoiledEggs. Chi avrebbe mai immaginato di conoscere una persona capace di essere tanto stupida quanto intelligente e con la quale avrei avuto così tante cose in comune, soprattutto in fatto di gusti? Sei un compagno che cerca sempre di coprirmi le spalle, sia nei giochi sia nella realtà. Non credo servano molte altre parole: posso soltanto dirti grazie socio, gls.

Manuel, alias Buch. Probabilmente il primo vero amico con cui abbia condiviso la passione per i videogiochi, conosciuto grazie a un incontro completamente casuale ad atletica. Anche quando ci perdiamo di vista per un po', so che nel momento del bisogno saresti pronto ad aiutarmi. Per questo e per tutte le ore passate insieme, grazie.

Fede, alias Flex_On_Desktop. Eviterò di riportare gli altri tuoi nickname, così da non rischiare una denuncia. Anche tu hai condiviso con me

innumerevoli serate passate a dire cavolate e altre trascorse, invece, a parlare degli aspetti più profondi delle nostre vite e delle nostre esperienze personali. Grazie per la compagnia e per il sostegno.

Cesare, alias Ceieh. Un'altra persona tanto intelligente quanto stupida, ma soprattutto un amico con cui poter scherzare e parlare del futuro, raggiungendo a volte livelli di profondità quasi preoccupanti. Grazie anche a te per esserci stato e per tutte le conversazioni che abbiamo condiviso.

Lollone. Una figura quasi mistica con cui si possono dire cavolate a non finire, ma anche una persona pronta ad aiutare quando serve. Grazie anche a te per aver fatto parte di questi anni.

Mattia, alias Sussolone. Sei stato una presenza costante nelle nostre serate su Discord e, anche se brontoli un po', quando si comincia a parlare di questioni tecniche sei sempre il migliore. Grande Sussolone.

Marco, alias Ronde. Cerchi sempre di renderti disponibile per i tuoi amici. Da quando ci siamo conosciuti sei migliorato tantissimo sotto molti aspetti e, allo stesso tempo, hai aiutato anche me a riscoprire e migliorare alcune parti del mio carattere. Grazie per il sostegno e per tutto quello che, magari senza nemmeno accorgertene, sei riuscito a trasmettermi. Giusto perché non può mancare: "Fiaa Rondelli, tutta colpa tua, come al solito..."

Filippo, alias Fil. Chi avrebbe mai immaginato che una persona presentatami da Ronde sarebbe diventata così affine a me e, in alcuni momenti, persino una fonte d'ispirazione? Sei un ragazzo che cerca continuamente di mettersi in gioco e che riesce a trasmettere agli altri la stessa voglia di farlo. Grazie per la tua amicizia e, soprattutto, per esserci.

Rai Proprio il classico ragazzo silenzioso delle superiori, a cui facevo battute alquanto discutibili durante i cambi d'ora (che però continuo a farti ancora oggi). Non mi sarei mai aspettato che saresti diventato uno dei miei compagni di giri e fiere, con una passione per anime e manga che in pochi possono vantare. Grazie, King Rai, per la tua presenza e per le nostre chiacchierate infinite.

Il gruppo storico

Questo gruppo è nato durante le superiori e, nonostante tutti questi anni, continua ancora a esistere. Nel tempo abbiamo forse perso qualche pezzo, ma ne abbiamo guadagnati tanti altri. Anche in questo caso, come per il gruppo precedente, l'ordine non è legato all'importanza.

Alvaro, alias il King. Sempre presente, dalle superiori fino ad oggi, con te bastano sempre pochi minuti per entrare subito in sintonia e iniziare a dire una quantità enorme di cavolate. Probabilmente era così già dai primi tempi in cui ci siamo conosciuti. Quando serve, però, riusciamo anche a fare discorsi seri e, cavolo, quanti ne abbiamo fatti. Grazie per il sostegno, zio.

Mighé. Sei lo stereotipo dell'ingegnere scoppiato. L'immagine di te con davanti quel libro di chimica, rimasto chiuso e mai aperto per sei mesi, è difficile da dimenticare. Ci sei stato sia nei momenti felici sia in quelli nei quali la nostra sanità mentale iniziava seriamente a vacillare. In un modo o nell'altro, però, siamo sempre riusciti a rialzarci. Grazie davvero.

Cale. C'è poco da dire: sei una di quelle persone che bisogna avere nella propria vita, anche se probabilmente una sola basta e avanza. Hai uno spirito d'iniziativa e una capacità di adattamento che hai coltivato e migliorato negli anni e che sono davvero impressionanti. Riesci a osservare il mondo con una leggerezza a volte quasi invidiabile e sai esserci quando serve. Non posso fare altro che ringraziarti davvero.

Gambaz. Quando penso all'amicizia, probabilmente sei la prima persona che mi viene in mente. Da semplici compagni di banco, conosciuti il primo giorno del terzo anno delle superiori, abbiamo costruito un'amicizia che faccio persino fatica a descrivere. Mi sei stato accanto soprattutto nei momenti peggiori. Anche quando magari non sapevi bene cosa dire, eri comunque lì, e questo è sempre stato ciò che contava davvero.

Dico spesso che sei l'unica persona capace di rimettermi in sesto in cinque minuti, anche quando sto malissimo. È qualcosa di raro e incredibilmente prezioso. Grazie, cicio, per quello che sei, per quello che fai e per esserci

sempre stato. Ti voglio bene.

Naturalmente, un grazie va anche a **Franci, Vale, Marta, Ari e Ali, Alina, Giuse, Giecc, Fuso**. Ognuno di voi, a modo proprio, mi ha accompagnato e sostenuto con la propria amicizia durante questi anni. Grazie davvero anche a voi.

Prima di concludere gli amici, devo per forza di cose ringraziare i tre compagni di corso della magistrale, con cui ho studiato e ripassato per la maggior parte degli esami. Grazie davvero, **Adriano, Davide e Luca**.

E, ovviamente, grazie anche a tutte le persone presenti che non ho citato: non me ne vogliate, ahah.

I professori

Vorrei infine ringraziare i tre professori che mi hanno accompagnato durante questi ultimi sette mesi.

Prof. Giacomo Cabri. È stato anche il primo dei tre che ho avuto la possibilità di conoscere. La sua disponibilità è stata davvero sorprendente: sia durante le lezioni sia nel corso del tirocinio, è sempre stato pronto ad aiutare gli studenti e, naturalmente, anche me. Che si trattasse di una spiegazione o di una questione burocratica, nella quale, come ben sappiamo, è un attimo rimanere intrappolati, ho sempre potuto contare sul suo aiuto. Grazie, quindi, per il suo modo di fare e di essere: una disponibilità davvero ammirevole, che ogni studente dovrebbe avere la fortuna di incontrare lungo il proprio percorso.

Dott. Filippo Muzzini, Ph.D. A differenza del Prof. Cabri, ho avuto la possibilità di conoscerlo soltanto durante il tirocinio. Nonostante il periodo più breve, posso dire che la sua guida e i consigli ricevuti in questi mesi siano stati fondamentali per permettermi di arrivare fin qui. È sempre stato disponibile a rispondere ai miei dubbi, ma anche a farmene nascere di nuovi, spingendomi a ragionare e a migliorare il lavoro. Anche lui rappresenta il tipo di persona che ogni studente dovrebbe avere la fortuna di incontrare lungo la propria strada.

Prof.ssa Manuela Montangero. Ho avuto la possibilità di frequentare le sue lezioni e, di conseguenza, di conoscerla già prima dell'esperienza del tirocinio. Posso dire oggi che le voci che circolavano tra gli studenti erano assolutamente fondate: una professoressa estremamente competente, capace di affrontare argomenti complessi e, allo stesso tempo, disponibile ad aiutare gli studenti a comprenderne non soltanto i concetti, ma anche le fondamenta. Durante il tirocinio è inoltre riuscita a spingermi a pormi nuove domande su ciò che stavo facendo, permettendomi di osservare il mio lavoro da prospettive differenti e di cercare il miglior risultato possibile. Grazie anche per le numerose chiacchierate nei momenti liberi dalle lezioni e dal tirocinio: arri-

vato alla fine della magistrale, è bello essere riuscito a costruire un rapporto di questo tipo con i propri professori.

Grazie, quindi, a tutti e tre. Avete reso questa esperienza davvero stimolante e coinvolgente. Spero sinceramente che la passione che mettete nel vostro lavoro di docenti rimanga sempre la stessa, perché siete riusciti a trasmetterla. E magari, chissà, in futuro le nostre strade potrebbero incontrarsi di nuovo: non si può mai sapere.

Per concludere, grazie a tutti voi, familiari, amici e professori. A modo mio, vi stimo e vi voglio bene.

GANBARE!

Bibliografia

- [CabCMM24] Giacomo Cabri, Denny Ciccia, Manuela Montangero, and Filippo Muzzini. Integrating smart traffic lights for enhanced urban air quality in smart cities. In *IEEE/ACM Symposium on Edge Computing (SEC)*, 2024.
- [ChaK25] Tamrat D. Chala and László T. Kóczy. Agent-based intelligent fuzzy traffic signal control system for multiple road intersection systems. *Mathematics (MDPI)*, 2025.
- [Fai25] Mahi Faiza. Management of traffic light by evolutionary algorithm. *Journal of Intelligent Systems and Engineering Management (JISEM)*, 10(61s), 2025.
- [HacS24] Israa Kadhim Hachim and Dr. Husham L. Swadi. Multi-objective optimization of traffic signal timings for minimizing waiting time, co2 emissions, and fuel consumption at intersections. *Kufa Journal of Engineering*, 2024.
- [HuaZ24] Ke Huang and Jianjun Zhu. A novel method for reducing vehicle emissions utilizing iot-based is-apcpso algorithm. *Applied Artificial Intelligence*, 2024.
- [INF19] INFRAS. *Handbook Emission Factors for Road Transport (HBEFA), version 4*. INFRAS, Bern, Switzerland, 2019.
- [KolKBA23] Máté Kolat, Bálint Kővári, Tamás Bécsi, and Szilárd Aradi. Multi-agent reinforcement learning for traffic signal control: A cooperative approach. *Sustainability (MDPI)*, 2023.
- [KorA24] E. Korkmaz and A. P. Akgüngör. A hybrid traffic controller system based on flower pollination algorithm and type-2 fuzzy

zy logic optimized with crow search algorithm for signalized intersections. *Soft Computing*, 2024.

- [LerU21] Ponlathep Lertworawanich and Panumas Unhasut. A co emission-based adaptive signal control for isolated intersections. *Journal of the Air & Waste Management Association*, 2021.
- [LiaWMN24] Shubing Liao, Yaxin Wu, Kanghua Ma, and Yunyun Niu. Ant colony optimization with look-ahead mechanism for dynamic traffic signal control of iov systems. *IEEE Internet of Things Journal*, 2024.
- [LópBBW⁺18] Pablo Álvarez López, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Eva-marie Wießner. Microscopic traffic simulation using SUMO. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2575–2582. IEEE, 2018.
- [MonGF⁺18] Jaime Espinoza Mondragón, José Alfredo Jiménez García, José Martín Medina Flores, José Antonio Vázquez López, and Sandra Téllez Vázquez. Experiments simulation and design to set traffic lights’ operation rules. *Transport Policy (Elsevier)*, 2018.
- [SanRMD24] Osvaldo Santos, Fernando Ribeiro, José Metrôlho, and Rogério Dionísio. Using smart traffic lights to reduce co2 emissions and improve traffic flow at intersections: Simulation of an intersection in a small portuguese city. *Applied System Innovation (MDPI)*, 2024.
- [ShaZWL24] Junqi Shao, Ke Zhang, Anyou Wang, and Shen Li. Dynamically signal timing optimization of isolated intersection traffic

- lights based on a dual-layer framework. *Journal of Advanced Transportation*, 2024.
- [WanZZ⁺24] Shan Wang, Yu Zhao, Shaoqi Zhang, Dongbo Wang, Chao Wang, and Bowen Gong. A multi-objective optimization method for single intersection signals considering low emissions. *Sustainability (MDPI)*, 2024.
- [YouBR23] Maram Bani Younes, Azzedine Boukerche, and Floriano De Rango. Smartlight: A smart efficient traffic light scheduling algorithm for green road intersections. *Ad Hoc Networks*, 2023.
- [YuaZ24] Zihao Yuan and Xiaoqing Zeng. Dynamic network-level traffic speed and signal control in connected vehicle environment. *Sensors (MDPI)*, 2024.
- [ZhaCHZ24] Gongquan Zhang, Fangrong Chang, Helai Huang, and Zilong Zhou. Dual-objective reinforcement learning-based adaptive traffic signal control for decarbonization and efficiency optimization. *Mathematics (MDPI)*, 2024.
- [ZhaLY⁺25] Shengda Zhao, Zhekun Liu, Jiaxin Yu, Bocheng Ju, Liang Wang, Xiaodong Zhang, and Xinghua Zhang. Simulated bifurcation with high-dimensional expansion for traffic signal optimization on real-world networks, 2025.
- [ZriKK25] Ismail Zrigui, Samira Khouli, and Mohamed Larbi Kerkeb. Adaptive traffic signal control using ai and distributed systems for smart urban mobility. *Journal of Theoretical and Applied Information Technology (JATIT)*, 103(8), 2025.