

ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA

SCUOLA DI INGEGNERIA E ARCHITETTURA
DIPARTIMENTO DI INFORMATICA - SCIENZA E INGEGNERIA
CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

TESI DI LAUREA
in
SISTEMI INTELLIGENTI

**TUNING AUTOMATICO DI PARAMETRI IN UN SISTEMA DI
CONTROLLO SEMAFORICO AUTO-ORGANIZZANTE**

CANDIDATO:
RICCARDO BELLETTI

RELATORE:
Chiar.ma Prof.ssa
MICHELA MILANO

CORRELATORE:
Prof.
ALESSIO BONFIETTI

Anno Accademico 2012/13
Sessione III

a Marco

Indice

Introduzione	1
1 Stato dell'arte dei regolatori semaforici	5
1.1 Controllo a tempi fissi	6
1.1.1 Tempi di sicurezza	8
1.1.2 Flusso di saturazione	10
1.1.3 Manovre ostacolate	12
1.1.4 Progetto della fasatura	13
1.1.5 Durata del ciclo semaforico	14
1.1.6 Criteri di progetto	17
1.1.7 Inserzione, disinserzione e commutazione di piani	20
1.2 Controllo semi-attuato	22
1.2.1 Cambio di fase	22
1.2.2 Adattamento tempo di offset	23
1.2.3 Richiesta di fase	23
1.2.4 Adattamento della durata del verde	23
1.3 Controllo completamente attuato	24
1.3.1 Struttura di un regolatore attuato dal traffico	26
1.3.2 Tipologie di controllo attuato	28
1.3.2.1 Criterio a densità	28
1.3.2.2 Criterio a diminuzione di densità	29
1.3.2.3 Criterio a volume-densità	29
1.3.2.4 Criterio ad occupazione-densità	29
1.3.2.5 Controllo basato sulle fasi	29

1.3.2.6	Controllo basato sui gruppi di segnali	29
2	Approccio Self-Organized	31
2.1	Concetto di auto-organizzazione	32
2.2	Impianti semaforici auto-organizzati	33
2.2.1	Auto-organizzazione ad assegnazione di priorità	34
2.2.1.1	Strategia di ottimizzazione	34
2.2.1.2	Strategia stabilizzatrice	36
2.2.2	SOTL (Self-Organizing Traffic Lights)	37
2.3	Auto-organizzazione in COLOMBO	38
2.3.1	Politica Swarm di alto livello	39
2.3.1.1	Selezione della politica di basso livello	40
2.3.2	Politiche di basso livello	42
2.3.2.1	Default policy	43
2.3.2.2	Request policy	43
2.3.2.3	Phase policy	44
2.3.2.4	Platoon policy	44
2.3.2.5	Marching policy	45
2.3.2.6	Congestion policy	45
3	Parameter Tuning	47
3.1	Configurazione automatica di algoritmi	48
3.1.1	Regolazione on-line	48
3.1.2	Definizione del problema di regolazione dei parametri	49
3.2	Configurazione automatica tramite iterated racing	51
3.2.1	Il modulo irace: descrizione dell'algoritmo	52
4	Risultati sperimentali	57
4.1	Tuning dei parametri della Swarm policy	57
4.1.1	Descrizione dello spazio dei parametri	59
4.1.2	Configurazione di irace	61
4.1.3	Lo script di valutazione hook-run	62
4.1.3.1	Preparazione del file descrittore la logica del semaforo	63
4.1.3.2	Esecuzione della simulazione	63

4.1.3.3	Elaborazione del risultato	64
4.2	Test eseguiti	64
4.2.1	Descrizione dell'architettura computazionale utilizzata	65
4.2.1.1	Informazioni tecniche sul server	66
4.2.1.2	Informazioni tecniche sui client	66
4.2.2	Regolazione di un incrocio singolo	67
4.2.3	Regolazione di una rete complessa	73
	Conclusioni	79
A	Codice dello script hook-run	83
B	Risultati del Parameter Tuning della Swarm policy	87
B.1	Regolazione di un incrocio singolo	87
B.1.1	Grafici comparativi tra le configurazioni migliori e peggiori	87
B.1.2	Risultati dei confronti tra le configurazioni migliori e peggiori	90
B.2	Regolazione di una rete complessa	95
B.2.1	Grafici comparativi tra le configurazioni migliori e peggiori	95
B.2.2	Risultati dei confronti tra le configurazioni migliori e peggiori	98
	Bibliografia	104

Elenco delle figure

1.1	Andamento del ritardo per autoveicolo equivalente	18
1.2	Punti di inserimento delle fasi e adattamento della durata del verde	25
1.3	Sequenza d'impulsi di una spira magnetica	26
2.1	Flussi di pedoni che si auto-organizzano	34
2.2	Funzione di stimolo	41
4.1	Definizione di un parametro per irace	59
4.2	Definizione dello spazio dei parametri regolati	60
4.3	Opzioni definite per il tuning della Swarm policy	61
4.4	Esempio di informazioni per un veicolo riportati nel file tripinfo	64
4.5	Schema esemplificativo dell'architettura	66
4.6	Primo scenario: un singolo incrocio semplice	67
4.7	Confronto globale tra le configurazioni migliori e peggiori (incrocio singolo)	68
4.8	Confronto tra la configurazione migliore e peggiore per la F1 (incrocio singolo)	69
4.9	Confronto tra la configurazione migliore e peggiore per la F3 (incrocio singolo)	71
4.10	Confronto tra la configurazione migliore e peggiore per la F5 (incrocio singolo)	71
4.11	Secondo scenario: una piccola porzione della rete stradale di Bologna	74
4.12	Confronto globale tra le configurazioni migliori e peggiori (Pasubio)	75
4.13	Confronto tra la configurazione migliore e peggiore per la F2 (Pasubio) . .	76
4.14	Confronto tra la configurazione migliore e peggiore per la F3 (Pasubio) . .	76
4.15	Confronto tra la configurazione migliore e peggiore per la F4 (Pasubio) . .	77

Elenco delle tabelle

1.1	Lunghezze dei veicoli	9
1.2	Velocità di sgombero e tempi di uscita	9
1.3	Autovetture equivalenti (Ae)	11
1.4	Livelli di servizio	18
4.1	Parametri propri della politica swarm	58
4.2	Parametri della funzione di stimolo delle politiche di basso livello.	59
4.3	Medie e mediane delle performance delle configurazioni migliori e peggiori (incrocio singolo)	70
4.4	Valori delle configurazioni d'élite migliori (incrocio singolo)	72
4.5	Valori delle configurazioni d'élite peggiori (incrocio singolo)	72
4.6	Medie e mediane delle performance delle configurazioni migliori e peggiori (Pasubio)	78
4.7	Valori delle configurazioni d'élite migliori (Pasubio)	78
4.8	Valori delle configurazioni d'élite peggiori (Pasubio)	78

Introduzione

La gestione del traffico veicolare è una delle principali problematiche che affliggono le città moderne. Il crescente numero di vetture in circolazione, siano questi mezzi privati o appartenenti al trasporto pubblico, pone nuove sfide sul piano dell'ottimizzazione del traffico. Questo non significa solo fluidificare i flussi veicolari ma anche ridurre l'inquinamento prodotto dai mezzi e i costi delle infrastrutture utilizzate per la progettazione dei sistemi di controllo semaforico. Attualmente si fa un largo impiego di sensori per rilevare il traffico, tra cui i più comuni sono le spire magnetiche, la cui installazione e gestione implica costi elevati. In questo contesto si colloca il progetto COLOMBO, inserito all'interno del Settimo programma quadro di ricerca (FP7) dell'Unione Europea. Tra gli obiettivi del progetto vi è l'ideazione di nuovi sistemi di regolazione semaforica capaci di auto-organizzarsi, traendo ispirazione e conoscenze dal campo dell'intelligenza artificiale noto come *swarm intelligence*. A questo scopo sono state sviluppate diverse politiche di gestione semaforica, le quali sono scelte ad alto livello da una macro-politica chiamata *Swarm*.

Gli algoritmi che implementano il funzionamento di questa politica, come ogni altro algoritmo esistente, definiscono dei parametri i cui valori ne influenzano le prestazioni. L'operazione di regolazione dei parametri, o *tuning*, è spesso condotta empiricamente e, tipicamente, comporta una notevole mole di lavoro. Dal momento che la ricerca dei parametri altro non è che un problema di ottimizzazione, lo si può automatizzare.

Lo scopo di questo elaborato è impiegare un approccio di *automatic parameter tuning* per regolare i parametri della politica *Swarm* per la gestione di sistemi semaforici auto-organizzanti, dimostrando l'efficacia e il guadagno prestazionale di questo strumento rispetto ad un tradizionale approccio manuale.

Nel capitolo 1 verrà esposto lo stato dell'arte dei controllori semaforici, a partire dal tradizionale controllo a tempi fissi. In questa prima tipologia di modalità di controllo i dati

sul traffico utilizzati sono storici e manca una forma di feedback in tempo reale con cui adattare il comportamento dei controllori semaforici. Ogni aspetto della regolazione viene considerato sulla base delle condizioni medie note per particolari ore della giornata e per determinati giorni della settimana, portando a definire, quando opportuno, molteplici piani semaforici da alternare tra loro. Vengono esposti tutti gli elementi necessari alla realizzazione di un piano semaforico, in quanto è su questa base che si appoggia ogni altra modalità di controllo, facendo riferimento alle normative italiane, trattate per esteso in [Cam01] e integrando con riferimenti a [HFB⁺13].

La trattazione prosegue con le modalità di controllo semi-attuate e attuate. Le prime sono ancora fortemente legate ad una definizione del piano semaforico come nel caso a tempi fissi, ma compare un primo utilizzo di informazioni provenienti dal traffico reale finalizzate a modificare alcuni aspetti del comportamento del controllore semaforico. Nel caso attuato, invece, viene a mancare un piano semaforico di riferimento e ogni decisione sulla gestione delle intersezioni è fornita da un sistema fortemente centralizzato che sfrutta le informazioni provenienti da una vasta gamma possibile di sensori installati nelle strade, tra cui la più adottata è la spira magnetica.

Il capitolo 2 illustra un nuovo tipo di approccio al problema della regolazione semaforica, basandosi sul concetto di auto-organizzazione, un insieme di meccanismi definiti all'interno dell'area delle intelligenze artificiali, più in particolare nel settore della *swarm intelligence*. Il controllo basato su auto-organizzazione dei singoli semafori, sfruttando nozioni della già citata *swarm intelligence*, è un obiettivo del progetto COLOMBO, ma non è il primo caso proposto. Vengono presentati due approcci precedenti, tra cui SOTL (Self-Organizing Traffic Lights), proposto in [CGD08], da cui COLOMBO trae ispirazione.

In seguito è esposto l'approccio ideato al suo interno, in cui si ha una gestione del traffico su due livelli distinti: uno microscopico, finalizzato alla sola regolazione corretta del traffico, ed uno macroscopico, che ha il compito di scegliere il corretto approccio da utilizzare a basso livello. Viene quindi illustrato il funzionamento dell'algoritmo di selezione della politica corretta, la *Swarm policy*, e il comportamento delle politiche di basso livello implementate.

Il nodo cruciale di questo elaborato viene trattato nel capitolo 3, in cui è esposto il significato del problema della regolazione dei parametri, da dove trae origine e come viene affrontato, con riferimenti a [Bir09], da cui è tratta la definizione di *tuning problem* così come è stato affrontato nell'ambito di questo lavoro, e a [LIDLSB11], in cui viene forn-

to l'approccio denominato *iterated racing*, adottato per la regolazione dei parametri della politica Swarm, e descritto il software *irace*, utilizzato per questo scopo.

La trattazione prosegue con il capitolo 4, in cui sono esposti gli scenari utilizzati per il tuning di Swarm. Questa è implementata all'interno del simulatore di traffico SUMO, utilizzato nel contesto del progetto COLOMBO ma sviluppato indipendentemente da questo. L'analisi della validità del tuning automatico dei parametri verrà valutata confrontando le migliori configurazioni trovate da *irace*, rispetto alla minimizzazione dei tempi di attesa medi dei veicoli, con le peggiori configurazioni, ottenute sempre tramite regolazione automatica ma con obiettivo invertito. Nella comparazione tra le performance dei valori ottimali e inadatti dei parametri sarà possibile individuare lo spazio di variabilità del criterio di riferimento, il tempo medio di attesa dei veicoli, e quindi apprezzare le potenzialità dell'automated parameter tuning.

Capitolo 1

Stato dell'arte dei regolatori semaforici

Le *modalità di controllo* dei regolatori semaforici descrivono il processo di stesura di un piano semaforico e l'approccio utilizzato per la gestione del traffico nell'intersezione controllata dall'impianto. Si possono individuare due livelli di modalità di controllo, a seconda degli elementi che vengono presi in considerazione.

Un primo livello, definito *macroscopico*, considera informazioni come la densità di traffico media o la durata media della congestione dei tratti stradali interessati. Sulla base di queste informazioni, unite ai dati sui carichi di traffico a lungo termine della rete stradale e delle intersezioni, si possono definire piani semaforici dipendenti dagli orari o dal traffico. Nello specifico, si parla di *scelta del piano semaforico dipendente dall'orario o dipendente dal traffico*. Nel primo caso, si hanno a disposizione una serie di differenti piani semaforici regolati in base a condizioni di traffico note e predicibili, per esempio legate agli orari di punta. Il piano da utilizzare è quindi scelto in base al giorno e all'orario attuale. Questo tipo di modalità di controllo funziona bene fino a che le condizioni di traffico sono stabili e, come detto precedentemente, predicibili. L'intera pianificazione dei piani e degli orari di cambio della selezione viene fatta offline. Il secondo caso, invece, è quello dei *piani semaforici scelti in base alle condizioni del traffico*. Anche in questa situazione sono presenti più piani semaforici definiti a priori e la selezione tra questi avviene sulla base dell'effettivo traffico rilevato. Le informazioni rilevate vengono utilizzate all'interno di equazioni per scegliere il piano semaforico più opportuno e possono anche servire per sviluppare metodi di controllo basati su modelli.

L'altro livello di analisi è detto *microscopico*, in cui sono considerati cambiamenti improvvisi dei flussi di traffico, che possono avvenire all'interno di uno solo ciclo semaforico e

portano a definire tre tipi di modalità di controllo: statico, adattivo e a formazione di piano. Queste modalità comunque prevedono una definizione a priori del piano semaforico, se non tutto almeno in parte. Il caso *statico* o *a tempi fissi*, come suggerisce il nome, non prevede modifiche negli elementi del piano semaforico, i quali restano immutati una volta definiti. I metodi *adattivi*, detti anche *parzialmente dipendenti dal traffico*, consentono la modifica di singole caratteristiche dei piani semaforici in base alle condizioni del traffico. Questo approccio prevede una durata prefissata del ciclo semaforico e la possibilità di alterare:

durata del periodo di verde variando la durata del verde in base alla situazione corrente.

La durata del ciclo è fissa ma viene consentito un tempo di allungamento di durata variabile;

cambio di fase in cui viene alterata la sequenza delle fasi del piano semaforico;

richiesta di fase dove una particolare fase richiesta in uno o più punti del piano semaforico, incidendo (negativamente) sulle durate dei verdi delle altre fasi;

tempo di offset in cui si varia l'inizio relativo tra periodi di verde durante un ciclo.

Infine, si parla di *controllo a formazione di piano* per indicare una modalità di controllo in cui tutti gli elementi del piano semaforico sono modificabili. Non esiste un piano base con durata del ciclo fissa, come nei metodi parzialmente dipendenti dal traffico, ma la regolazione avviene in base al traffico rilevato, motivo per cui si parla anche di *controllo totalmente dipendente dal traffico*.

Data l'organizzazione prettamente teorica del livello microscopico, si preferisce parlare di controllo a tempi fissi e dipendente dal traffico, quest'ultimo suddiviso in parzialmente dipendente e totalmente dipendente[HFB⁺13] o, alternativamente, *semi-attuato* e *attuato*. [dIedT02]

1.1 Controllo a tempi fissi

Il controllo a tempi fissi è la forma più semplice di controllo semaforico realizzabile. I parametri del piano semaforico (durata del ciclo, tempi di verde, tempi di sicurezza, composizione e sequenza delle fasi[Cam01]) sono determinati staticamente in base ad una determinata situazione di traffico che si vuole gestire, la cui misurazione stabilisce l'efficacia del

piano stesso.[HFB⁺13] Ciò implica un degrado delle prestazioni in caso di variabilità del traffico, problema che può comunque essere aggirato determinando, per diverse condizioni di traffico stabilite a priori, diversi piani semaforici tra cui poter scegliere.[Cam01]

Un piano semaforico a tempi fissi è caratterizzato da:

- Tempi di sicurezza;
- Composizione delle fasi;
- Sequenza delle fasi;
- Durata del ciclo;
- Tempi di verde.

Tutti questi elementi devono essere determinati sulla base delle condizioni di traffico che si vogliono gestire con il piano semaforico. Secondo [Cam01], i passi per la progettazione di un piano semaforico sono i seguenti:

- Calcolo dei tempi di sicurezza;
- Progetto della fasatura;
- Calcolo dei flussi critici e dei flussi di saturazione;
- Calcolo della durata del ciclo;
- Calcolo dei tempi di verde;
- Disegno del piano semaforico;
- Verifica della capacità;
- Verifica del tempo perso e del numero di arresti.

A questi si aggiungo due ulteriori passi, qualora si sta realizzando più di un piano da usare alternativamente a seconda delle fasce orarie:

- Determinazione degli intervalli di tempo in cui inserire il piano;
- Determinazione degli istanti di commutazione.

1.1.1 Tempi di sicurezza

I tempi di sicurezza¹ sono calcolati in base alla geometria dell'intersezione da regolare e facendo ipotesi cautelative sulle velocità adottate dai veicoli durante l'ingresso o l'uscita dall'incrocio, rispettivamente a inizio o fine del verde. Vengono calcolati per tutte le coppie di correnti in conflitto e prendono in considerazione le loro traiettorie, le quali si possono ottenere dalla conformazione dell'intersezione. Il calcolo del tempo di sicurezza è quindi ottenuto da

$$t_s = t_u + t_e - t_i \quad (1.1)$$

con

t_s tempo di sicurezza;

t_u tempo di uscita, corrispondente al tempo dopo la fine del verde in cui un veicolo può ancora uscire per effetto del giallo;

t_e tempo di sgombero, corrispondente al tempo necessario ad un veicolo in uscita per superare il punto di conflitto dell'intersezione;

t_i tempo d'ingresso, corrispondente al tempo richiesto ad un veicolo sopraggiungente in direzione antagonista di percorrere lo spazio tra la linea d'arresto e il punto di conflitto.

I tempi di sicurezza così calcolati sono poi raccolti in una matrice $T_s = [t_{sij}]^2$ dove il generico elemento t_{sij} rappresenta il massimo tempo tra la fine dell' i -esimo verde e l'inizio del j -esimo verde.

Il tempo di sgombero è calcolato come

$$t_e = \frac{(l_e + l_v)}{v_m} \quad (1.2)$$

con

l_e la distanza di sgombero, calcolata dalla linea d'arresto fino al punto di conflitto più distante;

¹intergreen time in [HFB⁺13]

²intergreen matrix in [HFB⁺13]

Tabella 1.1: Lunghezze dei veicoli

Tipi di veicolo	l_v (m)
Cicli	0
Veicoli (sia leggeri sia pesanti)	6
Tram	15

[Cam01]

Tabella 1.2: Velocità di sgombero e tempi di uscita

Corrente in uscita	v_m (m/s)	t_u (s)
Veicoli che procedono dritto	10	3
Veicoli in svolta (raggio ≥ 10 m)	7	2
Veicoli in svolta (raggio < 10 m)	5	2
Tram o Bus senza fermata prima dell'intersezione	≤ 8	3
	8 - 14	5
	14 - 19	7
Tram o Bus con fermata prima dell'intersezione*	V_{\max}	0
Biciclette	4	1
Pedoni	1 - 1,5	0

* Nel caso di mezzi pubblici con fermata in corrispondenza dell'intersezione il tempo di sgombero è calcolato basandosi sull'accelerazione e sulla massima velocità raggiungibile dal mezzo. Si considerano valori di accelerazione tra 0,7 e 1,2 m/s² per i tram e tra 1,0 e 1,5 m/s² per i bus.

[Cam01, HFB⁺13]

l_v la lunghezza fittizia del veicolo in fase di sgombero, come riportata in tabella 1.1;

v_m la velocità media di percorrenza dei veicoli in fase di sgombero, come riportata in tabella 1.2.

Al tempo di sgombero va poi aggiunto quello d'uscita t_u , ovvero il tempo che intercorre tra la fine del verde di un segnale e l'inizio del tempo di sgombero. Anche questo varia in base alla velocità e alle caratteristiche del veicolo e i valori da considerare sono riportati in tabella 1.2.[Cam01]

Il tempo complessivo di sgombero è quindi calcolato come

$$t_{te} = t_u + t_e = t_u + \frac{(l_e + l_v)}{v_m} \quad (1.3)$$

[HFB⁺13]

Infine, per il calcolo del tempo d'ingresso t_i bisogna determinare la distanza d'ingresso l_i , corrispondente a quella tra la linea d'arresto e il punto di conflitto più vicino. Per quanto riguarda i pedoni, tipicamente questa distanza è da considerarsi nulla. Per veicoli leggeri si suppone una velocità media di 11 m/s (40 km/h) per il primo veicolo che percorrerà tale tratto, per cui si ottiene

$$t_i = \frac{l_i}{11} \quad (1.4)$$

Per i mezzi pubblici si hanno sempre due casi, dipendenti dalla presenza o meno della fermata prima dell'intersezione. Nel primo caso si suppone una velocità nulla alla partenza del veicolo e accelerazione nell'ordine di 1 m/s^2 , altrimenti l'ipotesi è che arrivino con velocità pari a $5,5 \text{ m/s}$ (20 km/h) e accelerino fino a raggiungere la loro velocità massima v_{\max} . Il tempo d'ingresso è quindi calcolato come

$$t_i = \sqrt{\frac{2 * l_i}{a}} \quad \text{per} \quad t_i \leq \frac{v_{\max}}{a} \quad (1.5)$$

Nel caso siano presenti piste ciclabili, si suppone che le biciclette arrivino con velocità media pari a 5 m/s , di conseguenza

$$t_i = \frac{l_i}{5} \quad (1.6)$$

Infine, se per i pedoni la distanza d'ingresso non è nulla in quanto i veicoli non passano nell'immediata vicinanza del marciapiede, la velocità considerata è $1,5 \text{ m/s}$, per cui si ottiene

$$t_i = \frac{l_i}{1,5} \quad (1.7)$$

[Cam01, HFB⁺13]

1.1.2 Flusso di saturazione

Con il termine di *flusso di saturazione* si definisce il valore del flusso di veicoli che può attraversare un attestamento semaforizzato con segnale verde in presenza di coda continua e viene espresso in $\frac{\text{autovetture equivalenti}}{\text{unità di tempo}}$, tipicamente Ae/s oppure Ae/h . I valori comunemente utilizzati sono riportati in tabella 1.3 nella pagina successiva

Il calcolo del flusso di saturazione dipende da molteplici fattori e non è definito pre-

Tabella 1.3: Autovetture equivalenti (A_e)

Tipo veicolo	A_e
Biciclette	0,20
Motocicli	0,33* - 0,50 ⁺
Autovetture	1,00
Autobus	2,00* ⁺ - 3,00 [#]
Veicoli pesanti	2,00
Veicoli pesanti con rimorchio / Tram	2,50
Autotreni e autoarticolati	5,00

* [Cam01] + [HFB⁺13, dIedT02] # [Cap09]

cisamente [HFB⁺13]. Vengono quindi riportati due diversi approcci. Il primo, fornito da [Cam01], propone una stima del flusso di saturazione ideale s_0 , pari a 1900 A_e/h riferito ad una corsia di circa 3,6 m in piano, in assenza di mezzi pesanti, parcheggio, fermate e manovre di svolta, a cui applicare diversi fattori correttivi che rappresentano l'influenza dei diversi fattori individuati. Si ha

$$s = s_0 * N * k_{larg} * k_{vp} * k_{pend} * k_{parch} * k_{bus} * k_A * k_{ut} * k_{dx} * k_{sx} \quad (1.8)$$

con

N numero di corsie;

k_{larg} effetto della larghezza delle corsie;

k_{vp} presenza di veicoli pesanti;

k_{pend} effetto della pendenza;

k_{parch} presenza di parcheggio a lato della strada;

k_{bus} presenza di fermate dei bus;

k_A area in cui si trova l'intersezione;

k_{ut} coefficiente di utilizzazione delle corsie;

k_{dx} effetto delle svolte a destra;

k_{sx} effetto delle svolte a sinistra.

In [HFB⁺13], invece, viene proposto un approccio più semplice, in cui si tiene conto della sola larghezza delle corsie. Si ottiene

$$s = 165 * w + 45 \quad (1.9)$$

per ogni intervallo di 15 minuti. Il valore ottenuto può poi essere moltiplicato per un fattore tra 0 e 1 che deriva da osservazioni e misure relative all'intersezione sotto esame.

1.1.3 Manovre ostacolate

All'interno di un'intersezione semaforica alcune manovre possono venir ostacolate dalla presenza di correnti, definite antagoniste, aventi diritto di precedenza. Queste possono essere sia veicolari sia pedonali. Il caso più semplice è quello della svolta a destra di veicoli, i quali devono sempre dare la precedenza ai pedoni che camminano parallelamente alla strada da loro percorsa. Il caso più complicato, invece, è la svolta a sinistra di veicoli, in cui si hanno molteplici correnti antagoniste da considerare, sia veicolari provenienti dalla direzione opposta sia pedonali. La presenza di queste manovre ovviamente influisce sulle prestazioni dell'intersezione e possono essere gestite attraverso la regolazione semaforica, qualora sia reso necessario dalla geometria dell'intersezione (presenza di corsie esclusive o condivise) e dall'effettiva entità del traffico della manovra sotto esame e dei flussi antagonisti.

In base alla regolazione semaforica adottata, si individuano tre tipologie di fasi differenti in cui la manovra ostacolata può avvenire:

fase protetta i veicoli possono svoltare senza pericoli di conflitti con le correnti antagoniste;

fase non protetta i veicoli in svolta devono prestare attenzione alle correnti antagoniste, alle quali devono dare precedenza. Si parla anche di fase *permessa*;

fase parzialmente protetta la svolta avviene parzialmente in condizioni protette e parzialmente in condizioni non protette.[Cam01]

1.1.4 Progetto della fasatura

Il progetto della fasatura consiste in due obiettivi: individuazione delle fasi e determinazione della loro successione. Gli attestamenti dell'intersezione sono suddivisi in correnti e, per ognuna di queste, viene specificato il modo d'uso delle rispettive corsie. Ad ogni corrente viene inoltre assegnato un segnale e tutti quelli che controllano correnti non in conflitto possono essere raggruppati in una fase. Di conseguenza, il numero di fasi è determinato dalla suddivisione delle correnti e dalla necessità di farle defluire tutte. Determinate le fasi, si può stabilire in che sequenza debbano attivarsi e progettare le transizioni.

I fattori che influiscono sulla fasatura sono l'uso delle corsie e le manovre di svolta a destra e a sinistra in quanto, come illustrato in 1.1.3 a fronte, possono essere inserite in fasi appositamente studiate per garantirne la sicurezza o meno. Per quanto riguarda le modalità di sfruttamento delle corsie, queste impongono tre tipi di vincoli sulla definizione delle fasi, nello specifico si parla di

simultaneità manovre veicolari *distinte* che utilizzano *promiscuamente* le stesse corsie devono avere il verde simultaneamente;

sequenzialità manovre veicolari *distinte* che utilizzano corsie *specializzate* possono avere il verde in fasi differenti;

ritardato inizio del verde una corrente con precedenza non può avere il verde in un istante di tempo successivo a quello in cui lo ottiene una corrente compatibile condizionata.

I vincoli imposti dalle manovre a destra e a sinistra, invece, rientrano in un discorso di ottimizzazione della capacità e della sicurezza dell'intersezione.

Da un lato è necessario massimizzare la prima, determinando quindi un numero di fasi minimo che garantisca il deflusso di tutte le manovre. Ciò lo si ottiene accorpando quante più correnti possibili in una stessa fase, compatibilmente con i loro tempi di verde, e riducendo la somma dei tempi inutilizzati dalle correnti per via delle transizioni di fase e dei tempi di sicurezza. Con il termine di *transizione di fase* si fa riferimento al tempo tra l'istante della prima commutazione al rosso tra i segnali della fase uscente e l'ultima commutazione al verde della fase entrante. La durata di una transizione di fase è, come minimo, pari a quella del tempo di sicurezza ma può durare di più per soddisfare particolari altre condizioni.

Dall'altro lato, invece, bisogna prendere in considerazione i conflitti che possono verificarsi nell'inserimento di fasi non protette ed i vantaggi, in termini di sicurezza e gestione delle correnti veicolari, dati dall'utilizzo delle fasi protette.

Le manovre più problematiche sono le svolte a sinistra, dato l'elevato numero di conflitti che generano. Proteggere completamente una svolta a sinistra è necessario nel momento in cui il traffico in direzione opposta presenti una molteplicità di punti di conflitto, sia molto veloce o se una delle due correnti sia particolarmente intensa. La protezione consente di smaltire in breve tempo i veicoli sulla manovra di svolta e produce vantaggi in termini di sicurezza, nel caso la svolta a sinistra goda di ridotta visibilità. Se le correnti antagoniste sono di entità modesta, invece, si può anche ricorrere alla svolta non protetta. Proteggere parzialmente una svolta, infine, consente ai veicoli di passare liberamente per un limitato periodo di tempo, ottenuto attraverso il prolungamento o l'anticipazione di un segnale di verde, e obbliga il rispetto delle precedenza durante il resto della manovra. Durante questa parte della svolta, sarebbe necessario informare i veicoli della possibilità di conflitti ma nel Codice della Strada non è previsto alcun modo per farlo.

Le svolte a destra sono generalmente meno problematiche ma, anche in questo caso, sono consigliate alcune riflessioni. Garantire protezione è necessario solo in presenza di correnti in conflitto particolarmente elevate, se la velocità dei veicoli in svolta possa costituire pericolo per le correnti antagoniste o se la manovra viene effettuata su più corsie. Scegliere di non proteggere la svolta comporta la possibilità di segnalare, tipicamente attraverso l'uso di luce gialla lampeggiante, il conflitto con la corrente pedonale. Infine, la svolta a destra può anche non essere segnalata se questa viene effettuata su una corsia dedicata con presenza di isola spartitraffico.[Cam01]

1.1.5 Durata del ciclo semaforico

Partendo dalle nozioni di flusso di saturazione, tempi persi e verde effettivo, si può arrivare a calcolare la durata minima del ciclo semaforico. La prima è stata definita in 1.1.2 a pagina 10. I tempi persi sono dovuti da parti delle transizioni di fase³ e dai transitori dovuti dall'avviamento delle code all'inizio del verde e di uscita dall'intersezione alla fine del verde e durante il giallo. La somma dei tempi persi legati a inizio e fine del verde sono

³Il tutto rosso tra la fine di un giallo e l'inizio del verde della fase successiva, derivabile dalle durate della transizione e del tempo di giallo.

calcolabili come

$$l = l_1 + l_2 = V + G - v \quad (1.10)$$

con

l_1 il tempo perso all'avviamento, dovuto all'inerzia dei veicoli quando scatta il verde;

l_2 il tempo perso all'uscita, dato dal graduale arresto dei veicoli;

V la durata del verde;

G la durata del giallo;

v il verde efficace.

Proprio a causa di questi transitori presenti all'inizio e alla fine della fase di verde, in cui il flusso dei veicoli non è costante, viene conveniente definire il *verde efficace*. Questo corrisponde a un tempo di verde con capacità equivalente a quello reale ma in cui il valore del flusso è costante. In termini matematici, il verde efficace è ottenuto attraverso una funzione $a(t)$, che esprime l'andamento del flusso dei veicoli in partenza nel tempo, e da un valore costante s , corrispondente al valore costante del flusso per cui viene definito il verde efficace. Si ottiene

$$v_k = \frac{1}{s} \int_0^{V+G} a(t) dt \quad (1.11)$$

Si può ora procedere al calcolo della durata minima⁴ del ciclo semaforico. Ponendo

f_k il valore del massimo flusso entrante per la fase k (Ae/s);

s_k il valore del flusso di saturazione per la fase k (Ae/s);

P il tempo perso totale (s);

v_k il tempo di verde effettivo per la fase k (s);

$y_k = f_k/s_k$ il rapporto dei flussi.

⁴La durata minima è quella necessaria per smaltire i flussi reali, i quali sono espressi in termini di autovetture equivalenti.

si può definire il tempo perso totale come

$$P = C_m - \sum_{k=1}^{nf} v_k \quad (1.12)$$

e, imponendo l'uguaglianza tra i flussi entranti ed uscenti

$$C_m * f_k = v_k * s_k \quad (1.13)$$

si arriva a definire la durata del ciclo minimo come

$$C_m = \frac{P}{1 - Y} \quad \text{con} \quad Y = \sum_{k=1}^{nf} y_k \quad (1.14)$$

Dato che i flussi veicolari, nella realtà, non sono costanti e continui, è più opportuno usare una diversa espressione per il calcolo della durata del ciclo. Questa seconda espressione consente di avere un ritardo minimo per i veicoli in attesa

$$C_o = \frac{1,5 * P + 5}{1 - Y} \quad (1.15)$$

Il ritardo minimo complessivo si ottiene se vi è proporzionalità tra le durate del verde effettivo e i rapporti dei flussi y_k , relazione ottenuta attraverso

$$v_k = \frac{y_k}{Y} (C_o - P) \quad (1.16)$$

in questo modo il verde di ogni fase è calcolato in base alla durata del ciclo, dei tempi persi e dei rapporti tra massimo flusso entrante e flusso di saturazione. La durata del verde è comunque soggetta a vincoli. Di norma, il verde non deve essere inferiore ai 10 secondi⁵ ma può essere abbassata fino a 5 se i flussi veicolari sono particolarmente bassi. Pari a 5 secondi è anche la durata minima del verde per mezzi pubblici, biciclette e pedoni. Nel caso siano presenti svolte a sinistra senza frecce direzionali, invece, la durata minima non dovrebbe essere inferiore ai 10 secondi, anche questi abbassabili a 5 in presenza delle frecce a indicare la parte protetta della svolta. Il verde può essere sia anticipato sia ritardato, a seconda delle necessità. Il primo caso si può avere in presenza di conflitti tra pedoni,

⁵15 secondi nel caso delle correnti principali.

beneficiari dell'anticipazione, e traffico veicolare. L'entità dell'anticipazione si basa sul tempo necessario per impegnare l'area di conflitto. Il verde può venir ritardato in caso di svolte a sinistra non protetta, quando è necessario consentire ai veicoli in svolta e in attesa di poter sgombrare l'area dell'incrocio senza rischi. Si stima un tempo di 2-4 secondi come ragionevole per consentire il completamento della svolta.

Così come il verde ha dei vincoli, anche il rosso complementare ne ha. La sua durata massima è legata a diversi fattori sia dipendenti dalla geometria dell'intersezione (disponibilità di spazi di accumulo per veicoli, pedoni e ciclisti) sia umani. Le persone in coda, infatti, riescono a tollerare un tempo di attesa limitato, calcolato in 60 secondi per pedoni e ciclisti e fino al doppio per i guidatori di veicoli.[Cam01]

1.1.6 Criteri di progetto

La principale metrica di valutazione di un impianto semaforico è data dal ritardo medio dei veicoli in transito. Con *ritardo* si indica il tempo aggiuntivo necessario al veicolo per attraversare l'intersezione rispetto alla situazione senza semaforizzazione. Le componenti del ritardo sono la frenata, l'attesa e l'avviamento. Quest'ultima parte viene solitamente trascurata, introducendo errori accettabili rispetto alla totalità dei ritardi, quando si adotta la semplificazione del *ritardo ridotto*. Come individuato da [Cam01], la letteratura propone molte espressioni per la valutazione del ritardo e sceglie di riportare la seguente

$$r = \frac{C * (1 - \lambda)^2}{2 * (1 - \lambda x)} + \frac{x^2}{2 * f * (1 - x)} - 0,65 * \left(\frac{C}{f^2}\right)^{\frac{1}{3}} * x^{(2+5\lambda)} \quad (1.17)$$

dove

r indica il ritardo (s/Ae);

C indica la durata del ciclo (s);

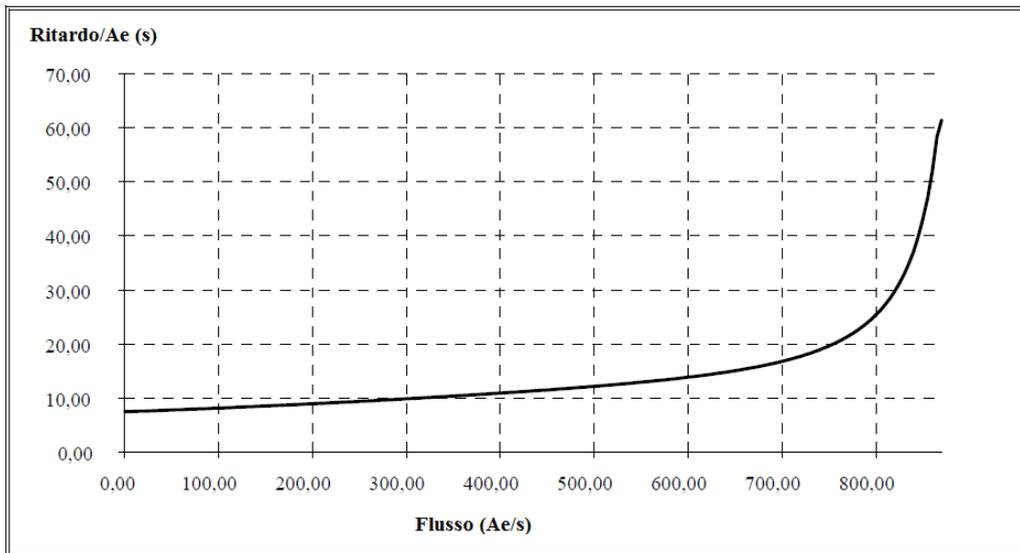
v indica il verde efficace (s);

s indica il flusso di saturazione (Ae/s);

f indica il flusso entrante (Ae/s)

$\lambda = v/C$ indica la percentuale di verde;

Figura 1.1: Andamento del ritardo per autoveicolo equivalente



[Cam01]

Tabella 1.4: Livelli di servizio

Livello di servizio	Ritardo per veicolo (s)
A	≤ 10
B	$> 10, \leq 20$
C	$> 20, \leq 35$
D	$> 35, \leq 55$
E	$> 55, \leq 80$
F	> 80

[Cam01]

$x = f/(\lambda s)$ indica il coefficiente di saturazione.

Come esempio, in figura 1.1 viene riportato l'andamento per autoveicolo equivalente ponendo $C = 60 s$, $v = 30 s$, $s = 0,5 Ae/s$.

Il valore del ritardo serve anche per definire il *livello di servizio* dell'intersezione semaforizzata, indice della misura degli effetti negativi della semaforizzazione in termini di comfort, consumi di carburante e tempo di tragitto. Per definire il livello di servizio si fa riferimento ad un periodo di analisi di 15 minuti, misurati sul campo o valutati attraverso una procedura standardizzata, ed i vari livelli coi rispettivi valori di ritardo medio sono riportati alla tabella 1.4.

Nella misura del ritardo rientrano diversi fattori, tra i quali durata del ciclo, percentuale di verde, grado di saturazione e coordinamento con gli impianti semaforici vicini. I valori di livello di servizio (LdS) della tabella 1.4 a fronte corrispondono alle seguenti indicazioni:

LdS A gli impianti sono ben coordinati e la maggior parte dei veicoli arriva alla commutazione del segnale verde, senza bisogno di fermarsi;

LdS B indica un buon livello di coordinamento con gli altri impianti oppure l'impiego di cicli di breve durata;

LdS C è sintomo di uno scarso coordinamento tra gli impianti semaforici o di cicli di lunga durata ma nonostante l'elevato numero di veicoli che si fermano alcuni riescono ancora a superare l'intersezione senza fermarsi;

LdS D la congestione dell'intersezione è visibile. I ritardi possono essere causati da un coordinamento sfavorevole, cicli di elevata durata o elevati gradi di saturazione. La maggior parte dei veicoli si ferma e può verificarsi la sovrasaturazione dell'incrocio durante alcuni cicli;

LdS E i ritardi raggiungono il massimo accettabile dagli utenti ed è sintomo di un coordinamento sfavorevole, durate di cicli elevati ed elevati indici di saturazione. Diventa sempre più frequente la perdita di un ciclo;

LdS F i ritardi sono considerati inaccettabili dalla maggioranza degli utenti ed è sintomo di elevati indici di saturazione, oltre che di uno sfavorevole coordinamento con gli altri semafori e di un'elevata durata dei cicli.

Altre metriche di interesse sono le code, il numero di arresti e la capacità complessiva dell'intersezione. Per la prima di queste, è interessante valutarne la lunghezza, verificabile all'inizio del tempo di verde. Vi sono due modi per indicare il numero medio di autoveicoli equivalenti in una coda per una corsia:

$$n_1 = f((C - v)/2 + r) \quad n_2 = f(C - v) \quad (1.18)$$

viene considerato il valore massimo tra n_1 e n_2 , indicato come N_q , e la lunghezza della coda è ottenuta considerando una lunghezza media per autoveicolo equivalente. Con un valore

di 5,6 metri, si ha

$$q = 5,6 * Nq \quad (1.19)$$

Il numero di arresti è solitamente affiancato al ritardo medio. Vengono conteggiati anche i veicoli che devono frenare ma non si fermano completamente. Si parte dal numero di veicoli in coda, calcolato precedentemente, e si considera il caso in cui questi riescano a defluire tutti nel tempo di verde, ovvero se vale

$$Nq/(s - f) < v \quad (1.20)$$

in questo caso la percentuale del numero di arresti è pari a

$$N_a\% = \frac{Nq}{(f * C * (1 - f/s))} * 100 \quad (1.21)$$

Diversamente, se vale la condizione opposta

$$Nq/(s - f) > v \quad (1.22)$$

la percentuale è calcolata come

$$N_a\% = \left(\frac{Nq}{f * C} + \frac{v}{C} \right) * 100 \quad (1.23)$$

Ultima metrica è la capacità complessiva. Ciò che si vuole ottenere è la sua massimizzazione, importante dato che condiziona fortemente la capacità della rete stradale. Nel caso di un piano semaforico, la capacità si può calcolare determinando i flussi massimi smaltibili da ogni segnale quando è verde e confrontandoli con i flussi effettivi. Il segnale per cui è minima la differenza tra flusso massimo e flusso minimo è detto *segnale critico* dell'intersezione.[Cam01]

1.1.7 Inserzione, disinserzione e commutazione di piani

Se per un'intersezione semaforica vengono previsti piani diversi per differenti situazioni di traffico, è necessario anche definire in quali istanti di tempo debba avvenire l'inserzione, la commutazione e la disinserzione di un piano. L'insieme di questi istanti di tempo, definiti

nell'arco di una giornata, unitamente ai relativi piani definisce un *programma giornaliero di attivazione*. Il numero di piani definibile è relativo all'andamento del traffico in una giornata ma tipicamente si usano almeno quattro piani, due per gli orari di punta di mattina e sera, uno per le ore serali e uno per le ore intermedie.

Tuttavia, un programma giornaliero può non essere sufficiente per coprire tutti i giorni della settimana. Infatti, il traffico presenta caratteristiche diverse in giornate diverse, distinguendo il week-end dai giorni infrasettimanali e anche tra questi si possono avere differenze a causa della presenza di mercati o della concomitanza col giorno di chiusura delle attività. Si definisce quindi un insieme di programmi giornalieri che prende il nome di *programma settimanale di attivazione*.

Le operazioni di inserzione, disinserzione e commutazione devono essere fatte nella massima sicurezza, senza disturbare il deflusso del traffico ed evitando una percezione negativa da parte dell'utente. Si rende quindi necessaria una particolare accortezza, in fase di progetto, per la determinazione degli istanti in cui effettuare le modifiche.

La *commutazione* deve avvenire in modo da lasciare inalterato lo stato dei segnali, ovvero bisogna trovare un istante temporale nel piano di origine e in quello di destinazione in cui i segnali abbiano tutti lo stesso aspetto. La commutazione deve, inoltre, rispondere ad altri vincoli. L'istante scelto non deve contenere tempi di sgombero e non deve ridurre i tempi di verde sotto i valori minimi di sicurezza. Se tale istante non è identificabile, è necessario modificare uno dei due piani per soddisfare questi requisiti o, in caso estremo, progettare un piano per la transizione.

L'*inserzione* di un piano avviene partendo dallo stato di impianto funzionante a lampeggio attraverso uno stato intermedio di tutto giallo o tutto rosso. Si preferisce usare il primo in quanto non induce confusione negli utenti, a patto di garantire le condizioni di sicurezza.

La *disinserzione* di un piano è il passaggio allo stato di giallo lampeggiante, configurazione che può attivarsi in un qualunque istante in caso siano rilevati malfunzionamenti. Di conseguenza, non richiede particolari precauzioni, anche se è consigliabile effettuarla quando si ha il via libera sulle correnti principali e inserendo una fase di tutto giallo, in modo da rispettare gli sgomberi.[Cam01]

1.2 Controllo semi-attuato

Alla base dei controllori semaforici semi-attuati vi è la definizione di un piano semaforico secondo le stesse modalità del controllo a tempi fissi. La differenza, rispetto a questi, sta nel fatto di poter modificare in tempo reale alcune caratteristiche del piano sulla base del traffico rilevato[HFB⁺13]. Questo tipo di controllo può essere adottato da intersezioni sia isolate sia coordinate ed è motivato dall'opportunità di poter considerare situazioni di traffico che deviano dalle medie utilizzate per la realizzazione del piano a tempi fissi. Tipicamente sono correnti di traffico con valori di flusso molto basso, a cui conviene dare il verde solo quando si presentano, oppure sono situazioni eccezionali, che cioè non si verificano ad ogni ciclo, dove si vuole favorire una particolare corrente, per esempio a causa della presenza di un mezzo pubblico.

Nella progettazione di un controllo semaforico semi-attuato vi sono tre istanti di tempo che è necessario definire:

- L'istante di esame delle richieste attivate;
- L'istante di rientro nel funzionamento normale;
- L'istante di ripristino delle richieste attivate.

L'esame della presenza di richieste viene fatto immediatamente prima della transizione alla fase attuata, così da raccogliere richieste fino all'ultimo momento utile e, se necessario, viene effettuata la transizione di fase. Il rientro al funzionamento normale è di interesse rilevante solo nel caso in cui l'attuazione avvenga con una commutazione fra piani semaforici e si traduce nell'individuare l'istante di commutazione dal piano modificato a quello base. La scelta di questo istante non è particolarmente vincolata ma deve essere trascorso il tempo necessario alla transizione tra la fase attuata e quella successiva. Infine, l'istante di ripristino delle richieste dipende dal bilanciamento tra le necessità di non perdere richieste lasciate insoddisfatte e di non reinserire, nel ciclo successivo, una fase non necessaria. Solitamente, questo istante coincide con l'inizio del giallo della fase attuata.[Cam01]

1.2.1 Cambio di fase

Il numero di fasi del ciclo viene mantenuto ma è possibile cambiarne la sequenza a seguito di una precisa richiesta. Per esempio, questo tipo di modifica può essere impiegato per faci-

litare il trasporto pubblico in caso il tempo di arrivo previsto non coincida con un possibile verde del semaforo.

Al fine di ridurre i possibili effetti negativi indotti dall'alterazione della sequenza originale delle fasi, nei cicli successivi bisognerebbe mantenere il nuovo ordine. Inoltre, per evitare un'eccessiva instabilità, il cambio della sequenza deve essere impiegato solo se necessario.[HFB⁺13]

1.2.2 Adattamento tempo di offset

Si può scegliere di variare l'inizio di ogni fase di verde di un certo valore definito. Questo tipo di attuazione è di particolare interesse quando si vogliono coordinare i periodi di verde di singole fasi con l'ingresso dell'intersezione vicina.[HFB⁺13]

1.2.3 Richiesta di fase

Con la richiesta di fase si inserisce una fase all'interno della sequenza stabilita per abilitare il passaggio ad una corrente di traffico non sempre presente. Il principale vantaggio sta nel poter evitare perdite di tempo legate a fasi non strettamente necessarie a causa di un flusso di traffico molto basso. Combinando la richiesta di fase per il trasporto pubblico con l'adattamento della fase di verde, descritto in 1.2.4, si possono evitare molte interferenze.

Affinché questo tipo di attuazione sia favorevole è necessario ridurre al minimo i tempi di attesa delle correnti richiedenti, condizione che si può soddisfare prevedendo, all'interno del piano semaforico, molteplici istanti temporali in cui rendere disponibile la fase su richiesta. Inoltre, i momenti decisionali devono poter essere ritardati il più possibile, compatibilmente con le possibilità di coordinamento. Di fatto, il modo più semplice per implementarlo è prevedendo un unico punto di inserimento all'interno della sequenza delle fasi.[HFB⁺13]

1.2.4 Adattamento della durata del verde

La durata della fase di verde viene adeguata sulla base dell'effettiva richiesta da parte dei veicoli in arrivo, tipicamente estendendola oltre un certo valore di durata minima fino a raggiungere un valore massimo di durata, per questo si parla anche di allungamento della fase di verde. Il raggiungimento della durata massima del verde non è il solo modo per

terminare la fase, è anche possibile che questa venga fatta concludere anticipatamente a causa di richieste per la fase successiva. Esistono diversi criteri atti a stabilire se è necessario concludere il periodo di verde corrente al fine di favorire una corrente di traffico diversa. Questi sono:

- intervalli di tempo;
- gradi di occupazione;
- lunghezza delle congestioni;
- valori caratteristici dei modelli, per esempio tempi di percorrenza o fermate.

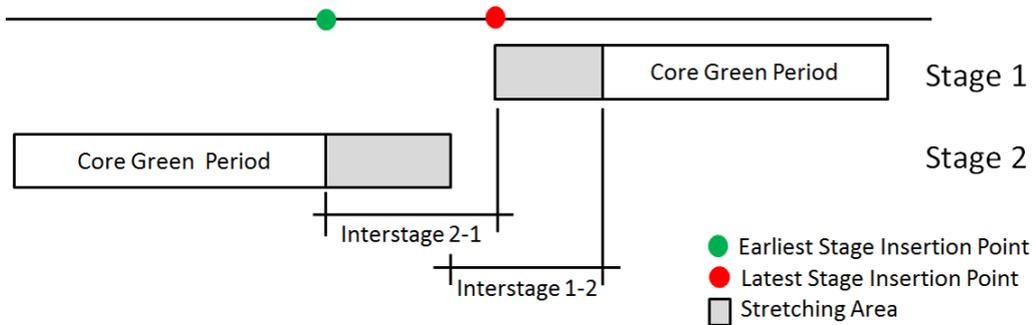
Per questo tipo di attuazione vengono impiegati dei rilevatori il cui compito è di calcolare la distanza, in termini temporali, tra un veicolo e quello che lo segue. Questa rilevazione, se entro un certo valore prefissato, consente di stabilire se è necessario prolungare il verde oltre la sua durata minima. Tipicamente, l'intervallo tra due veicoli usato è di 2-3 secondi ma può essere alzato in particolari condizioni stradali.

Come nel controllo a tempi fissi, anche nell'adattamento della fase di verde si hanno durate prefissate dei cicli semaforici, del numero e della sequenza delle fasi. L'unica differenza sta negli istanti di inizio e di fine dei periodi di verde, che possono variare, come già indicato, entro specifici limiti. Caratteristiche del piano semaforico sono il primo e ultimo istante possibile di inserimento delle fasi. Questi istanti temporali, di conseguenza, determinano di quanto può estendersi la durata del verde, come illustrato in figura 1.2 a fronte. Il primo punto di inserimento possibile indica l'istante temporale in cui la durata minima del verde della fase in corso è terminata mentre l'ultimo indica il termine dell'area di allungamento. Dal momento che la durata della transizione di fase tra quella corrente e quella successiva è fissa, vengono automaticamente definiti i due istanti di inserimento della fase successiva, contemporaneamente alla sua area di allungamento.[HFB⁺13]

1.3 Controllo completamente attuato

Nel controllo completamente attuato, o più semplicemente attuato, non esiste più un piano semaforico di riferimento ma i tempi di verde vengono determinati automaticamente in funzione del traffico reale. Resta comunque necessario rispettare i tempi di sicurezza e i

Figura 1.2: Punti di inserimento delle fasi e adattamento della durata del verde



[HFB⁺13]

vincoli legati all'inserzione e alla conclusione dei singoli periodi di verde. L'obiettivo di queste tecniche di controllo è di adattare il verde semaforico alle reali esigenze del traffico e dei singoli utenti, minimizzando gli aspetti negativi, per esempio le attese, che la regolazione semaforica introduce.

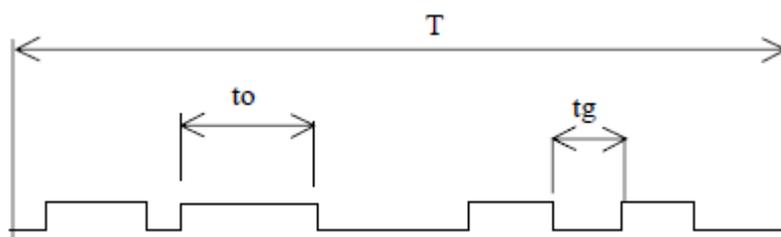
Tipicamente si sceglie di fare uso di un controllo completamente attuato nei soli casi in cui l'intersezione da controllare sia soggetta a correnti di traffico altamente variabili, situazione in cui un tradizionale controllo a tempi fissi risulterebbe ingestibile. Altri motivi per cui si vuole utilizzare questa tipologia di controllo sono la presenza di una corrente di traffico molto debole, la necessità di ridurre al minimo le interruzioni del verde sulla corrente principale, il miglioramento della coordinazione con gli impianti adiacenti o, più semplicemente, un'intensità di flussi di traffico talmente bassa da non giustificare l'introduzione di una regolazione semaforica, resa però necessaria per motivi di sicurezza. [Cam01]

Le metodologie di controllo attuato vengono distinte tra loro attraverso lo stato base dei segnali a cui i semafori vengono commutati in mancanza di richieste da parte delle correnti di traffico. Gli stati base sono tre:

- Sempre verde per la direzione principale;
- Tutto rosso / verde immediato;
- Mantenimento dell'ultimo stato.

Caratteristiche del piano non sono più il numero e la sequenza delle fasi o la durata del ciclo ma la durata minima e massima del verde per le varie fasi, il valore massimo dei tempi di

Figura 1.3: Sequenza d'impulsi di una spira magnetica



[Cam01]

attesa e i livelli di priorità tra le fasi. Si rende ovviamente necessaria la presenza di rilevatori del traffico per tutte quelle correnti che è necessario rilevare.

Il vantaggio ottenuto sta nella possibilità di assegnare alle diverse correnti di traffico il verde che gli spetta per il tempo strettamente necessario, nel rispetto dei tempi di sicurezza e di sgombero richiesti. Il controllo così ottenuto è capace di adeguarsi alle fluttuazioni imprevedute dei flussi di traffico e ridurre i tempi di attesa. Non è un tipo di controllo consigliato in caso di schemi di traffico definiti e regolari, per cui è più indicato ed economico un tradizionale piano semaforico a tempi fissi.[HFB⁺13]

1.3.1 Struttura di un regolatore attuato dal traffico

I dispositivi di misura del traffico più utilizzati sono i sensori a spira magnetica, che vengono installati nel manto stradale e rilevano la presenza di un veicolo attraverso la variazione del campo magnetico che questi producono al loro passaggio. I sensori sono poi collegati ai *rilevatori*, dispositivi in grado di trasformare la presenza del veicolo in un impulso elettrico utilizzato per ricavare i parametri utilizzati dalla logica di controllo del traffico.

Prendendo ad esempio il caso della spira magnetica, questa genera una sequenza d'impulsi (figura 1.3) all'interno di un dato periodo di campionamento T . Ogni veicolo, al passaggio sulla spira, genera un impulso di durata t_o pari al tempo in cui il veicolo è presente sul sensore. Con queste prime due misure è possibile misurare il flusso della corrente veicolare, contando il numero di veicoli che attraversano la spira all'interno dell'intervallo di campionamento, e il *tasso di occupazione* τ della spira, calcolato come

$$\tau = \frac{\sum t_{o_i}}{T} \quad (1.24)$$

Il tasso di occupazione tende all'unità in presenza di coda e, affiancato al conteggio dei veicoli, fornisce una stima dello stato della circolazione. Un'altra grandezza di interesse è il gap temporale tg tra due veicoli consecutivi.[Cam01]

In generale, i sensori utilizzati per la rilevazione delle informazioni sul traffico devono trovarsi in zone ben precise, nel rispetto dei seguenti vincoli:

- Non devono essere installati in aree dove è frequente il cambio di corsia;
- Se usati per la registrazione del volume del traffico, devono trovarsi in una zona dove questo scorra liberamente;
- Se usati per la registrazione del livello di occupazione, devono trovarsi in aree a rischio congestione ma non dove i veicoli possano fermarsi in condizioni normali.

Buona norma è anche sfruttare più zone di misurazione o quante più condizioni possibili per il processo decisionale ed evitare di cambiare troppo frequentemente i segnali.

Dato l'impatto e il costo dei sensori, a seconda del tipo di intersezione si possono avere diverse situazioni. Nel caso di strade importanti con flusso di traffico uniforme che intersecano strade secondarie poco utilizzate, si preferisce mantenere il verde sulla direttrice principale e utilizzare i sensori nelle sole strade intersecanti, al fine di dargli il verde quando viene rilevata una richiesta. L'uso dei sensori su tutte le direzioni di un incrocio si ha solo se i volumi di traffico sono relativamente simili ma a distribuzione variabile.[HFB⁺13]

Ogni regolatore semaforico che applichi un controllo attuato può avere, inoltre, uno o più piani semaforici di riserva da sfruttare in caso vi siano dei malfunzionamenti ai sensori usati per la rilevazione del traffico.[Cam01]

La logica di controllo è la componente del sistema in cui risiedono le condizioni logiche che legano tra loro le informazioni caratteristiche del traffico, usate da quel particolare regolatore semaforico, e i vincoli temporali che caratterizzano il piano semaforico. Assieme a questo devono essere definite le priorità con cui risolvere i conflitti tra richieste conflittuali. La logica di controllo è definibile sia attraverso una descrizione verbale, nella maggioranza dei casi, sia attraverso l'ausilio di diagrammi di flusso. Infine, è necessario sottoporla a test prima che venga installata nell'intersezione da controllare, così da far emergere immediatamente eventuali errori di realizzazione. Solo successivamente si può verificare su strada se la logica sviluppata produce gli effetti desiderati sul controllo del traffico.

Nella realizzazione di un regolatore semaforico con controllo attuato è necessario che siano definite le durate minime e massime dei verdi, i tempi di offset e le regole utilizzate per la commutazione dei gruppi di segnali. In aggiuntam si possono predeterminare il numero massimo di fasi, il massimo periodo di rosso o di blocco a seguito di una richiesta e le sequenze ottimali di fasi per diverse situazioni. Si può inoltre prevedere che solo alcuni elementi del piano semaforico siano modificabili in alcuni precisi momenti dell'arco della giornata.[HFB⁺13]

1.3.2 Tipologie di controllo attuato

Le tecniche di controllo attuato esistenti sono molto diversificate tra loro [Cam01] ma, di base, vanno ad agire su tre fattori:

- sequenza variabile delle fasi;
- durate variabili dei verdi per ogni fase;
- durate variabili dei cicli semaforici causate dalla variabilità dei verdi.

Per quanto riguarda la modifica della sequenza delle fasi, esistono due tecniche di controllo principali: nel primo caso si possono realizzare dei piani con diverse e alternative sequenze di fasi, nel secondo caso il piano viene costruito attraverso la combinazione di fasi dipendenti dal traffico.

Le modalità di controllo legate all'adattamento del periodo di verde si basano sul controllo dei gap temporali tra i veicoli, sull'uso di metriche legate al grado di occupazione dell'intersezione oppure attraverso la rilevazione delle situazioni di congestione.[HFB⁺13] Ricordando che il periodo di verde deve comunque rispettare dei vincoli di durata minima e massima, sono ora presentati alcuni dei criteri usati per stabilire quando è necessario terminarlo prima del raggiungimento del suo valore massimo.

1.3.2.1 Criterio a densità

Si basa sul gap temporale tra due veicoli. Viene quindi stabilito un valore soglia Δt , rappresentante il massimo tempo che può intercorrere tra il passaggio di due veicoli sul sensore. Se il gap tra veicoli è più basso di Δt , il verde viene prolungato di pari valore, altrimenti viene terminato.

1.3.2.2 Criterio a diminuzione di densità

Il criterio è simile al precedente ma il valore soglia non è costante. Col passare del tempo, il gap tra due veicoli deve essere sempre minore per garantire il mantenimento del verde. L'idea è quella di concedere prolungamenti più elevati all'inizio, quando la coda si muove più lentamente, e ridurli mano a mano che, in teoria, la coda si dirada.

1.3.2.3 Criterio a volume-densità

Il verde è compreso tra un valore minimo e uno massimo e mantenuto sulla base del gap tra i veicoli e dell'occupazione dei sensori di rilevazione del traffico.

1.3.2.4 Criterio ad occupazione-densità

Vengono utilizzati due diversi andamenti: uno in salita, legato al periodo di assenza dei veicoli, ed uno in discesa, legato al tempo di occupazione dei sensori. Con questo criterio si riesce a considerare meglio il passaggio di veicoli lenti o pesanti.[Cam01]

1.3.2.5 Controllo basato sulle fasi

Questo tipo di controllo attuato non va ad agire sulle durate dei verdi ma sulla sequenza delle fasi. Ai segnali del gruppo associato alla fase sono associate delle condizioni legate al cambio di fase. Quando compare una richiesta da parte di un flusso di traffico bloccato, tutte queste condizioni vengono controllate e, se tutte sono realizzate, è possibile passare ad una nuova fase. Tra due fasi si ha sempre una transizione della durata necessaria a far rispettare i tempi di sicurezza previsti.

1.3.2.6 Controllo basato sui gruppi di segnali

Questa modalità di controllo elimina il concetto di fase e necessita della sola dichiarazione di compatibilità o conflittualità tra i gruppi di segnali. Rispetto al controllo precedente, è molto più flessibile:

- La terminazione del verde di un gruppo di segnali non è legata a quella di altri gruppi;

- L'inizio del verde per una corrente richiedente è vincolata alle sole condizioni di terminazione associate ai segnali verdi conflittuali e alla presenza di richieste a maggiore priorità;
- Tutte le possibili combinazioni di movimenti compatibili sono ammesse.

Venendo a mancare la definizione delle fasi, non si ha più la presenza delle transizioni di fase a garantire, per esempio, il soddisfacimento dei tempi di sicurezza e sgombero, per cui, prima di effettuare una commutazione di segnale, è necessario assicurarsi che ogni vincolo sia stato rispettato. Ad esempio, per le correnti di traffico attualmente libere è necessario che siano stati garantiti i tempi minimi di verde e controllare che non abbiano bloccato per un tempo eccessivo eventuali direzioni antagoniste che richiedono il verde. Dualmente, per i gruppi di segnali bloccati va verificato che siano presenti richieste e, qualora ce ne siano da parte anche di altri gruppi, quale sia la più prioritaria.[HFB⁺13]

Capitolo 2

Approccio Self-Organized

Il settore delle intelligenze artificiali legato alla *swarm intelligence* studia sistemi di agenti semplici che, cooperando tra loro, mostrano un comportamento emergente definito intelligente. L'idea a cui si fa riferimento è la metafora delle colonie di insetti sociali, per esempio formiche o api, dove ogni individuo dimostra di avere una propria autonomia ma, nonostante questo, l'intera colonia ha una sua organizzazione senza la presenza esplicita di un supervisore sovrastante tutti gli individui.

Il motivo per cui ci si ispira a questo modello è perché una colonia di insetti sociali è, a tutti gli effetti, un sistema decentralizzato capace di risolvere problemi. Questi, tipicamente, sono la ricerca di cibo, l'espansione o costruzione del nido, la risposta a stimoli ed eventi esterni, tra cui pericoli o attacchi di cui bisogna avvertire l'intera colonia. Queste sono sfide che hanno diverse controparti nel mondo dell'ingegneria e dell'informatica e l'aspetto più interessante degli insetti sociali è la loro capacità di risolvere, in maniera flessibile e robusta, questi problemi. La flessibilità consente di adattarsi al mutare dell'ambiente circostante, quindi di adeguarsi a possibili situazioni impreviste, mentre la robustezza consente alla colonia la possibilità di funzionare anche in caso di fallimento di alcuni individui. Infine, gli insetti hanno capacità limitate, per cui è facile rappresentarli tramite dei semplici agenti autonomi capaci di auto-organizzarsi e di comunicare tra loro.

I sistemi che si basano su questa metafora sono costituiti da semplici agenti interagenti tra loro col fine di risolvere un problema che interessa l'intero gruppo. Il principale problema legato a questo approccio sta nel fatto che la soluzione al problema emerge dal comportamento e dalle interazioni degli agenti, non è cioè "programmabile". La sfida sta quindi nell'individuare quali comportamenti inserire negli agenti e strutturare le interazioni

tra loro in modo da ottenere la soluzione al problema richiesto.[BDT99]

2.1 Concetto di auto-organizzazione

L'auto-organizzazione è un insieme di meccanismi dinamici in cui le interazioni tra componenti di un sistema fanno emergere comportamenti strutturati a livello globale. La cooperazione a basso livello si basa esclusivamente sull'uso di informazioni disponibili localmente, senza la guida di uno schema globale imposto esternamente che, invece, ne è il risultato. Per ottenere auto-organizzazione sono necessari:

feedback positivo detto anche meccanismo di amplificazione, è un tipo di interazione che promuove la formazione di strutture o che rafforza comportamenti comuni;

feedback negativo controbilancia il feedback positivo e funge da stabilizzatore, tipicamente è rappresentato da azioni di saturazione, esaurimento o competizione;

fluttuazioni più precisamente, la loro amplificazione. Sono fluttuazioni, per esempio, le decisioni casuali di deviare dal comportamento prevalente o l'introduzione di errori. Le fluttuazioni sono fondamentali per aprire strade verso nuove soluzioni e fungono da nucleo per la formazione di nuove strutture;

interazioni multiple la capacità degli individui di creare strutture auto-organizzate non è sufficiente, è necessario che questi siano anche in grado di sfruttare i risultati, ovvero le strutture, ottenuti da altri oltre ai propri.

Le interazioni tra gli agenti del sistema possono essere di natura diretta o indiretta. Il secondo tipo è quello più interessante, in quanto la comunicazione non avviene tramite un ovvio scambio tra due entità, come nel caso diretto, ma attraverso l'interazione con l'ambiente circostante[BDT99]. Un meccanismo di organizzazione indiretta, sfruttato anche per i controllori semaforici di COLOMBO, è la *stigmergia*, forma di comunicazione che prevede la modifica dell'ambiente da parte di un agente e una reazione al cambiamento da parte degli altri individui costituenti il sistema. Un esempio di stigmergia lo si ha nelle colonie di formiche, le quali si coordinano attraverso l'uso del feromone: questa sostanza viene rilasciata lungo il percorso effettuato da una formica dal formicaio alla fonte di cibo, formando una traccia che viene seguita dalle altre. Occasionalmente, altre formiche potranno divergere

da questo percorso cercando strade diverse per raggiungere la stessa fonte di cibo, ovvero lo stesso obiettivo. La scelta tra i diversi percorsi, per gli altri individui, è automatica: il feromone è una sostanza che evapora nel tempo e, di conseguenza, la via con la traccia più intensa indica la via più breve verso il cibo, in quanto i più frequenti passaggi delle formiche rafforzano la traccia che, come precedentemente detto, altrimenti evaporerebbe.[HFB⁺13]

Attraverso l'interazione tra gli agenti, il sistema riceve un feedback che lo regola, adattandosi a situazioni impreviste e arrivando dinamicamente alla soluzione cercata per il problema proposto. Lo stesso risultato, ottenuto con questo approccio, può essere raggiunto anche sfruttando un sistema centralizzato, ma la complessità risultante è estremamente più elevata. Una semplice definizione proposta per il concetto di sistema auto-organizzante è la seguente:

Un sistema descritto come auto-organizzante è uno i cui elementi interagiscono al fine di raggiungere dinamicamente una funzione o comportamento globale.[CGD08]

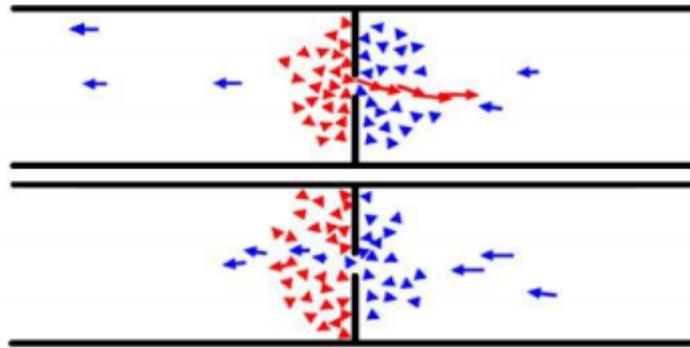
2.2 Impianti semaforici auto-organizzati

L'obiettivo del progetto COLOMBO consiste nella realizzazione di sistemi semaforici intelligenti, ispirati dalla ricerca nell'ambito della swarm intelligence. Sono stati sviluppati diversi algoritmi di controllo semaforico a livello microscopico, il cui funzionamento è sempre dipendente dalle informazioni fornite dal traffico. Queste diverse modalità sono complementari le une alle altre e prendono il nome di *policy*, ad indicare il fatto che non sono state ideate per venir impiegate singolarmente ma che verranno sfruttate, a livello macroscopico, da un unico algoritmo.[HFB⁺13]

Le modalità di controllo tradizionali descritte nel capitolo 1 mirano ad ottimizzare l'uso della rete stradale da parte di differenti tipologie di utenza. Tuttavia, un coordinamento ottimale dei flussi veicolari lo si riesce ad ottenere fintanto che questi rispecchiano le condizioni medie usate per la progettazione del piano semaforico. Con i sistemi di controllo attuato si riesce ad ottenere una gestione del traffico più flessibile ma, tipicamente, sono centralizzati e comportano grande complessità.

L'approccio auto-organizzante mira ad ottenere lo stesso risultato di coordinazione globale della rete semaforica senza l'uso di unità centrali ma, diversamente, affidandosi al-

Figura 2.1: Flussi di pedoni che si auto-organizzano



[LH08]

l'interazione locale tra i singoli semafori e i veicoli che li avvicinano[CGD08]. Quello proposto dal progetto COLOMBO, descritto in 2.3 a pagina 38, non è il primo.

2.2.1 Auto-organizzazione ad assegnazione di priorità

Questo approccio, elaborato da [LH08], si ispira ad un precedente modello di auto-organizzazione dei flussi pedonali in prossimità di un collo di bottiglia. Come illustrato in figura 2.1, si osserva un comportamento oscillatorio emergente risultante dalle diverse pressioni dei due gruppi di persone in attesa ai due lati della strettoia. Più pedoni si accalcano, più è alta la pressione, la quale si riduce quando le persone riescono ad attraversare il collo di bottiglia. La direzione di attraversamento si alterna in base a quale dei due gruppi esercita la maggior pressione, di fatto come se fosse presente un semaforo a regolare il transito. La stessa idea è stata applicata ai veicoli e al controllo semaforico, ideando una modalità di controllo che combina una strategia di ottimizzazione con una stabilizzatrice.

2.2.1.1 Strategia di ottimizzazione

Ad ogni flusso veicolare accedente l'intersezione è assegnato un indice i e la pressione esercitata è data da un indice di priorità dinamico $\pi_i(t)$. Il flusso a priorità maggiore, ovvero quello che deve essere servito, è memorizzato come

$$\sigma(t) = \arg \max_i \pi_i(t) \quad (2.1)$$

Il passo successivo è stabilire una formula per l'indice di priorità $\pi_i(t)$ che minimizzi il tempo di attesa totale, limitandosi a considerare i soli veicoli il cui tempo di attesa futuro dipende direttamente dall'attuale configurazione del semaforo. Viene inoltre fatta l'ipotesi semplificativa che tutte le correnti di traffico siano in conflitto tra loro, ovvero che venga servito solo un flusso alla volta. L'ordine ottimale di servizio è determinato confrontando le priorità di coppie di flussi in competizione. Quest'osservazione consente di ottenere una definizione per π_i partendo dallo studio di solo due flussi antagonisti, denominati 1 e 2. Al tempo corrente t , si indica il tempo di setup¹ delle due correnti come τ_1 e τ_2 , il numero previsto di veicoli con \hat{n}_1 e \hat{n}_2 e il tempo di verde necessario per questi con \hat{g}_1 e \hat{g}_2 . Ipotizzando che si stia servendo il flusso 1 ($\sigma = 1$), il controllore semaforico ha due scelte:

1. completare il servizio per il flusso 1 prima di servire il 2;
2. servire immediatamente il flusso 2, con il costo di un tempo di setup extra per tornare a servire l'1.

Calcolando l'aumento complessivo dei tempi di attesa pronosticati per ognuna delle due opzioni e confrontandole, si può effettuare la scelta ottima. Nello specifico, è ottimale la prima opzione se

$$(\tau_1 + \hat{g}_1) \hat{n}_2 < \Delta \hat{w}_1 + (\tau_2 + \hat{g}_2) \hat{n}_1 \quad (2.2)$$

dove $\Delta \hat{w}_1$ indica l'incremento nei tempi di attesa previsti per il flusso 1 causato dall'interruzione prematura data dal secondo caso. Il primo termine della disequazione mostra l'incremento totale dei tempi di attesa causato dalla scelta di completare il servizio per il flusso 1, il secondo termine, invece, riporta l'esito della scelta alternativa. Da questa condizione si può determinare la funzione per ottenere il valore dei due indici di priorità π_i :

$$\pi_i = \frac{\hat{n}_1}{\tau_1 + \hat{g}_1} > \frac{\hat{n}_2}{\Delta \hat{w}_1 / \hat{n}_1 + \tau_2 + \hat{g}_2} = \pi_2 \quad (2.3)$$

Di particolare interesse è l'osservazione di queste due funzioni: l'indice di priorità per il flusso 1 dipende dalle sole variabili ad esso legate. Diverso è il caso dell'indice del flusso 2, che è legato al primo dal termine $\Delta \hat{w}_1 / \hat{n}_1$, il cui significato è dato dalla penalità che si ha nel terminare prematuramente il servizio del flusso 1. Il termine può essere generalizzato in $\Delta \hat{w}_\sigma / \hat{n}_\sigma$, ad indicare la generica penalità legata alla terminazione anticipata del flusso

¹Con tempo di setup ci si riferisce al tempo necessario per dare il verde alla corrente per cui è calcolato.

correntemente servito, da cui la definizione del termine

$$\tau_{i,\sigma}^{pen} = \begin{cases} \Delta \hat{w}_\sigma / \hat{n}_\sigma & \text{se } i \neq \sigma \\ 0 & \text{se } i = \sigma \end{cases} \quad (2.4)$$

potendo così introdurre una formulazione generalizzata per l'indice di priorità π_i come

$$\pi_i = \frac{\hat{n}_i}{\tau_{i,\sigma}^{pen} + \tau_i + \hat{g}_i} \quad (2.5)$$

L'indice di priorità dipende quindi dal numero previsto di veicoli \hat{n}_i che dovranno essere serviti durante il periodo di tempo $\tau_i + \hat{g}_i$, consentendo di prendere in considerazione una stima della situazione futura, e i tempi persi a causa di possibili cambi di servizio anticipati (il termine $\tau_{i,\sigma}^{pen}$).

La strategia così delineata è in grado ottenere un'ottimizzazione a livello locale, ovvero dei singoli semafori, ma da sola non garantisce l'ottimalità globale della rete semaforica, per cui è necessario introdurre una strategia stabilizzatrice.

2.2.1.2 Strategia stabilizzatrice

In ambito di controllo semaforico, si dice che un singolo impianto sia *stabile* se è capace di mantenere limitate le lunghezze delle code, con l'ovvia osservazione che il traffico non superi la capacità dell'intersezione. La sola strategia di ottimizzazione, per esempio a causa di commutazioni troppo frequenti, può portare ad un uso non ottimale di tale capacità, anche se il livello di traffico non è critico. Lo scopo della strategia stabilizzatrice è di supervisionare il funzionamento dell'impianto semaforico mantenendolo stabile. Si definisce un insieme ordinato di priorità Ω , il quale contiene gli indici i dei flussi selezionati secondo il seguente criterio:

- un flusso i viene inserito in Ω appena viene superata una soglia n_i^{crit} critica di veicoli in attesa di essere serviti;
- un flusso i viene rimosso da Ω quando la sua coda è svuotata.

Gli elementi di Ω sono serviti FIFO² e la strategia di controllo deve sempre fare riferimento al primo elemento dell'insieme per definire il flusso più prioritario. La combinazione delle

²First In First Out, ovvero il primo che arriva è il primo ad essere servito.

due strategie porta quindi a

$$\sigma = \begin{cases} \text{testa } \Omega & \text{se } \Omega \neq \emptyset \\ \arg \max_i \pi_i(t) & \text{altrimenti} \end{cases} \quad (2.6)$$

L'azione stabilizzatrice entra in gioco solo quando la strategia di ottimizzazione fallisce nel mantenere le code sotto una certa lunghezza. Quindi, finché l'ottimizzazione mantiene il suo comportamento desiderato, la strategia stabilizzatrice non interverrà mai.

2.2.2 SOTL (Self-Organizing Traffic Lights)

Il metodo SOTL, proposto in [CGD08], prevede che ogni impianto semaforico tenga traccia del numero di veicoli in arrivo, sulle sole direttrici con rosse, sfruttando un contatore k_i , resettato ogni volta che la luce a cui è associato commuta a rosso. Al raggiungimento di un valore soglia θ si ha il cambio di servizio, portando a giallo e, successivamente, rosso la direttrice attualmente servita. Lo scopo è di dare il verde ad una corrente di traffico solo quando questa commutazione consenta ad un nutrito gruppo di veicoli di passare in blocco, inducendo un implotonamento del traffico.

Il vantaggio di avere veicoli organizzati in plotoni sta nella logica di commutazione dei semafori: se un gruppo di macchine sufficientemente numeroso si muove compatto dirigendosi verso un semaforo, questo, rilevandolo in anticipo, può commutare a verde la direttrice di approccio del plotone e consentirgli di attraversare l'intersezione senza fermarsi, facendo così emergere un *corridoio verde*. Inoltre, l'organizzazione in plotoni implica la presenza di vuoti all'interno del flusso veicolare, i quali possono essere riempiti da plotoni di correnti antagoniste, riducendo le interferenze tra le due correnti.

Per evitare una frequenza eccessivamente elevata della commutazione di stato da parte del semaforo, viene introdotto un vincolo temporale di durata minima φ_{min} della configurazione delle luci. Infine, per facilitare ulteriormente l'implotonamento, il controllore semaforico, prima di servire un'altra direttrice, deve verificare se l'incrocio è attualmente attraversato da veicoli, così da non spezzare il plotone. Di conseguenza, una luce rossa non diventa mai verde se è presente almeno un veicolo a distanza ω dall'intersezione a cui è attualmente concesso l'attraversamento. Questa condizione, come già detto precedentemente, serve a mantenere coesi i plotoni ma è critica in presenza di traffico intenso, situazione in cui un gruppo di veicoli molto grande può impedire la commutazione dei segnali semafori-

ci per un periodo di tempo esteso. Per prevenire questo effetto collaterale, viene introdotta una seconda condizione, alternativa a quella appena descritta, che consente di spezzare un plotone se vi sono più di μ veicoli in arrivo sulla direttrice con luce verde.

Dato che l'intero sistema si basa esclusivamente sul traffico rilevato, l'impianto può trovarsi in condizione di consentire il verde sempre alle stesse direttrici di traffico se, su quelle antagoniste, non si presentano mai veicoli.

2.3 Auto-organizzazione in COLOMBO

La visione dell'infrastruttura semaforica è quella di un sistema auto-organizzante strutturato come segue. Fondamentalmente, l'architettura è quella di un sistema di controllo digitale ed è quindi presente un ciclo continuo all'interno del quale gli agenti iterano incessantemente. Le fasi di questo ciclo sono: percezione, valutazione, azione.

Le fonti di informazione sono i sensori posizionati sulla rete stradale e i dati così raccolti sono forniti ad una funzione di stimolo, la quale ha il compito di selezionare, in maniera probabilistica, la più efficace tra le possibili policy presenti nel sistema da mettere in esecuzione. Le diverse politiche tra cui poter scegliere sono definite per ottenere un'ottimale gestione del traffico per una particolare sua configurazione, motivo per cui sono state definite come complementari. Inoltre, la struttura della funzione di stimolo varia nel tempo: il sistema è pensato per evolversi e apprendere autonomamente, adattandosi al comportamento degli utenti della rete stradale. Sono così individuati i due livelli di controllo: uno di basso livello, legato alle politiche con cui viene regolato il traffico, ed uno ad alto livello, il cui scopo è selezionare la corretta policy a basso livello.

Gli agenti che costituiscono il sistema sono i controllori semaforici e tipicamente regolano un'intersezione ciascuno. Tra di loro non vi è alcuna comunicazione diretta, l'unica forma di coordinamento presente è stigmergica e tra semafori di intersezioni adiacenti. In questo modo la collaborazione avviene efficientemente e senza la necessità di una complessa infrastruttura centralizzata, la quale risulterebbe computazionalmente troppo onerosa e difficilmente ottimizzabile.

La diretta conseguenza della decentralizzazione del sistema è che le uniche fonti di informazioni dei singoli controllori semaforici sono le strade che si intersecano nell'intersezione controllata. Per ragioni che saranno spiegate in seguito, viene mantenuta una distinzione tra i flussi di traffico entranti l'intersezione e quelli uscenti. Infine, un sistema

così strutturato è di più facile implementazione e più reattivo alla variabilità, tipicamente elevata, delle condizioni del traffico veicolare.

Vengono abbandonati i concetti tradizionali di temporizzazioni e di fasi dei semafori, in quanto la commutazione di stato avviene a seguito della percezione di una diversa configurazione del traffico. In questo modo i tempi di reazione sono molto bassi e, in confronto alle tecniche tradizionali, quest'approccio consente una maggiore flessibilità nel dare il verde a diverse correnti di veicoli.[HFB⁺13]

2.3.1 Politica Swarm di alto livello

Come verrà illustrato in 2.3.2 a pagina 42, ogni politica di basso livello ha prestazioni differenti in base alle condizioni correnti del traffico. La politica *Swarm* di alto livello ha il compito di scegliere, tra queste, quella più corretta sulla base sia del traffico sia delle prestazioni ottenute dalle policy nell'intersezione controllata.

Come già accennato precedentemente nella pagina precedente, si fa uso di una funzione di stimolo il cui input è calcolato sulla base del traffico rilevato dai sensori. La definizione di traffico alto o basso è molto relativa e legata alle corsie e al loro numero, alla geometria dell'intersezione e ai percorsi seguiti dai veicoli. Una misurazione corretta dei flussi veicolari deve inoltre essere insensibile a picchi di breve durata ma capace di far reagire il sistema rapidamente a modifiche più persistenti. Per esempio, una coda di auto data da un matrimonio è una situazione che si risolve in tempi ragionevolmente brevi, mentre è molto più intenso il flusso veicolare che si ha in uscita da un teatro o uno stadio al termine di un evento, che può durare anche per svariati minuti. Per questi motivi è necessario astrarre il livello di traffico ricorrendo ad una metafora naturale: il feromone. Come illustrato nell'introduzione al capitolo, a pagina 32, il feromone ha la proprietà di essere additivo: quanti più individui seguono lo stesso percorso tanto più elevato sarà il livello di feromone. Questo, inoltre, evapora e più sono gli individui a percorrere lo stesso percorso meno avrà capacità di evaporare.

Su questa base si è scelto di utilizzare i livelli di feromone per rappresentare i livelli dei flussi di traffico: i veicoli, muovendosi, lasciano una traccia di feromone (virtuale). Nel caso di corsie sgombre, questo evaporerà mentre, al contrario, tenderà ad accumularsi velocemente in presenza, per esempio, di code ferme ad un semaforo. I parametri che regolano l'accumulo e l'evaporazione del feromone sono trattati alla sezione 4.1 a pagina 57. Co-

me già accennato a pagina 38, i flussi di traffico vengono distinti in entranti ed uscenti, di conseguenza la stessa distinzione viene fatta per i livelli di feromone. Diversi sono anche i meccanismi che regolano il loro accumulo: per le corsie in ingresso il feromone aumenta proporzionalmente al numero di veicoli per corsia mentre su quelle in uscita si accumula proporzionalmente alla differenza tra velocità massima consentita e velocità media corrente. La motivazione alla base di questa distinzione sta nel fatto che, per quanto affollate possano essere le direttrici in uscita dall'intersezione, se il traffico scorre velocemente non si hanno problemi di congestione. Dualmente, per le direttrici in ingresso, ci si aspetta che i veicoli si fermino al semaforo e quindi è di maggiore interesse il numero di mezzi presenti piuttosto che la loro velocità. I livelli di feromone delle corsie vengono infine mediati tra loro e usati come valori di input per la politica Swarm, ovvero per le funzioni di stimolo delle diverse policy di basso livello.

La funzione di stimolo è lo strumento utilizzato per calcolare l'interesse attuale per l'esecuzione della particolare politica, a cui lo stimolo è associato, sulla base dei livelli di feromone correnti. La funzione mappa la rispettiva policy in uno spazio $feromone_{in} \times feromone_{out}$, viene tarata sperimentalmente per individuare i livelli di feromone a cui la politica risponde meglio, ovvero le condizioni di traffico per cui vengono minimizzate particolari metriche d'interesse come l'attesa media dei veicoli o il ritardo medio subito da questi, e fornisce come risultato un valore reale compreso tra 0 e 1. Il tipo di funzione scelto è una gaussiana centrata nel punto in cui la politica ottiene i migliori risultati e i parametri che ne determinano altezza e posizione sono illustrati successivamente nella sezione 4.1. La sua formula è

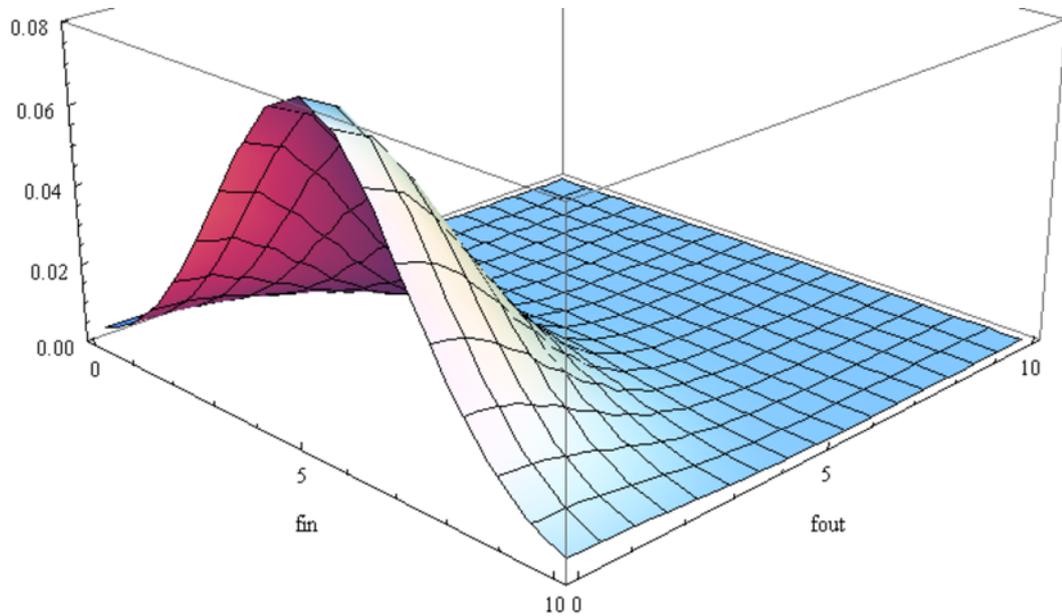
$$s = cox * e^{coxExp} \quad (2.7)$$

con

$$coxExp = \frac{-coxExp_{in}(f_{in} - offset_{in})^2}{divisor_{in}} + \frac{-coxExp_{out}(f_{out} - offset_{out})^2}{divisor_{out}} \quad (2.8)$$

2.3.1.1 Selezione della politica di basso livello

Ad ogni istante di valutazione la politica Swarm aggiorna i valori di feromone presente sulle corsie controllate e, contemporaneamente, valuta la probabilità di cambiare la policy in esecuzione. Se il test probabilistico da esito positivo, vengono calcolate tutte le funzioni di stimolo in accordo agli attuali livelli di feromone.

Figura 2.2: Funzione di stimolo

[HFB+13]

Anche la scelta della politica è soggetta a probabilità, dal momento che il valore della funzione di stimolo è interpretato come probabilità di selezionare la policy a cui è associata. Su questa scelta, entra in gioco il fattore evolutivo del sistema: ad ogni stimolo viene applicato, prima della selezione della politica, una soglia t che guidi la decisione verso quelle policy con soglia più bassa. Il valore, diverso per ogni politica, è variabile nel tempo: viene decrementato quando la policy è scelta in maniera stabile e, dualmente, incrementato quando non viene scelta. Questo meccanismo consente di scegliere sempre la politica che si apprende essere la più adatta per l'incrocio controllato e prende il nome di *rinforzo*. La soglia è però confinata tra un valore minimo, non nullo, e uno massimo, per cui non sarà mai impossibile selezionare una politica, per quanto basso o quasi nullo possa essere il risultato fornito dalla sua funzione di stimolo. Calcolato lo stimolo, t lo modifica attraverso la definizione del valore

$$\theta = \frac{s^2}{s^2 + t^2} \quad (2.9)$$

Ricapitolando, la determinazione della selezione di una policy di basso livello da parte di *Swarm* parte dai livelli di feromone, i quali vengono dati in input alle funzioni di stimolo, coi relativi valori di soglia, che forniscono una probabilità di selezione della politica associata,

fattore che viene rinforzato nel tempo.[HFB⁺13]

2.3.2 Politiche di basso livello

Mentre *Swarm* ragiona ad alto livello e sul lungo periodo, determinando in che modo vada gestito il traffico veicolare, le diverse policy di basso livello si occupano di assegnare il verde alle diverse corsie e regolarne la durata. Queste politiche sfruttano ancora il concetto di sequenza di fasi, anche se è diverso da quello definito nelle modalità di controllo semaforico tradizionali, pur condividendone il procedimento di definizione.

Già con il controllo a tempi fissi definito nella sezione 1.1 a pagina 6 si parlava di possibili piani semaforici alternativi tra loro. Questo approccio è ripreso in COLOMBO, ma applicato alle sequenze di fasi, concetto rimpiazzato da quello delle catene.

Una catena è una sequenza base di fasi ed è l'elemento attraverso cui viene definita al logica del controllore semaforico. Caratteristica delle catene è che siano intercambiabili tra loro, motivo per cui non si ha più un concetto di ciclo semaforico nel senso tradizionale del termine. Ciò è possibile ragionando sullo stato di tutto rosso: in questo caso ogni direzione di traffico è bloccata, di conseguenza non esiste alcun vincolo che obblighi l'assegnazione del verde ad una piuttosto che ad un'altra corsia. La selezione, in questo caso, dovrebbe avvenire sulla base del traffico in modo da favorire il decongestionamento delle corsie più impegnate. Questa libertà di scelta non si ha, invece, quando sono presenti delle luci verdi o gialle, le quali pongono dei vincoli sulla sequenza delle fasi, lasciando però la possibilità di alterarne la durata³. Sono quindi stati illustrati tutti gli elementi necessari a definire una catena.

Una catena è una sequenza di fasi non alterabile, definita partendo da una fase *target*, presentante una qualunque configurazione di luci possibile a seguito di uno stato di tutto rosso, che si conclude con una fase di tutto rosso denominata *commit*. Tra queste due, sono definite altre fasi la cui sequenza, va ribadito, è fissata. Quella in *target* e *commit* non è l'unica distinzione possibile per le fasi, le quali sono ulteriormente classificate in *decisional* e *transient*:

decisional una fase la cui durata può essere regolata. Il criterio con cui avviene questa decisione è caratteristico di ogni politica e, tipicamente, la variazione avviene tra

³In realtà le durate delle luci gialle sono fissate, per cui solo le fasi che presentano esclusivamente luci verdi e rosse hanno durata regolabile.

un valore minimo e massimo derivanti da considerazioni di sicurezza per l'impianto semaforico che si sta realizzando;

transient una fase la cui durata è fissa. Tipicamente sono definite transient tutte le fasi contenenti luci gialle, la cui durata è definita da normative stradali.

Quando una catena viene selezionata, questa è eseguita per intero. Raggiunta la fase commit della catena corrente, la politica di basso livello in uso sceglie la successiva fase target, determinando indirettamente la successiva sequenza di fasi. Durante l'esecuzione di una catena, la policy controlla la tipologia della fase corrente. Se è di tipo decisional, viene valutata la condizione di regolazione della durata, caratteristica di ogni politica, e la decisione presa è mantenuta fino al successivo istante di valutazione. Diversamente, se la fase è di tipo transient questa viene fatta eseguire fino al termine della sua durata prefissata.

Come già ribadito in precedenza, ciò che distingue le diverse politiche è il criterio con cui viene decisa la regolazione della durata delle fasi decisional. Questa condizione sfrutta anche la definizione di un'altra soglia, posta sul prodotto tra il numero di veicoli in attesa ad un rosso ed il tempo di attesa trascorso e verificata per ogni corsia in ingresso a cui viene dato il rosso.

2.3.2.1 Default policy

La politica Default viene definita staticamente, in completa analogia ad un piano semaforico a tempi fissi. Le informazioni sul traffico non vengono considerate e, una volta messa in esecuzione, questa policy non altera l'ordine delle fasi fino a che non viene sostituita da un'altra politica di basso livello. Non esiste alcun criterio di regolazione delle fasi decisional, dal momento che non possono neanche venir definite per questa politica.

2.3.2.2 Request policy

Request policy è la più reattiva tra tutte quelle definite nel progetto COLOMBO. La catena corrente viene rilasciata⁴ appena un'altra catena supera la soglia. La durata della fase corrente e i parametri di durata minima e massima vengono completamente ignorati. Benché

⁴In realtà è la fase decisional che viene rilasciata, ovvero terminata. Viene comunque usato il termine rilasciare in associazione alle catene, anche se è improprio, dal momento che, per molte politiche, vi è la tendenza ad indicare come decisional la sola fase target.

questo non costituisca problema per la durata massima, nel solo caso non fossero presenti veicoli in attesa, è fonte di preoccupazioni che venga ignorata la durata minima della fase, rendendo di fatto questa politica non aderente alle norme di sicurezza.

La politica è indicata in casi di traffico molto basso, per esempio di notte, quando il numero di veicoli in arrivo all'incrocio è ridotto. Per evitare una commutazione continua delle luci semaforiche e ridurre l'entropia del traffico, è stabilita una soglia di durata minima, non nulla, per cui è obbligatorio mantenere stabile lo stato del semaforo. Questa soglia, inoltre, non coincide con la durata minima della fase ma è un parametro fisso del semaforo comandato con questa politica.

2.3.2.3 Phase policy

La catena corrente viene terminata appena un'altra supera la soglia, in analogia con la politica Request, ma nel rispetto della durata minima della fase corrente. Questo la rende meno reattiva della policy precedente, rispetto alla quale rilascia il verde meno spesso. Phase è una politica adeguata per un flusso di traffico medio-basso, che gli consente di non commutare eccessivamente lo stato delle luci.

Come accade con la politica Request, se nessuna catena supera la soglia stabilita il verde viene mantenuto per un tempo indefinito, ignorando la durata massima stabilita per la fase corrente.

2.3.2.4 Platoon policy

Obiettivo di Platoon è l'implonamento dei veicoli, ottenuto cercando di far passare quanti più veicoli possibili prima di terminare il verde. La catena viene rilasciata solo se si verificano queste tre condizioni:

1. la durata minima della fase corrente è terminata;
2. vi sono catene che hanno superato la soglia;
3. sulle corsie attualmente verdi non vi sono più veicoli oppure la durata massima della fase corrente è terminata.

In analogia a Request e Phase, anche Platoon non rilascia mai il verde se non vi sono altre catene che lo richiedono. La durata massima della fase è quindi considerata solo e

soltanto per terminare la fase corrente, anche in presenza di veicoli in arrivo, al fine di favorire veicoli in attesa. Ciò porta, in condizioni di traffico elevato, a concludere ogni fase decisional quando queste raggiungono il loro valore di durata massimo, rendendo la sua definizione di importanza critica per le prestazioni della policy.

2.3.2.5 Marching policy

L'uso di questa politica è indicato quando le condizioni di traffico, su ogni direttrice, sono troppo intense per poter prendere decisioni riguardanti le corsie in ingresso. Sono quindi possibili due approcci: nel primo si ricorre ad una durata statica per le fasi decisional, nel secondo vengono considerate le corsie d'uscita, negando il traffico verso le corsie troppo intasate.

2.3.2.6 Congestion policy

Quest'ultima politica è pensata per risolvere le congestioni sulle corsie in uscita, situazione che vede la presenza di veicoli in attesa all'interno dell'intersezione. Al fine di evitare la formazione di ingorghi, la catena corrente viene terminata normalmente rilasciando i verdi il prima possibile e, arrivati alla fase commit di tutto rosso, la scelta della catena successiva viene inibita fino alla risoluzione della congestione. Di fatto, congestion non sceglie mai una nuova catena, dato che la politica Swarm sceglierà una nuova policy di basso livello non appena la congestione terminerà.[HFB⁺13]

Capitolo 3

Parameter Tuning

Il problema della regolazione dei parametri si presenta comunemente durante la progettazione e l'analisi di un algoritmo. La necessità di trovare una giusta configurazione è legata alle performance ottenibili da questo e, tipicamente, lo stesso algoritmo può richiedere valori differenti per i suoi parametri a seconda del contesto in cui viene impiegato. Per questo motivo, è comunemente affrontato come un problema di ottimizzazione, dove la funzione obiettivo è proprio la performance registrata dall'algoritmo su un insieme predefinito di istanze usate per questo scopo[HHS07, Bir09].

L'operazione di ricerca è tipicamente impegnativa e può richiedere molto più del tempo necessario per la definizione e il test dell'algoritmo stesso, dal momento che viene condotta tipicamente a mano dallo sviluppatore, basandosi sul buonsenso e andando per tentativi. Questo tipo di approccio, oltre ad essere estremamente laborioso, implica la necessità di un'elevata conoscenza dell'algoritmo che si vuole regolare, restringendo al solo ideatore o a poche altre persone il cerchio di coloro capaci di completare questo compito.

Diverse strategie sono state proposte per la configurazione degli algoritmi, tra questi lo studio dei rapporti tra parametri, componenti e algoritmo, la metodologia delle superfici di risposta e l'ottimizzazione locale. Nel primo caso, il problema da affrontare è forse più complicato di quello di regolazione e richiede la comprensione degli effetti prodotti dai singoli parametri o componenti sull'algoritmo in cui vengono usati. Il secondo, invece, è concettualmente simile alla discesa del gradiente o ad una ricerca locale. La procedura consiste in una ricerca iterata all'interno dello spazio dei parametri, partendo da un punto casuale, ed effettuando un test delle performance delle configurazioni usate dopo un certo numero di iterazioni, confrontandole con quelle dei punti vicini. Dovendo definire un vici-

nato, è necessario definire una misura di distanza tra punti dello spazio delle configurazioni, compito problematico quando si ha a che fare con valori categorici, ovvero privi di un ordine intrinseco. L'esito del confronto tra le configurazioni prevede la scelta del punto a cui si è arrivati, se questo si comporta meglio dei suoi vicini, oppure una nuova ricerca partendo dalla configurazione, tra quelle del vicinato, con le prestazioni migliori. L'approccio a ottimizzazione locale, concettualmente simile, è intrapreso quando il numero di parametri da regolare è elevato. Il procedimento prevede la modifica del valore di un parametro alla volta, accettando nuove configurazioni fino a che producono miglioramenti sensibili nelle performance dell'algoritmo. La ricerca è protratta fino a che l'ulteriore modifica dei valori assegnati ai parametri non produce più miglioramenti prestazionali. Questo metodo, invece che condotto a mano, può essere automatizzato, dal momento che coincide con una ricerca locale nello spazio dei parametri. [ADL06, Bir09, HHS07]

3.1 Configurazione automatica di algoritmi

L'approccio a ricerca locale iterata non è l'unico possibile per la configurazione automatica degli algoritmi. In generale, si possono individuare due macro-gruppi di strategie possibili: regolazione condotta on-line e regolazione condotta off-line. Il caso d'interesse è il secondo e viene adottata la definizione di *tuning problem*¹ proposta in [Bir09] in riferimento alle *metaeuristiche*, termine con cui ci si riferisce a modelli algoritmici generali, i quali necessitano di essere regolati in maniera opportuna al fine di produrre un algoritmo funzionante. Per completezza, e per avere un termine di paragone, si accenna brevemente anche al caso della regolazione on-line.

3.1.1 Regolazione on-line

Una strategia spesso utilizzata, in diversi ambiti, riguarda la regolazione automatica in tempo reale. Gli algoritmi che soddisfano questo tipo di problema non sono direttamente collegati al tuning problem definito in 3.1.2 ma svolgono un compito simile. Queste strategie sono spesso fondate su tecniche di apprendimento automatico e sono tipicamente legate al concetto di rinforzo.

¹problema di regolazione.

Il punto in comune sta nella modifica dei parametri dell'algoritmo in uso al fine di migliorarne le prestazioni. Di diverso si ha che questo viene fatto mentre l'algoritmo è in esecuzione, rendendo l'approccio ideale solo per la risoluzione di una singola istanza, tipicamente complessa. La regolazione è ottenuta attraverso un feedback dato dall'algoritmo stesso, solitamente legato al risultato parziale ottenuto, che fornisce le informazioni necessarie alla modifica in tempo reale dei parametri.

3.1.2 Definizione del problema di regolazione dei parametri

Le componenti necessarie per la definizione del tuning problem sono:

Θ l'insieme delle configurazioni candidate;

\mathcal{I} l'insieme delle istanze, tipicamente infinito;

$P_{\mathcal{I}}$ misura di probabilità su \mathcal{I} . Si definisce $P_{\mathcal{I}}(i)$ come la probabilità che l'istanza i sia scelta per essere risolta;

$t : \mathcal{I} \rightarrow \mathbb{R}$ funzione che associa ogni istanza al tempo necessario alla sua risoluzione;

c variabile casuale rappresentante il costo della miglior soluzione trovata simulando l'istanza $i \in \mathcal{I}$ con la configurazione $\theta \in \Theta$ per $t(i)$ secondi;

$C \subset \mathbb{R}$ l'insieme dei possibili valori di costo c della miglior soluzione trovata simulando l'istanza i con la configurazione θ ;

P_C misura di probabilità su C . Si definisce $P_C(c|\theta, i)$ come la probabilità che c sia il costo della miglior soluzione trovata simulando l'istanza i con la configurazione θ per $t(i)$ secondi;

$\mathcal{C}(\theta)$ equivalente a $\mathcal{C}(\theta|\Theta, \mathcal{I}, P_{\mathcal{I}}, P_C, t)$, è il criterio da ottimizzare rispetto a θ e rappresenta una misura della desiderabilità della configurazione θ ;

T è il tempo totale disponibile per sperimentare con le configurazioni candidate date sulle istanze disponibili prima di effettuare la selezione delle configurazioni desiderate.

Si può quindi definire il tuning problem attraverso la tupla $\langle \Theta, \mathcal{I}, P_{\mathcal{I}}, P_C, t, \mathcal{C}, T \rangle$ e la sua soluzione è la configurazione $\bar{\theta}$ tale che

$$\bar{\theta} = \arg \min_{\theta} \mathcal{C}(\theta) \quad (3.1)$$

Va menzionato che una metaeuristica è, tipicamente, un algoritmo stocastico e, di conseguenza, differenti esecuzioni con la stessa istanza e la stessa configurazione producono differenti risultati, i quali hanno costi diversi. La definizione del problema di regolazione non pone vincoli sul tipo di statistica da usare per la determinazione del criterio di ottimizzazione \mathcal{C} , lasciando libertà di azione. In [Bir09] viene considerato il valore atteso di c , dato che è la misura tipicamente più adottata. La definizione è la seguente:

$$\mathcal{C}(\theta) = E_{\mathcal{I}, \mathcal{C}}[c] = \int c dP_C(c|\theta, i) dP_{\mathcal{I}}(i) \quad (3.2)$$

dove l'integrale è un integrale di Lebesgue. Non sempre si ha una misura esplicita delle probabilità P_C e $P_{\mathcal{I}}$, per cui il calcolo del valore atteso può essere stimato, per esempio, ottenendo diversi valori di c attraverso la simulazione di diverse istanze $i \in \mathcal{I}$ con la medesima configurazione θ^2 . Questa definizione può generare problemi nel caso esista, per una determinata configurazione θ , un'istanza i con probabilità $P_{\mathcal{I}}(i) > 0$ tale che sia impossibile trovare una soluzione ammissibile entro il tempo $t(i)$. In questi casi di fallimento, è pratica comune assegnare un costo $c = \infty$ ma questo impedirebbe la convergenza dell'integrale. Di conseguenza, è necessario assegnare valore infinito anche a $\mathcal{C}(\theta)$ o imporre un valore massimo $B(i)$ al costo c .

L'obiettivo del tuning problem è, quindi, trovare quella particolare configurazione che minimizza il costo, ovvero massimizza le prestazioni, della metaeuristica su un particolare insieme di istanze. Si possono individuare due fasi distinte:

fase di sviluppo corrisponde allo sviluppo e al tuning dell'algoritmo e si basa su situazioni pregresse e rappresentative dell'insieme delle possibili istanze \mathcal{I} su cui la metaeuristica verrà impiegata;

fase di produzione corrisponde all'impiego sul campo dell'algoritmo sviluppato.

²[LIDLSB11].

La necessità che le istanze utilizzate durante la fase di sviluppo siano rappresentative dell'insieme da cui vengono estratte si basa sull'ipotesi, tipicamente condivisa, che sia possibile generalizzare eventi futuri partendo dall'esperienza pregressa. Da questo punto di vista, si può quindi dire che la regolazione di un algoritmo sia equivalente ad un problema di apprendimento.[Bir09]

3.2 Configurazione automatica tramite iterated racing

L'approccio che è stato seguito per la regolazione dei parametri della politica Swarm, riportati nel paragrafo 4.1 a pagina 57, è definito *iterated racing* ed è esposto in [LIDLSB11], dove viene anche proposto il pacchetto *irace*, implementazione di iterated racing, utilizzato per il tuning della Swarm policy.

Il metodo è strutturato in tre passi:

1. campionamento di una nuova configurazione sulla base di particolari distribuzioni di probabilità;
2. selezione delle migliori configurazioni tra quelle appena campionate, mettendole in competizione;
3. aggiornamento della distribuzione di probabilità del campionamento, influenzandole favorendo configurazioni simili alle migliori trovate.

Le distribuzioni di probabilità citate sono quelle dei singoli parametri e possono essere normali o discrete, a seconda che il parametro, a cui la singola distribuzione è associata, sia numerico o categorico. L'aggiornamento di una distribuzione si traduce nella modifica di media e deviazione standard, nel caso di una distribuzione normale, o di valori di probabilità discreta nell'altro caso ed è un'operazione mirata a facilitare la scelta, in futuro, di valori appartenenti alle migliori configurazioni trovate fino a quel momento.

Una volta campionato un insieme finito di nuove configurazioni, queste vengono confrontate tra loro attraverso una competizione, costituita da più corse. Ad ogni passo della gara viene valutato il comportamento delle singole configurazioni candidate su una stessa istanza e, al termine di tutte le esecuzioni, quelle che hanno dato performance peggiori vengono scartate. Il processo prosegue con una nuova corsa, dove competono solo le candidate sopravvissute, fino al raggiungimento di uno tra i seguenti criteri:

- superamento di una soglia minima del numero di configurazioni superstiti;
- uso del numero massimo di istanze possibili;
- esaurimento del budget computazionale a disposizione.

Quest'ultimo può essere definito come un valore massimo di tempo computazionale oppure di esperimenti, dove per esperimento si intende l'esecuzione di una configurazione su di un'istanza.

3.2.1 Il modulo irace: descrizione dell'algoritmo

Un'implementazione proposta per il metodo di iterated racing è il modulo *irace* scritto nel linguaggio di programmazione R. Per eseguire, richiede in ingresso la definizione di quattro elementi: un set di istanze I rappresentativo di \mathcal{I} , uno spazio dei parametri (X), una funzione di costo (\mathcal{C}) e un budget computazionale (B). Definito l'input, bisogna fornire una stima del numero di iterazioni N^{iter} per cui vanno ripetute le competizioni tra le configurazioni. Questo valore viene definito sulla base del numero di parametri come

$$N^{iter} = \lfloor 2 + \log_2 N^{param} \rfloor \quad (3.3)$$

Per ogni iterazione viene stabilito il limite massimo di budget computazionale utilizzabile B_j come

$$B_j = \frac{B - B_{used}}{N^{iter} - j + 1} \quad (3.4)$$

dove j indica il numero dell'iterazione, definendo quindi B_j come crescente col procedere delle iterazioni.

Ogni competizione ha inizio partendo da un insieme di configurazioni candidate Θ_j , il cui numero è anch'esso variabile nel tempo e definito come

$$|\Theta_j| = N_j = \left\lfloor \frac{B_j}{\mu + \min(5, j)} \right\rfloor \quad (3.5)$$

quindi decrescente col procedere delle iterazioni. La motivazione dietro al comportamento di B_j e di $|\Theta_j|$ è diretta conseguenza dell'aggiornamento delle distribuzioni di probabilità dei parametri al termine di una competizione. Procedendo con le iterazioni, si avranno

configurazioni sempre più simili, per cui serviranno più valutazioni per identificare quelle migliori, condizione soddisfatta aumentando il budget computazionale e riducendo il numero di candidate. Quest'ultima operazione non inficia la bontà del sistema, dal momento che, procedendo con le iterazioni, la scelta delle configurazioni candidate è sempre più influenzata dai risultati ottimali trovati precedentemente. Il valore μ rappresenta il rapporto tra il budget computazionale e il numero di configurazioni desiderato dall'utilizzatore. In ultimo, al fine di evitare di restare con poche configurazioni da far competere tra loro, dopo la quinta iterazione il numero di candidate è mantenuto costante.

Il primo set di configurazioni è ottenuto da un campionamento uniforme dello spazio dei parametri X . Il confronto tra le candidate non avviene fin dalla prima gara, cioè dopo la valutazione delle configurazioni sulla prima istanza, ma dopo aver utilizzato almeno T^{first} istanze. I risultati forniti vengono quindi confrontati statisticamente. Le candidate che performano significativamente peggio rispetto ad almeno un'altra configurazione sono eliminate dalla competizione e solo quelle che sopravvivono vengono fatte competere su un'altra istanza. In seguito, il test viene ripetuto ogni T^{each} istanze, dove il valore predefinito è $T^{each} = 1$. La successione delle corse viene terminata al verificarsi di almeno uno dei seguenti criteri di terminazione:

- il budget computazionale residuo è insufficiente a testare tutte le configurazioni superstiti su una nuova istanza ($B_j < N_j^{surv}$);
- viene raggiunto un numero minimo di configurazioni accettabili per eseguire una nuova competizione ($N_j^{surv} \leq N^{min}$).

Le configurazioni rimaste al termine di una competizione ricevono un punteggio r_z in base al tipo di test statistico adottato. Vengono infine selezionate le $N^{elite} = \min(N_j^{surv}, N^{min})$ configurazioni d'élite Θ^{elite} , corrispondenti a quelle col punteggio più basso, per poi proseguire con una nuova iterazione.

Iterazioni successive useranno un numero N_j di configurazioni calcolato come indicato all'equazione 3.5. Di queste, $N_j^{new} = N_j - N_{j-1}^{elite}$ sono nuove candidate generate sulla base dei risultati precedenti. Viene selezionata una configurazione base θ_z , campionandola nel gruppo d'élite Θ^{elite} , con probabilità

$$p_z = \frac{N_{j-1}^{elite} - r_z - 1}{N_{j-1}^{elite}(N_{j-1}^{elite} + 1)/2} \quad (3.6)$$

proporzionale al punteggio r_z ottenuto. Di conseguenza, configurazioni con punteggio più alto hanno maggiori possibilità di essere selezionate come genitori. Per ognuno dei X_d , $d = 1, \dots, N^{param}$ parametri viene campionato il nuovo valore, secondo le distribuzioni associate ai corrispondenti parametri del genitore θ_z , e l'ordine di valutazione è dato dal grafo delle condizioni di dipendenza, ovvero prima si considerano quelli indipendenti e successivamente i subordinati, solo nel caso le condizioni a cui sono associati siano soddisfatte. Inoltre, per un parametro subordinato mancante nella configurazione padre, che si attiva in quella nuova, il valore viene selezionato attraverso un campionamento uniforme, in analogia alla prima iterazione.

Per un parametro numerico X_d , definito con range $[\underline{x}_d, \bar{x}_d]$, il nuovo valore è campionato dalla distribuzione normale troncata $\mathcal{N}(x_d^z, \sigma_d^j)$, in modo da mantenerlo all'interno del range stabilito. Il valor medio x_d^z è pari al valore del d -esimo parametro della configurazione d'élite θ_z mentre la varianza σ_d^j , inizialmente pari a $(\bar{x}_d - \underline{x}_d)/2$, decresce ad ogni iterazione come

$$\sigma_d^j = \sigma_d^{j-1} \left(\frac{1}{N_j^{new}} \right)^{1/N^{param}} \quad (3.7)$$

consentendo di campionare valori sempre più vicini a quello presente nella configurazione padre, di fatto definendo una ricerca locale attorno alle migliori trovate nel corso delle diverse iterazioni. Lo spazio di ricerca, ad ogni iterazione, viene ridotto di un fattore costante, il cui valore è legato al numero di nuove configurazioni N_j^{new} che vengono generate.

Per un parametro categorico X_d , con livelli $X_d \in \{x_1, x_2, \dots, x_{n_d}\}$, il nuovo valore è campionato dalla distribuzione di probabilità discreta $\mathcal{P}^{j,z}(X_d)$. Inizialmente $\mathcal{P}^{1,z}(X_d)$ è uniformemente distribuita nel dominio di X_d e viene successivamente aggiornata come

$$\mathcal{P}^{j,z}(X_d = x_j) = \mathcal{P}^{j-1,z}(X_d = x_j) \left(1 - \frac{j-1}{N^{iter}} \right) + \Delta\mathcal{P} \quad (3.8)$$

dove

$$\Delta\mathcal{P} = \begin{cases} \frac{j-1}{N^{iter}} & \text{se } x_j = x_z \\ 0 & \text{altrimenti} \end{cases} \quad (3.9)$$

Le nuove configurazioni generate, con le rispettive distribuzioni modificate, sono unite con le configurazioni d'élite per costituire il nuovo set di candidate da impiegare nella nuova competizione.

All'esaurimento del budget computazionale prestabilito ($B_{used} > B$) o se il numero di

nuovi candidati è inferiore a quello delle configurazioni d'élite ($N_j \leq N_{j-1}^{elite}$, ovvero non è possibile generare nuove configurazioni) l'algoritmo viene fermato. Il raggiungimento del numero prestabilito di iterazioni, diversamente, non costituisce motivo di terminazione dell'algoritmo. In questo caso, si prosegue con una nuova iterazione e l'algoritmo procede fino al raggiungimento di una delle due condizioni precedentemente indicate.[LIDLSB11]

Capitolo 4

Risultati sperimentali

L'elaborazione della politica Swarm e della funzione di stimolo, usata dalla prima per la selezione delle politiche di basso livello, ha portato alla definizione dei parametri riportati in sezione 4.1. Il loro tuning è stato eseguito per due scenari di diversa complessità, un incrocio semplice ed una ridotta porzione della rete stradale di Bologna. Per questo scopo è stato impiegato *irace*, la cui descrizione è riportata in 3.2.1.

Il criterio di ottimizzazione scelto è la minimizzazione del tempo di attesa medio dei veicoli. Per dimostrare l'efficacia del tuning e la necessità che questo processo sia automatizzato, si sono volute ricercare sia le migliori configurazioni dei parametri, capaci quindi di minimizzare il criterio scelto, sia quelle peggiori, ricavate minimizzando l'inverso del tempo medio di attesa. Il confronto tra le prestazioni ottenute con le configurazioni migliori contro le altre ha lo scopo di evidenziare il guadagno prestazionale che si può ottenere attraverso l'impiego del tuning automatico, mostrando il range delle possibili performance entro cui ci si muove al variare dei valori assunti dai parametri degli algoritmi.

4.1 Tuning dei parametri della Swarm policy

I parametri individuati per il funzionamento della politica Swarm possono essere suddivisi in due gruppi. Il primo è strettamente legato alla policy ed è riportato nella tabella 4.1, mentre il secondo set, riportato in tabella 4.2, è legato ai parametri della funzione di stimolo di ogni singola politica di basso livello, per cui ne esiste una copia per ognuna di queste.

Oggetto di studio è stato il tuning del gruppo in tabella 4.1. Per questo scopo è stato utilizzato *irace*, il cui funzionamento è descritto in 3.2.1. Per poter eseguire, è necessario

Tabella 4.1: Parametri propri della politica swarm

Parametro	Valore predefinito	Significato
PHERO_MAXVAL	10,0	Massimo valore del livello di feromone, uguale sia per le corsie d'ingresso sia per quelle d'uscita.
BETA_NO	0,99	Coefficiente di evaporazione del feromone per le corsie d'ingresso.
GAMMA_NO	1,0	Coefficiente di accumulo del feromone per le corsie d'ingresso, da moltiplicare per il numero di veicoli.
BETA_SP	0,99	Coefficiente di evaporazione del feromone per le corsie d'uscita.
GAMMA_SP	1,0	Coefficiente di accumulo del feromone per le corsie d'uscita, da moltiplicare per la differenza tra la velocità massima consentita e quella media dei veicoli.
P_CHANGE	0,1	Probabilità di selezionare una nuova politica ogni volta che l'algoritmo di swarm viene invocato.
THETA_MAX	0,8	Valore massimo per le soglie delle funzioni di stimolo.
THETA_MIN	0,2	Valore minimo per le soglie delle funzioni di stimolo.
THETA_INIT	0,5	Valore iniziale per le soglie delle funzioni di stimolo.
LEARNING_COX	0,0005	Coefficiente d'apprendimento per l'aggiornamento delle soglie delle funzioni di stimolo.
FORGETTING_COX	0,0005	Coefficiente d'oblio per l'aggiornamento delle soglie delle funzioni di stimolo.

[HFB⁺13]

Tabella 4.2: Parametri della funzione di stimolo delle politiche di basso livello.

Parametro	Significato
STIM_COX_EXP	Fattore globale di scala.
STIM_OFFSET_IN	Offset relativo al livello di feromone per le corsie d'ingresso. Muove la gaussiana lungo l'asse f_{in} .
STIM_OFFSET_OUT	Offset relativo al livello di feromone per le corsie d'uscita. Muove la gaussiana lungo l'asse f_{out} .
STIM_COX_EXP_IN	Vale 0 se il livello di feromone nelle corsie d'ingresso non è considerato nella funzione di stimolo, 1 altrimenti.
STIM_COX_EXP_OUT	Vale 0 se il livello di feromone nelle corsie d'uscita non è considerato nella funzione di stimolo, 1 altrimenti.
STIM_DIVISOR_IN	Influenza la larghezza della gaussiana lungo l'asse f_{in} .
STIM_DIVISOR_OUT	Influenza la larghezza della gaussiana lungo l'asse f_{out} .

[HFB⁺13]**Figura 4.1:** Definizione di un parametro per irace

```
<nome> <etichetta> <tipo> <valori> [ | <condizione> ]
```

descrivere lo spazio dei parametri e fornire le istanze con cui effettuare la regolazione, la configurazione del tool e l'*hook-run*, una funzione o programma esterno che si occupi dell'applicazione della configurazione generata sull'istanza selezionata e ritorni il costo corrispondente.

4.1.1 Descrizione dello spazio dei parametri

Lo spazio dei parametri viene descritto attraverso una tabella, le cui righe, composte come in figura 4.1, hanno il seguente significato:

nome il nome del parametro;

Figura 4.2: Definizione dello spazio dei parametri regolati

# name	switch	type	values	[conds]
threshold	""	i	(5, 15)	
min_dec_phase_d	""	i	(3000, 7000)	
change_plan_p	""	r	(0.01000, 0.90000)	
gamma_sp	""	r	(0.10000, 1.00000)	
beta_sp	""	r	(0.10000, 0.99999)	
gamma_no	""	r	(0.10000, 1.00000)	
beta_no	""	r	(0.10000, 0.99999)	
theta_max	""	r	(0.50000, 1.00000)	
theta_min	""	r	(0.05000, 0.50000)	
learning_cox	""	r	(0.00010, 0.10000)	
forgetting_cox	""	r	(0.00010, 0.10000)	

etichetta l’etichetta associata al parametro. Corrisponde alla stringa passata, assieme al valore campionato per il parametro, all’hook-run e corrisponde ad un’opzione dell’invocazione a riga di comando di quella funzione o programma;

tipo il tipo del parametro, il cui valore può essere *integer*, *real*, *ordinal* o *categorical*¹ e indicati tramite la loro iniziale;

valori l’intervallo, per i parametri numerici, o l’insieme, per quelli categorici, di valori possibili;

condizione la condizione, opzionale, serve a indicare se un parametro è abilitato o no, rendendolo quindi subordinato. Il valore per questo parametro viene campionato solo se la condizione è vera, altrimenti né l’etichetta né il valore vengono passati come argomenti all’hook-run. La condizione, espressa con la sintassi di R, non deve innescare dipendenze cicliche tra parametri, altrimenti l’esecuzione viene terminata con errore.[LIDLSB11]

La descrizione dello spazio dei parametri utilizzata per il tuning della politica Swarm è riportata in figura 4.2.

¹Integer e real, rispettivamente intero e reale, sono i due tipi di parametri numerici possibili. Ordinal e categorical sono i due possibili parametri categorici, dove ordinal indica quei parametri non numerici per cui esiste una relazione d’ordine, ad esempio “primo, secondo, terzo”.

Figura 4.3: Opzioni definite per il tuning della Swarm policy

```
parameterFile <- "./parameters-sumo.txt"  
instanceDir <- "../Instances/Instances_4_1000_1500_10"  
maxExperiments <- 20000  
digits <- 5  
sampleInstances <- 1  
parallel <- 117
```

Rispetto ai parametri riportati in tabella 4.1 a pagina 58, vi sono alcune differenze. Il primo indicato, `threshold`, è il valore della soglia secondaria legata alla decisione della durata delle fasi decisional². Il parametro `min_dec_phase_d` determina invece la soglia di durata minima delle fasi decisional, indipendentemente dalla durata minima che queste devono avere, ed è espressa in millisecondi. Viene usata, per esempio, nella politica Request³. Non viene preso in considerazione `THETA_INIT`, per via delle limitazioni di irace riguardanti il tipo di condizioni di dipendenza esprimibili tra i parametri, ed il suo valore è prefissato pari a 0,5. Motivazioni analoghe guidano la scelta dei range per i parametri rappresentanti `THETA_MAX` e `THETA_MIN`.

4.1.2 Configurazione di irace

La configurazione di irace avviene attraverso la definizione di diverse opzioni. Queste possono essere impostate singolarmente, usando l'ambiente interattivo fornito da R, oppure attraverso l'uso di un file denominato `tune-conf`.^[LIDLSB11]

Le opzioni che sono state sfruttate sono riportate in figura 4.3. Significano:

parameterFile indica ad irace dove reperire la descrizione dello spazio dei parametri;

instanceDir indica ad irace dove reperire la descrizione delle istanze con cui valutare le configurazioni campionate;

maxExperiments è il budget computazionale a disposizione per la regolazione dei parametri;

²v. pagina 43.

³v. 2.3.2.2 a pagina 43.

digits è la precisione richiesta per quanto riguarda i parametri a valori reali;

sampleInstances se impostato, indica ad irace di campionare casualmente le istanze tra quelle disponibili, invece di sceglierle con un ordine predefinito sempre identico;

parallel indica ad irace se deve avviare multiple esecuzioni parallele di hook-run, valutando quindi più configurazioni diverse sulla stessa istanza contemporaneamente.

4.1.3 Lo script di valutazione hook-run

La valutazione delle configurazioni candidate è effettuata mediante l'esecuzione di un oggetto computazionale denominato *hook-run*. Questo può essere una funzione definita in R, il linguaggio con cui è scritto irace, o un programma esterno, caso d'interesse per il tuning della politica Swarm.

All'atto della sua invocazione, hook-run riceve, nel seguente ordine, l'istanza con cui effettuare la valutazione, un identificativo numerico della configurazione candidata univoco nell'ambito della competizione in corso e la configurazione da valutare, sotto forma di coppie etichetta-valore. In risposta deve essere fornito un valore numerico che corrisponda alla misura di costo della configurazione candidata per la particolare istanza fornita[LIDLSB11]. Ogni altro tipo di risposta non numerica viene valutata come errore dipendente da hook-run e risulta nella terminazione di irace, senza che venga fornita una soluzione anche parziale.

L'hook-run prodotto per la regolazione della politica Swarm, riportato integralmente in Appendice A, può essere suddiviso logicamente in quattro porzioni, come indicato in pseudocodice nell'algoritmo 1 nella pagina successiva, e la misura di costo che fornisce è il tempo di attesa medio dei veicoli all'interno della simulazione. Quest'ultima è effettuata dal software SUMO, simulatore di traffico sviluppato dal DLR⁴ e utilizzato all'interno del progetto COLOMBO. Nella sezione 4.2.1 a pagina 65 è fornita la descrizione dell'architettura computazionale utilizzata per questo scopo, per cui la motivazione di alcune scelte implementative è omessa e la spiegazione del primo blocco è rimandata.

⁴Deutsches Zentrum für Luft- und Raumfahrt e.V., letteralmente "Il centro tedesco per l'aviazione e i voli spaziali" ma spesso riportato in letteratura come German Aerospace Center.

Algoritmo 1 Pseudocodice di hook-run

```
hook-run(istanza, id_candidata, parametri){  
    <selezione del client a cui far eseguire la  
        simulazione>  
    <preparazione del file descrivente la logica del  
        semaforo>  
    <esecuzione della simulazione>  
    <elaborazione del risultato della simulazione e  
        rimozione dei file generati>  
}
```

4.1.3.1 Preparazione del file descrivente la logica del semaforo

Il software SUMO necessita di tre file descrittivi per poter eseguire una simulazione:

net-file contiene la struttura della rete stradale sotto esame, compresi i semafori installati e la loro logica;

route-file contiene la descrizione dei percorsi dei veicoli che devono essere simulati e, nel nostro caso di studio, corrisponde all'istanza con cui valutare i parametri di Swarm;

additional-file se usato, consente di modificare le logiche semaforiche dei controllori installati nella rete sotto esame.

Mentre la rete è fissa e i flussi veicolari vengono definiti a priori, costituendo l'insieme delle possibili istanze, il terzo file deve essere generato sul momento, dato che contiene la configurazione campionata da irace per essere valutata.

4.1.3.2 Esecuzione della simulazione

Ottenuto il file *additional*, è possibile eseguire il simulatore sul client precedentemente scelto per questo scopo. Al software SUMO viene specificato di produrre un particolare file, denominato *tripinfo*, contenente un riassunto dell'attività di ogni veicolo simulato. Il dato più interessante è denominato *waitSteps* e indica il numero di passi nella simulazione⁵ in cui il veicolo è rimasto fermo in attesa.

⁵Nelle nostre simulazioni, un passo equivale a un secondo.

Figura 4.4: Esempio di informazioni per un veicolo riportati nel file *tripinfo*

```
<tripinfo id="1" depart="134.00" departLane="3i_0" departPos="6.60" departSpeed="13.89" departDelay="0.00" arrival="281.00" arrivalLane="7o_0" arrivalPos="48.50" arrivalSpeed="13.89" duration="147.00" routeLength="2029.40" waitSteps="3" rerouteNo="0" devices="tripinfo_0hbefa_0" vType="type1" vaporized="">
  <emissions CO_abs="0" CO2_abs="316222.687381" HC_abs="49.478428" PMx_abs="20.338856" NOx_abs="531.005586" fuel_abs="126.073344"/>
</tripinfo>
```

4.1.3.3 Elaborazione del risultato

Il file *tripinfo* viene esaminato, ricercando i valori di *waitSteps* di ogni veicolo simulato, per calcolare il costo della configurazione esaminata sull'istanza selezionata, ovvero il tempo d'attesa medio dei veicoli. Questo valore viene quindi fornito come risultato ad irace.

4.2 Test eseguiti

Si è voluta testare l'efficacia del tuning dei parametri su due scenari di diversa complessità. In entrambi i casi, i parametri regolati erano condivisi tra tutti i semafori presenti nella rete stradale sotto esame.

Con riferimento all'equazione 1.9 a pagina 12, è stato calcolato un valore del flusso di saturazione pari a $2000 Ae/h$ e, da questo, sono state determinate empiricamente cinque diverse fasce di flussi di traffico. Il tuning problem della politica Swarm è stato affrontato per ognuno di questi. Le ripartizioni, ed i rispettivi range, sono:

Fascia 1 (F1) con intervallo $[0, 300] Ae/h$ e corrispondente ad un traffico leggero;

Fascia 2 (F2) con intervallo $[300, 600] Ae/h$ e corrispondente ad un traffico medio-basso;

Fascia 3 (F3) con intervallo $[600, 1000] Ae/h$ e corrispondente ad un traffico medio-elevato;

Fascia 4 (F4) con intervallo $[1000, 1500] Ae/h$ e corrispondente ad un traffico elevato;

Fascia 5 (F5) con intervallo $[1500, 2000]$ Ae/h e corrispondente ad una congestione della rete stradale.

Lo scopo è di valutare l'efficacia del parameter tuning, individuando l'ampiezza dello spazio di ricerca dei parametri. Per ognuna di queste fasce, oltre alla ricerca delle configurazioni *migliori*, è stata condotta anche la ricerca di quelle *peggiori*, invertendo la funzione di costo fornita ad irace. Sono poi stati eseguiti test paralleli con i due set di valori trovati, per ognuna delle cinque fasce, su un insieme di trenta istanze. Il divario tra le prestazioni delle due configurazioni è direttamente collegato all'ampiezza dello spazio di ricerca dei parametri, per cui al crescere del primo aumenta anche il secondo. Di conseguenza, uno spazio di ricerca ampio implica una maggior facilità di scelta di un insieme di valori non ottimale dei parametri, se non si ricorre al tuning automatico.

Le modalità di generazione delle istanze di test sono le stesse impiegate per la produzione di quelle usate per la regolazione. Il criterio di valutazione usato per i test è il medesimo impiegato per la funzione di costo del tuning, ovvero il tempo medio d'attesa dei veicoli.

4.2.1 Descrizione dell'architettura computazionale utilizzata

Per l'esecuzione del tuning con irace è stata realizzata un'architettura client-server, comunicanti via SSH⁶. Sul server sono presenti irace, i programmi accessori indirettamente utilizzati da questo attraverso lo script hook-run, i file di descrizione degli scenari e le istanze utilizzate per il tuning, ovvero i file che descrivono i percorsi dei veicoli nello scenario. Il compito del server è di gestire la parte di generazione e confronto delle configurazioni dei parametri generate, attraverso il software irace, mentre la simulazione vera e propria, eseguita dal software SUMO, avviene sui singoli client. Ulteriore compito del server è il dispaccio delle simulazioni da eseguire, richieste da irace, al client di competenza. Avendo a disposizione un elevato numero di macchine, si è scelto di avere al più una simulazione in corso per singolo client, dal momento che già le istanze rappresentanti condizioni di traffico medio comportavano un uso del 100% della CPU da parte di SUMO.

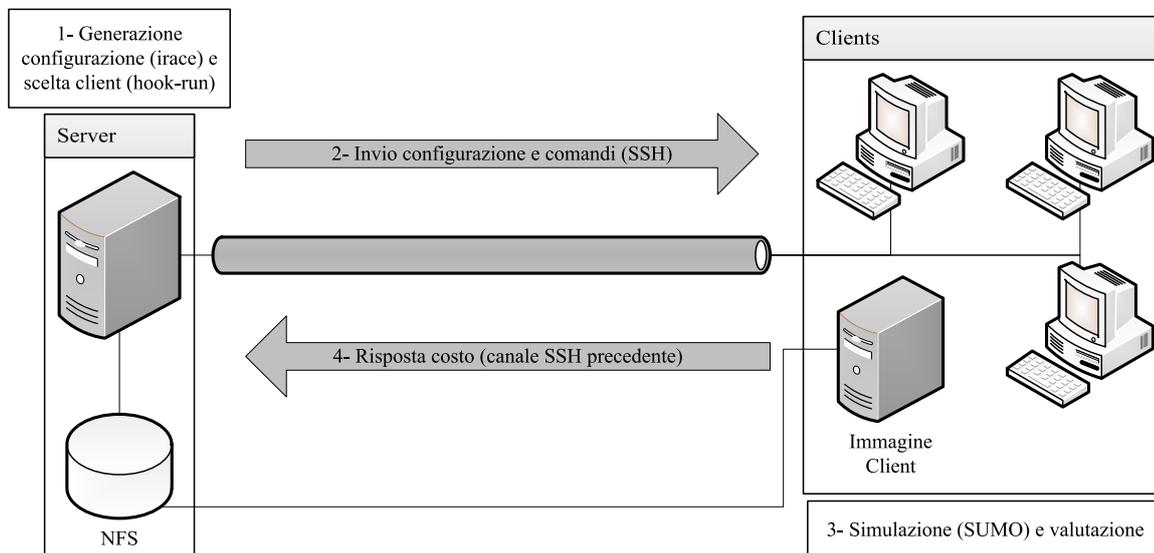
Lato client, l'unico applicativo presente localmente è il simulatore SUMO, mentre ogni altro file necessario⁷ viene caricato dal server attraverso un file system condiviso NFS⁸.

⁶Secure SHell

⁷I file di descrizione degli scenari, delle istanze e i programmi accessori per la generazione del file additional e per la valutazione del tripinfo.

⁸Network File System

Figura 4.5: Schema esemplificativo dell'architettura



Quest'implementazione è stata resa necessaria per via della natura dei client: questi, infatti, caricano tutti un'unica immagine di un sistema operativo Linux, accessibile in sola lettura per evitare ovvi problemi di concorrenza durante la scrittura. Salvo per le modifiche sul simulatore, il cui aggiornamento richiede un intervento diretto sul client e, quindi, un'interruzione di servizio, quest'approccio consente di essere indipendenti e di caricare nuovi scenari al momento del bisogno, senza la necessità di riconfigurare i client.

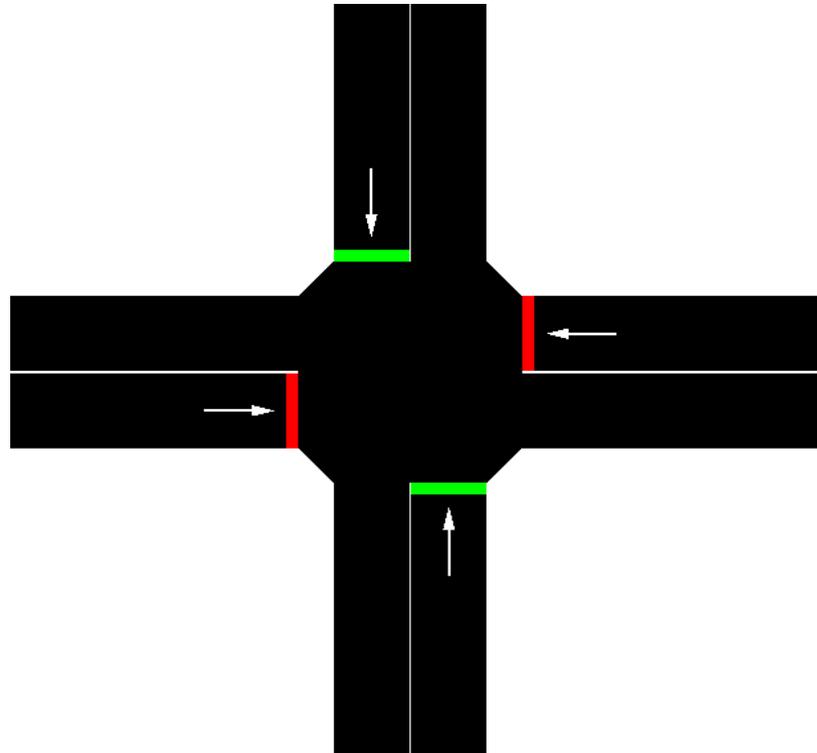
4.2.1.1 Informazioni tecniche sul server

Il server è una macchina virtuale ospitata su hypervisor KVM⁹, la macchina ospite fisica è equipaggiata con due processori Intel[®] Xeon[®] X5570 con frequenza di clock 2,93 GHz e 48 GB di memoria RAM. Lo storage è remoto e la connessione ad esso avviene attraverso quattro canali da 1 GB.

4.2.1.2 Informazioni tecniche sui client

Le macchine client, nominalmente 120, sono dotate di un processore Intel[®] Core[™] i5-2400 con frequenza di clock 3,1 GHz e 4 GB di memoria RAM. Un grosso vantaggio è

⁹Kernel-based Virtual Machine

Figura 4.6: Primo scenario: un singolo incrocio semplice

rappresentato dalla mancanza di disco fisso, fattore che rende la macchina meno prona a problemi in caso di mancanza di corrente.

Il collegamento col centro di calcolo, in cui si trova il server, passa attraverso uno switch, che convoglia tutti i client con canali a 100 MB, e due canali da 1 GB. La presenza di connessioni ridondate, sia in questo caso sia in quello del server, fornisce resistenza ai malfunzionamenti e moltiplicazione della capacità di trasporto dei dati.

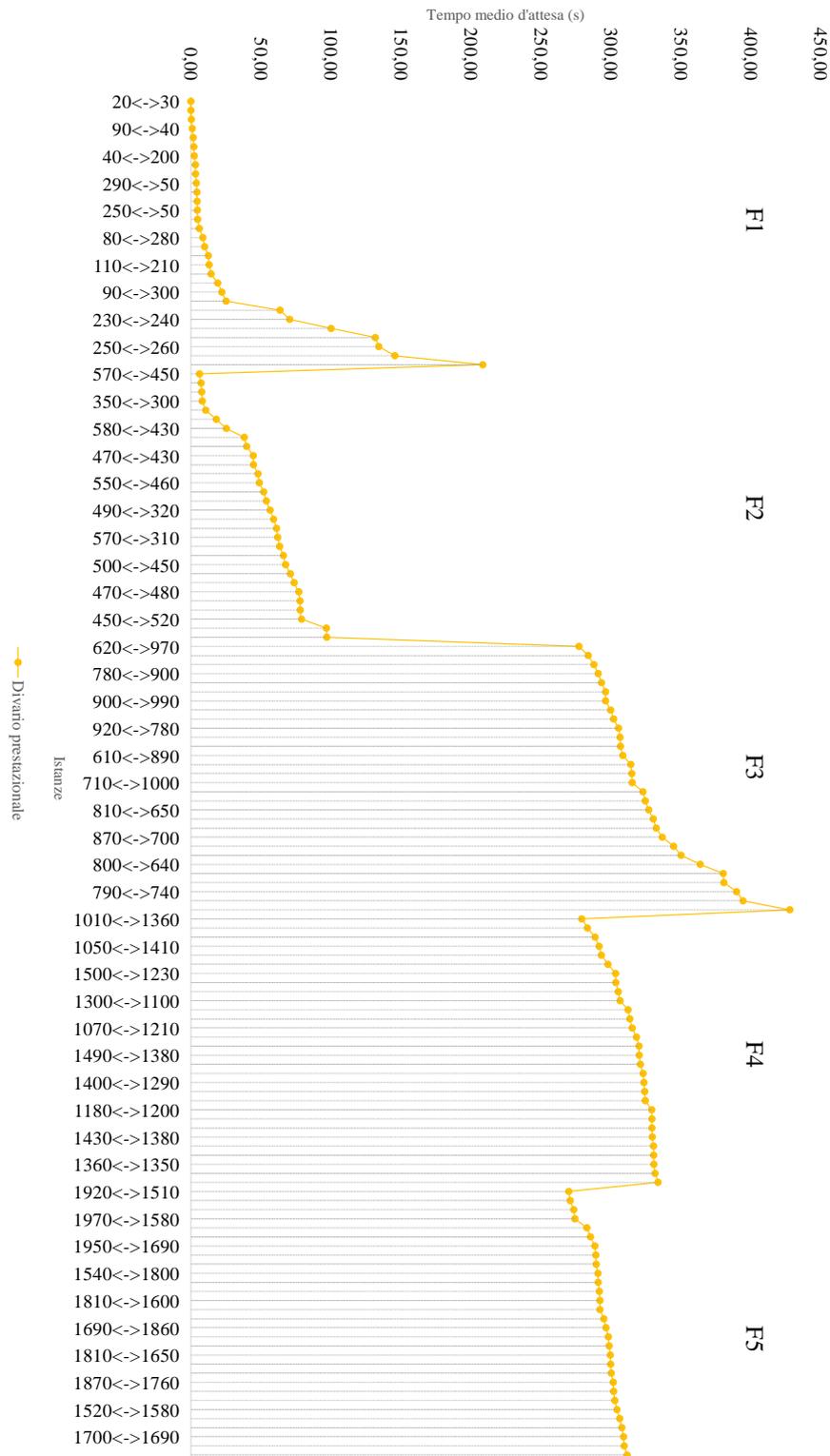
4.2.2 Regolazione di un incrocio singolo

Il primo scenario utilizzato è una semplice rete formata da un solo incrocio tra da due strade a singola corsia per senso di marcia, come illustrato in figura 4.6. In questo scenario, i veicoli possono transitare solo in direzione nord-sud, est-ovest e viceversa. La scelta è stata motivata dalla semplicità di modellazione che questa condizione rappresentava.

Le istanze costituenti l'insieme I^{10} vengono definite a partire da tutte le possibili coppie

¹⁰v. 3.2.1 a pagina 52.

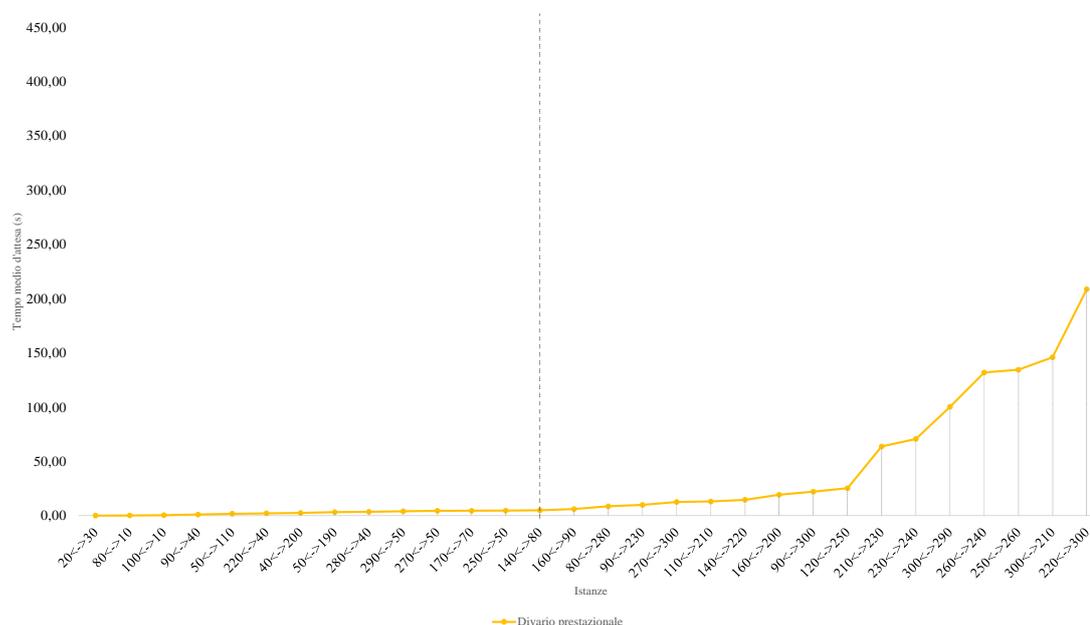
Figura 4.7: Confronto globale tra le configurazioni migliori e peggiori (incrocio singolo)



di due valori tra il minimo ed il massimo dell'intervallo di traffico relativo, con passo pari a 10. Il significato corrisponde al numero di veicoli in un'ora, per direzione, sulle direttrici est-ovest e nord-sud, per cui coppie con uguali valori ma diverso ordine sono considerate come diverse. L'istante di ingresso nella simulazione dei veicoli, definito in secondi, è stabilito casualmente sulla base della frequenza desiderata e l'inserimento avviene fino a due ore di tempo simulato.

In figura 4.7 è riportato il *gap*¹¹ tra le due configurazioni (calcolato come la distanza assoluta tra le loro soluzioni), per ognuna delle cinque fasce di traffico¹². Sull'asse delle ordinate sono riportate le istanze utilizzate mentre sulle ascisse è indicato il tempo medio di attesa dei veicoli, espresso in secondi. Globalmente si può notare la netta differenza che le divide, soprattutto nelle condizioni di traffico più elevato. Le discontinuità nella crescita della curva descritta corrispondono alla transizione da una fascia di traffico alla successiva.

Figura 4.8: Confronto tra la configurazione migliore e peggiore per la F1 (incrocio singolo)



Focalizzandosi sulle condizioni di traffico della fascia F1, il cui grafico è riportato in figura 4.8, si può notare come, in assenza di flussi veicolari consistenti su almeno una delle

¹¹ chiamato anche *divario prestazionale*.

¹² I valori corrispondenti a tutte le istanze di traffico sono riportate in appendice B.

Tabella 4.3: Medie e mediane delle performance delle configurazioni migliori e peggiori (incrocio singolo)

Fascia	Migliore (s)		Peggior (s)		Divario (s)	
	Media	Mediana	Media	Mediana	Media	Mediana
F1	4,29	3,45	38,32	10,73	34,05	7,26
F2	27,52	22,39	78,86	81,49	51,33	55,24
F3	102,85	97,07	429,7	431,65	315,85	315,06
F4	140	142,07	454,8	461,25	314,80	320,19
F5	141,57	142,59	435,76	438,754	294,19	295,74

due direttrici, le prestazioni delle due configurazioni siano sostanzialmente identiche. Per tanto, si può dedurre che il tuning per condizioni di traffico pressochè inesistente non apporta miglioramenti. All'aumentare del flusso di traffico, si può notare un incremento nella divergenza delle prestazioni (come indicato dall'aumento della pendenza della curva), e quindi un ampliamento dello spazio di ricerca dei parametri. Questo andamento prosegue e diventa più marcato nella fascia F3, riportata in figura 4.9, dove raggiunge il suo massimo. Nelle due fasce ad alto traffico (F4 e F5), si ha conferma di questa divergenza, come evidenziato in figura 4.10. Il *gap* evidenzia l'estensione dello spazio di ricerca dei parametri, indicando quindi le potenzialità di miglioramento introdotte dal processo di regolazione automatica dei parametri.

In tabella 4.3 sono riportati i valori di medie e mediane delle prestazioni delle configurazioni *migliore* e *peggiore*, assieme a quelli del divario tra queste.

Al di fuori del problema di tuning, si possono condurre ulteriori riflessioni. La prima è legata alle fasce di traffico, nello specifico la fascia F3 (figura 4.9). Osservando il tratto iniziale e finale del divario prestazionale, si nota una certa irregolarità rispetto al trend più lineare delle altre fasce, che può far pensare ad una riconsiderazione di queste suddivisioni (per esempio separando in due fasce diverse la fascia F3).

Una seconda riflessione può essere fatta su alcuni dei parametri, osservando i valori tuned, riportati in tabella 4.4, e i loro andamenti:

THRESHOLD il suo valore tende a diminuire all'incrementare dei flussi di traffico, ad indicare la necessità di terminare più velocemente le fasi decisional così da servire più velocemente diverse correnti di traffico;

P_CHANGE il valore regolato è notevolmente diverso da quello deciso in [HFB⁺13], ad

Figura 4.9: Confronto tra la configurazione migliore e peggiore per la F3 (incrocio singolo)

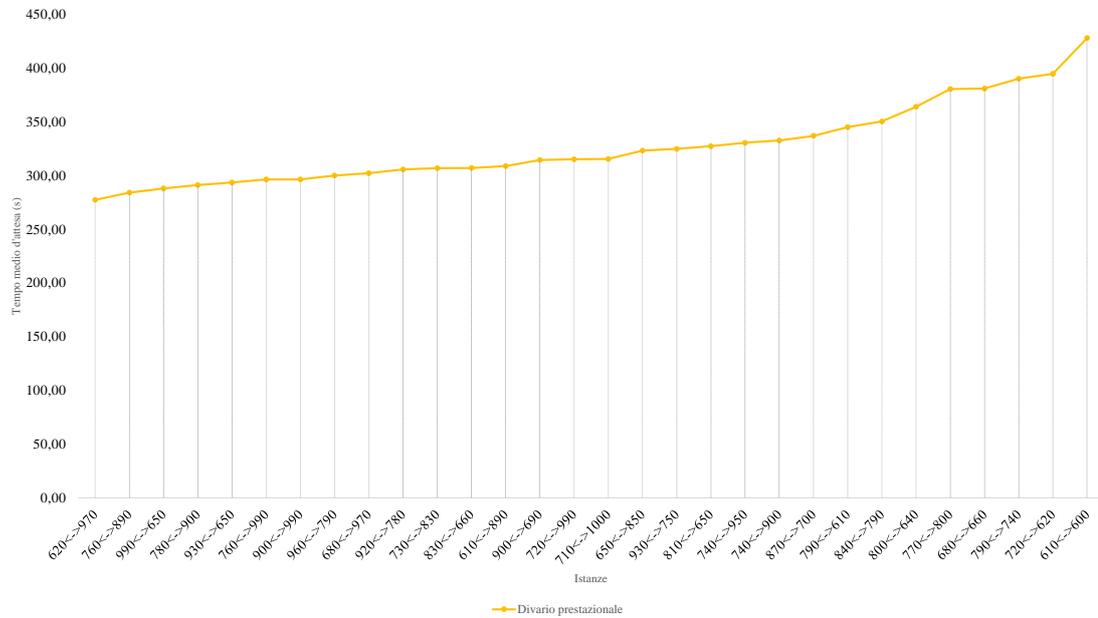


Figura 4.10: Confronto tra la configurazione migliore e peggiore per la F5 (incrocio singolo)

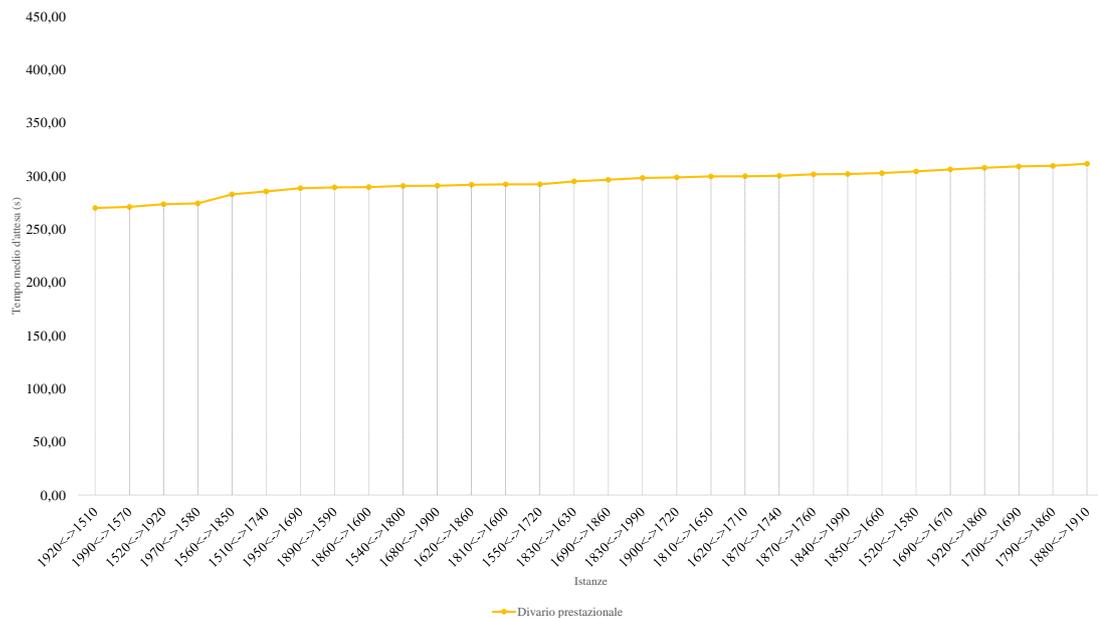


Tabella 4.4: Valori delle configurazioni d'élite migliori (incrocio singolo)

Parametro	Range	F1	F2	F3	F4	F5
THRESHOLD	[5 - 15]	14	13	14	11	8
MIN_DEC_PHASE_D	[3000 - 7000]	6937	3325	4649	5788	4081
P_CHANGE	[0,01 - 0,9]	0,7338	0,8635	0,5410	0,8780	0,8684
GAMMA_SP	[0,1 - 1,0]	0,1083	0,1050	0,3171	0,3386	0,3103
BETA_SP	[0,1 - 0,99]	0,9520	0,8641	0,2160	0,1348	0,1956
GAMMA_NO	[0,1 - 1,0]	0,2693	0,4148	0,1133	0,4715	0,4940
BETA_NO	[0,1 - 0,99]	0,4023	0,3808	0,6607	0,2907	0,2281
THETA_MAX	[0,5 - 1,0]	0,5222	0,5183	0,5069	0,5022	0,5107
THETA_MIN	[0,05 - 0,5]	0,4278	0,3301	0,0529	0,4080	0,4015
LEARNING_COX	[0,0001 - 0,1]	0,0827	0,0086	0,0918	0,0423	0,0678
FORGETTING_COX	[0,0001 - 0,1]	0,0573	0,0493	0,0502	0,0031	0,0668

Tabella 4.5: Valori delle configurazioni d'élite peggiori (incrocio singolo)

Parametro	Range	F1	F2	F3	F4	F5
THRESHOLD	[5 - 15]	7	12	13	14	12
MIN_DEC_PHASE_D	[3000 - 7000]	6297	3897	3928	5932	5009
P_CHANGE	[0,01 - 0,9]	0,0141	0,0108	0,0101	0,01	0,011
GAMMA_SP	[0,1 - 1,0]	0,1224	0,281	0,1876	0,2796	0,3344
BETA_SP	[0,1 - 0,99]	0,2612	0,2406	0,4713	0,1239	0,1298
GAMMA_NO	[0,1 - 1,0]	0,1188	0,7049	0,876	0,7662	0,1001
BETA_NO	[0,1 - 0,99]	0,1221	0,7269	0,7582	0,8861	0,1318
THETA_MAX	[0,5 - 1,0]	0,9148	0,9945	0,9969	0,9181	0,8489
THETA_MIN	[0,05 - 0,5]	0,0622	0,1683	0,1403	0,3784	0,2844
LEARNING_COX	[0,0001 - 0,1]	0,0969	0,0079	0,0598	0,0438	0,043
FORGETTING_COX	[0,0001 - 0,1]	0,0017	0,0534	0,0312	0,0315	0,0483

indicare la necessità di una probabilità più elevata per la sostituzione della politica di basso livello scelta da Swarm, che può tradursi in una maggior reattività;

GAMMA/BETA_SP si individuano due andamenti legati alla gestione del feromone in uscita. Da un lato, il fattore d'accumulo è molto ridotto e tende ad essere lievemente aumentato all'incremento dei flussi di traffico, dall'altro vi è la tendenza ad un'evaporazione estremamente ridotta in condizioni di traffico lieve¹³, situazione che si ribalta

¹³Il coefficiente di evaporazione viene moltiplicato per il feromone attuale, quindi un alto valore di questo coefficiente implica una bassa evaporazione.

in presenza di congestione. Il comportamento emergente si può interpretare con la necessità di passare dal solo interessamento dell'avvenuta uscita di veicoli quando non vi è traffico, dato che si ha un ridotto accumulo di feromone ma una sua elevata persistenza, al riconoscimento dell'avvenuto sgombero dell'incrocio, data l'elevata evaporazione del feromone;

THETA MAX/MIN escludendo il caso della F3, si nota come i due valori tendano ad avere uno scarto ridotto, suggerendo una limitata libertà d'azione del meccanismo di rinforzo per la selezione delle politiche, fattore che può anche essere legato alla struttura dello scenario in esame. Da notare che THETA_MAX assuma sempre valori vicini al minimo del suo range, ad indicare la possibilità di ridefinirlo¹⁴.

4.2.3 Regolazione di una rete complessa

Il secondo scenario, denominato *Pasubio* e riportato in figura 4.11, nella pagina seguente, è estratto dalla rete stradale di Bologna. L'area considerata è delimitata a sud da uno dei tratti stradali più prони alla congestione della città, Viale Gandhi, e a nord da Via Saffi, altro tratto viario molto trafficato. Buona parte delle strade secondarie è stata riprodotta, tra cui via Pasubio, che da il nome allo scenario. Le cinque intersezioni evidenziate in figura 4.11, nel contesto della simulazione, sono gestite da impianti semaforici auto-organizzanti che fanno uso della Swarm policy.

Per questo scenario, le istanze costituenti l'insieme I^{15} sono state generate avvalendosi di due tool accessori del simulatore SUMO. Il primo consente di inserire, con un intervallo temporale a piacere, veicoli con punto di partenza ed arrivo scelti casualmente tra tutti i nodi della rete stradale, con la possibilità di dar maggior peso a strade con più corsie o a vertici posti sulla frontiera della rete. Per mantenere un'analogia con lo scenario descritto in 4.2.2, si è scelto di generare auto che entrino ed escano dalla simulazione dai nodi esterni della rete, oltre a dar maggior peso alle strade a più corsie. Il secondo tool ha il compito di costruire il percorso che ogni veicolo, generato precedentemente, dovrà seguire. Per questo scenario, la congestione della rete è stata calcolata pari al prodotto tra il numero di ingressi della rete e il valore di flusso di congestione per una singola corsia. Essendo

¹⁴Si ricorda che la scelta è stata dettata per necessità dalle limitazioni imposte sui vincoli tra parametri da irace, v. 4.1.1 a pagina 59.

¹⁵v. 3.2.1 a pagina 52.

Figura 4.11: Secondo scenario: una piccola porzione della rete stradale di Bologna



nove i nodi di ingresso presenti sulla frontiera, il valore del flusso di saturazione è pari a $9 * 2000 Ae/h = 18000 Ae/h$. Di conseguenza, i valori degli intervalli di flusso veicolare delle cinque fasi definite precedentemente continuano a valere, solo moltiplicati per 9. Sono quindi state generate tutte le istanze tra il valore minimo ed il valore massimo di flusso veicolare con passo $2 * 9$, diversamente dal passo 10 dello scenario semplice.

Data la maggior complessità della rete e il numero globalmente più elevato di veicoli presenti, al fine di ottenere tempi di simulazione in linea con quelli del primo scenario, il loro inserimento è distribuito sono nella prima ora di simulazione.

In figura 4.12 è riportato il gap tra le due configurazioni, per ognuna delle cinque fasce di traffico¹⁶. Osservando nello specifico le fasi F1 e F2, si nota un comportamento analogo a quanto descritto in 4.2.2. In caso di scarso traffico ancora una volta la configurazione dei parametri è poco influente, mentre all'aumentare dell'intensità dei flussi veicolari aumenta di conseguenza la distanza tra le prestazioni delle due configurazioni, come evidenziato in fascia F2 (figura 4.13), andando a marcare l'importanza del parameter tuning.

¹⁶I valori corrispondenti a tutte le istanze di traffico sono riportate in appendice B.

Figura 4.12: Confronto globale tra le configurazioni migliori e peggiori (Pasubio)

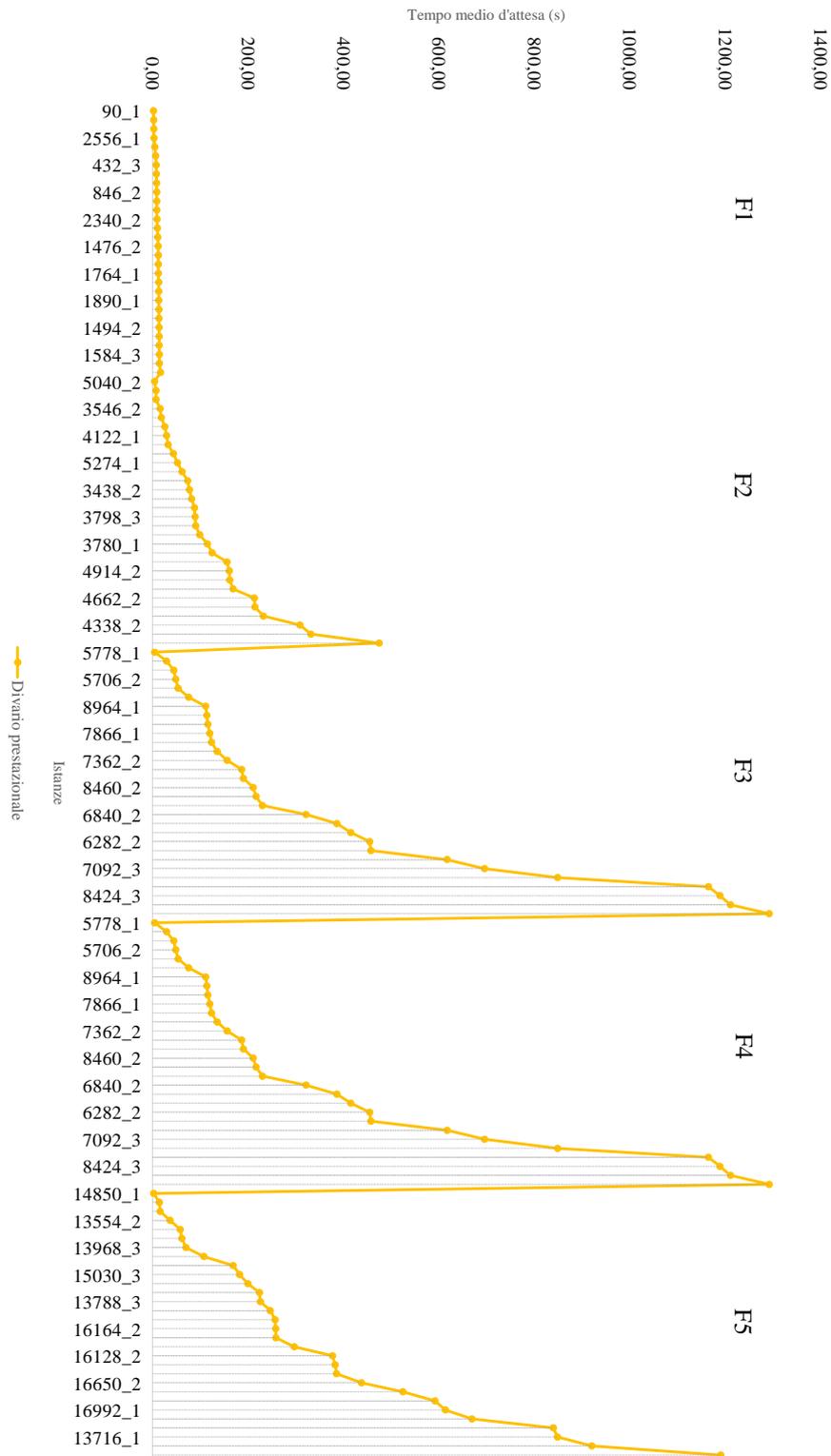


Figura 4.13: Confronto tra la configurazione migliore e peggiore per la F2 (Pasubio)

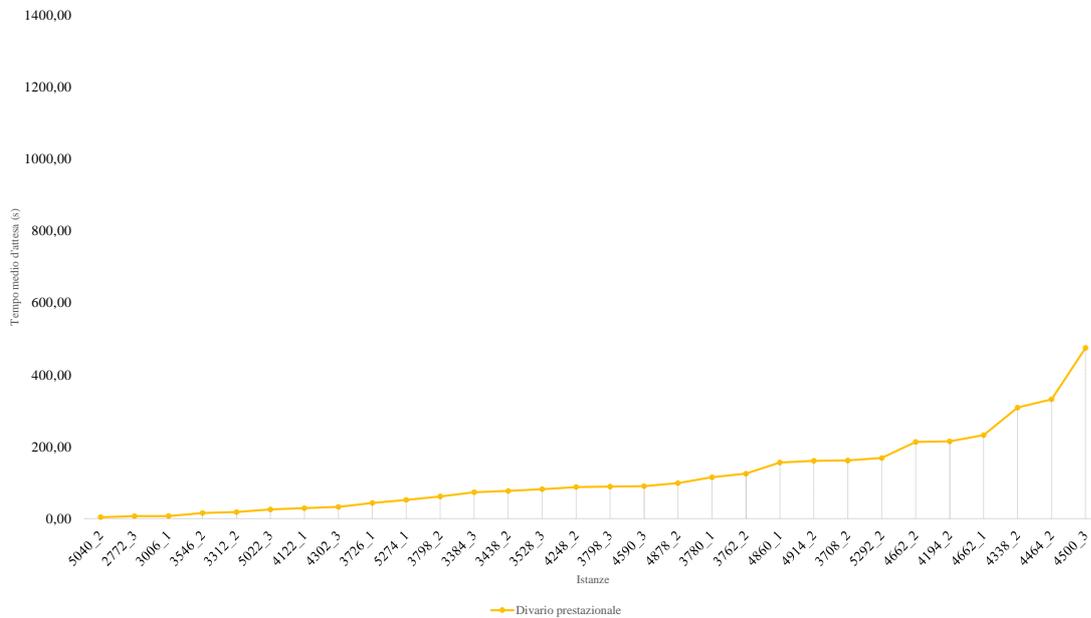


Figura 4.14: Confronto tra la configurazione migliore e peggiore per la F3 (Pasubio)

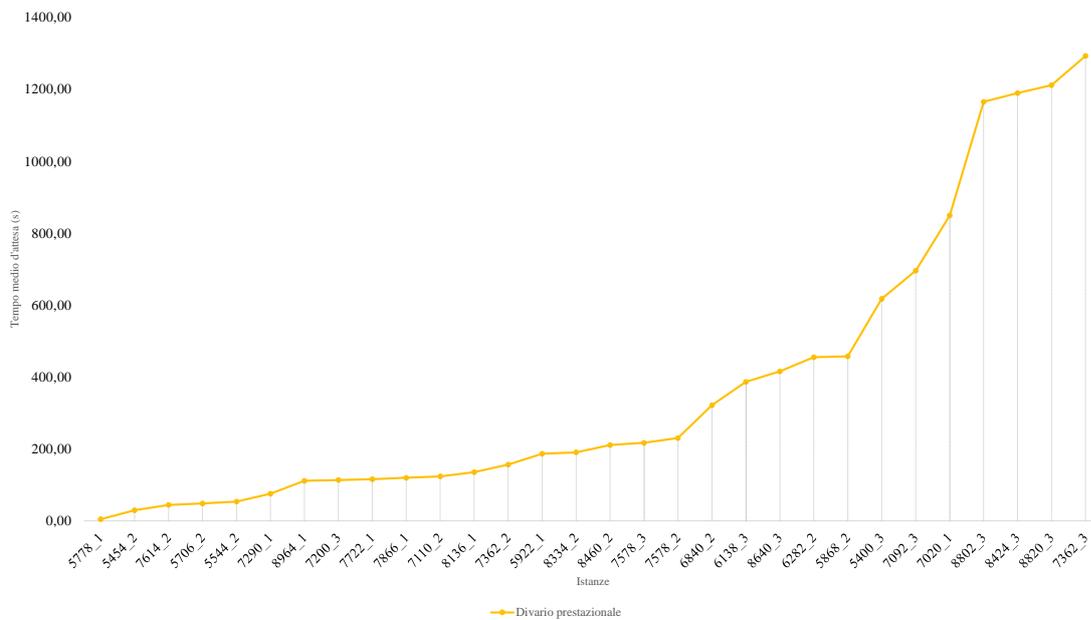
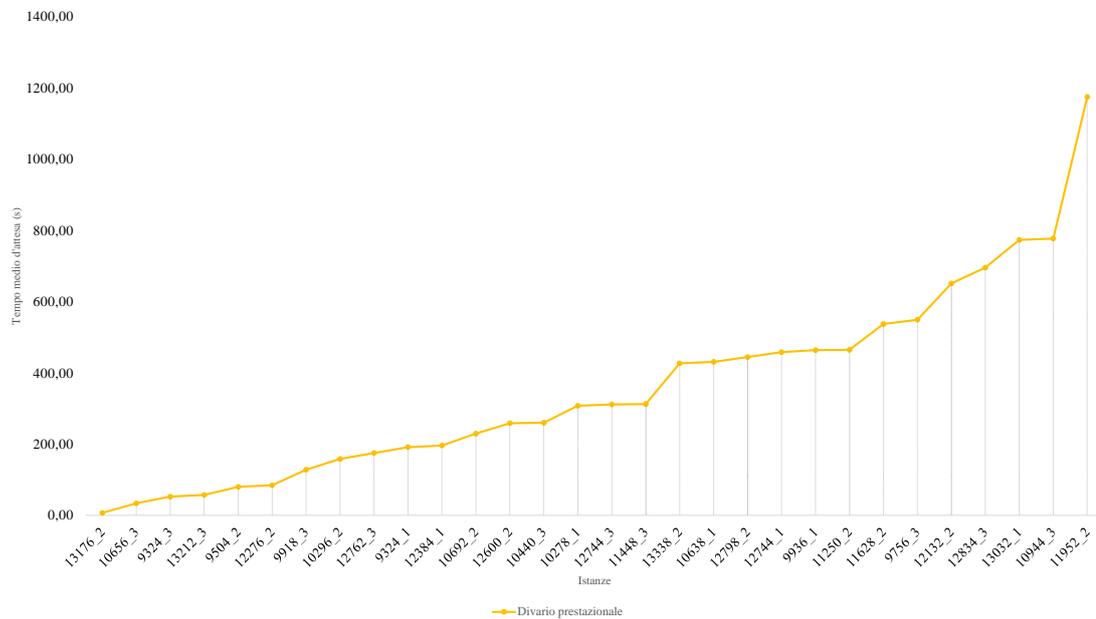


Figura 4.15: Confronto tra la configurazione migliore e peggiore per la F4 (Pasubio)



Nelle figure 4.12, 4.13, 4.14 e 4.15 si nota un andamento differente rispetto a quello ottenuto nei corrispondenti grafici dello scenario precedente. Pur non avendo la stessa linearità, il divario tra le prestazioni delle due configurazioni non è costante, anzi assume valori considerevoli, come si può notare dai picchi presenti nei tratti finali di ogni grafico. Questo comportamento conferma quanto evidenziato in 4.2.2, ovvero che, essendo la distanza tra le due prestazioni elevata, il possibile spazio di ricerca dei parametri è a sua volta vasto, rivelando il vantaggio ottenuto dal condurre un'esplorazione attraverso l'uso di un tool automatico.

Come evidenzia la tabella 4.6 nella pagina seguente, la media delle prestazioni date dalla configurazione ottimale è sempre inferiore rispetto al secondo caso, come lo è pure la mediana, ad esclusione della fascia F4, indicando quindi prestazioni globalmente migliori rispetto al caso peggiore.

Dai risultati di questi esperimenti si può quindi concludere che l'utilizzo di parameter tuning sia conveniente rispetto alla configurazione manuale dei parametri.

Tabella 4.6: Medie e mediane delle performance delle configurazioni migliori e peggiori (Pasubio)

Fascia	Migliore (s)		Peggiora (s)		Divario (s)	
	Media	Mediana	Media	Mediana	Media	Mediana
F1	41.64	39.15	51.62	51.37	10,10	11,46
F2	146.23	84.90	256.59	211.56	118,66	88,47
F3	384.51	375.65	645.62	510.44	374,13	200,79
F4	596.41	598.19	628.11	531.86	356,16	309,31
F5	642.78	621.37	775.30	694.70	348,90	257,42

Tabella 4.7: Valori delle configurazioni d'élite migliori (Pasubio)

Parametro	Range	F1	F2	F3	F4	F5
THRESHOLD	[5 - 15]	15	9	7	14	12
MIN_DEC_PHASE_D	[3000 - 7000]	4549	5818	5634	6448	6564
P_CHANGE	[0,01 - 0,9]	0,8331	0,8905	0,8256	0,4094	0,5058
GAMMA_SP	[0,1 - 1,0]	0,8106	0,3223	0,1714	0,3544	0,2482
BETA_SP	[0,1 - 0,99]	0,8704	0,4852	0,3034	0,1967	0,7880
GAMMA_NO	[0,1 - 1,0]	0,5403	0,2035	0,6367	0,1485	0,8235
BETA_NO	[0,1 - 0,99]	0,3392	0,4483	0,9996	0,4821	0,9391
THETA_MAX	[0,5 - 1,0]	0,6014	0,5086	0,8013	0,9151	0,8064
THETA_MIN	[0,05 - 0,5]	0,4324	0,4761	0,2170	0,2220	0,4532
LEARNING_COX	[0,0001 - 0,1]	0,0510	0,0512	0,0569	0,0392	0,0896
FORGETTING_COX	[0,0001 - 0,1]	0,0440	0,0118	0,0373	0,0930	0,0507

Tabella 4.8: Valori delle configurazioni d'élite peggiori (Pasubio)

Parametro	Range	F1	F2	F3	F4	F5
THRESHOLD	[5 - 15]	15	6	7	9	8
MIN_DEC_PHASE_D	[3000 - 7000]	5153	6085	4544	6488	6586
P_CHANGE	[0,01 - 0,9]	0,5692	0,0187	0,8872	0,4484	0,0402
GAMMA_SP	[0,1 - 1,0]	0,4163	0,6287	0,8862	0,7373	0,9544
BETA_SP	[0,1 - 0,99]	0,9831	0,7272	0,9484	0,7797	0,7197
GAMMA_NO	[0,1 - 1,0]	0,4757	0,3018	0,1576	0,4323	0,2166
BETA_NO	[0,1 - 0,99]	0,9849	0,4412	0,5734	0,5444	0,5571
THETA_MAX	[0,5 - 1,0]	0,5462	0,9867	0,8063	0,8233	0,8764
THETA_MIN	[0,05 - 0,5]	0,3735	0,4098	0,1810	0,0694	0,2111
LEARNING_COX	[0,0001 - 0,1]	0,0303	0,0998	0,0252	0,0181	0,0494
FORGETTING_COX	[0,0001 - 0,1]	0,0452	0,0153	0,0559	0,0140	0,0982

Conclusioni

Questo elaborato si poneva lo scopo di impiegare tecniche di *automatic parameter tuning* su di un innovativo sistema di regolazione semaforica auto-organizzante basato su concetti derivanti dal settore della *swarm intelligence*, sviluppato dal progetto europeo COLOMBO all'interno del Settimo programma quadro di ricerca (FP7).

Per comprendere meglio il contesto della gestione del traffico veicolare attraverso l'uso di semafori, è stata proposta una trattazione dello stato dell'arte delle modalità di controllo dei regolatori semaforici, partendo dal controllo a tempi fissi. Questa forma di regolazione è alla base di ogni altro sistema realizzato successivamente. In seguito, sono stati esposti i concetti del funzionamento del controllo semi-attuato e, in ultimo, dei sistemi completamente attuati, illustrando quali modifiche introducono al controllo a tempi fissi.

Una volta compreso lo stato dell'arte dei regolatori semaforici, sono state fornite la definizione e la spiegazione del concetto di auto-organizzazione, introducendo il significato di un elemento ripreso dal sistema semaforico del progetto COLOMBO: il feromone. Sono poi stati accennati casi precedenti, noti in letteratura, di approcci auto-organizzanti per la gestione del traffico e, infine, descritto il sistema proposto nel progetto COLOMBO. Si è data una visione d'insieme, illustrando come i flussi di traffico in ingresso ed uscita da un'intersezione, occultati sotto l'astrazione del feromone, fungano da valori d'ingresso a particolari funzioni dette *funzioni di stimolo*, la cui risposta è la desiderabilità della politica a cui sono associate. Queste *policy*, dette di basso livello, hanno il compito di gestire il traffico secondo differenti criteri e il dovere di scegliere quale utilizzare, sfruttando le informazioni date dalle funzioni di stimolo, ricade sulla politica di alto livello *Swarm*. Quest'ultima, come le altre, si basa su degli algoritmi, i quali hanno al loro interno dei parametri che necessitano di essere configurati accuratamente.

Ogni algoritmo, non solo quello di Swarm, basa il suo funzionamento su dei parametri i cui valori ne condizionano le prestazioni. Questo è il fulcro della regolazione dei pa-

rametri, o *parameter tuning*. Solitamente è un compito che viene condotto manualmente dall'ideatore dell'algoritmo, costando più fatica della sua stessa progettazione. Si è quindi data un'inquadratura generale degli approcci automatici alla regolazione dei parametri e, successivamente, si è focalizzata la situazione d'interesse per questo elaborato: la regolazione off-line di algoritmi. Dopo aver dato una definizione rigorosa del *tuning problem*, se ne è proposto un approccio risolutivo noto in letteratura, detto *iterated racing*, e una descrizione del software che lo implementa, *irace*, utilizzato poi per la regolazione della politica Swarm.

Si è quindi potuto procedere alla parte sperimentale, in cui è stata data una breve spiegazione di quali fossero i parametri che governano il funzionamento della Swarm policy e quali interessano le funzioni di stimolo associate alle altre politiche di basso livello. Interesse dell'elaborato era mostrare l'efficacia del *parameter tuning* attraverso la valutazione dell'estensione dello spazio dei parametri. Ciò è stato dimostrato attraverso la regolazione del gruppo di parametri legati alla politica Swarm. Per questo compito sono stati definiti due scenari. Il primo, un semplice incrocio singolo, ha dimostrato le enormi potenzialità di miglioramento prestazionale che il *parameter tuning* è capace di fornire. Ciò è stato possibile dedurlo attraverso l'individuazione del gap tra le soluzioni fornite dalle configurazioni *migliori* e quelle *peggiori*, indicativo della dimensione dello spazio dei parametri. Le configurazioni *peggiori* sono state ottenute sempre utilizzando *irace*, semplicemente invertendo il criterio di ottimalità usato per trovare le *migliori*.

L'esperimento è stato poi ripetuto su uno scenario più complesso, estratto dalla rete stradale di Bologna, in cui l'efficacia del *parameter tuning* ha trovato nuova conferma, sulla base delle forti divergenze tra le performance della configurazione migliore e della peggiore, seppur con un andamento differente rispetto a quanto riscontrato nel primo scenario, sulle cui cause si possono condurre studi futuri.

Fra gli altri i possibili sviluppi di questo lavoro vi è l'estensione del *tuning* ai parametri legati alle funzioni di stimolo delle politiche di basso livello, ignorati nell'ambito di questa tesi. Inoltre, come accennato nei risultati del primo scenario, può essere interessante una riconsiderazione delle fasce di traffico usate in questo elaborato, conducendo maggiori studi sulle dinamiche di congestione o integrando modelli comportamentali dei guidatori. Di sicuro interesse, come studio preliminare, è la verifica della necessità di partizionare in fasce il range dei valori per il flusso veicolare, confrontando questi risultati con quelli di un *tuning* effettuato considerandolo nel suo intero. Infine, i risultati forniti dal secondo scenario

portano a pensare ad un diverso approccio nella generazione dell'insieme delle istanze usate per il tuning, in modo da tenere in considerazione problematiche come i percorsi dei veicoli o la presenza di possibili colli di bottiglia nella rete stradale, fattori che non emergono nell'ambito del primo scenario. Legato a questo aspetto e al funzionamento di irace, si può condurre un ulteriore studio su come costruire un insieme di istanze ottimale per questo pacchetto.

Appendice A

Codice dello script hook-run

```
1 #!/bin/bash
2 #####
3 # This hook is to tune the policies used by SUMO software.
4 #
5 # PARAMETERS:
6 # $1 is the instance name
7 # $2 is the candidate number
8 # The rest ($* after 'shift 2') are parameters to make the .
   add file
9 #
10 # RETURN VALUE:
11 # Mean waitSteps value
12 #####
13 # In case of error, we print the current time:
14 error() {
15     echo "`TZ=UTC date`: error: $@" >&2
16     exit 1
17 }
18 # Selection of the client
19 LOCK=/var/lock/clients
20 exec 200>$LOCK
```

APPENDICE A. Codice dello script hook-run

```
21 flock -x 200
22 CLIENTS=$(ls -l ../clients | wc -l)
23 if [ $CLIENTS = 0 ]; then
24     error "no client available"
25     flock -u 200
26 fi
27 CLIENT=$(ls -lv ../clients | head -1)
28
29 # We take the client
30 rm -rf ../clients/$CLIENT
31 flock -u 200
32
33 # The instance name and the candidate id are the first
    parameters
34 INSTANCE=$1
35 CANDIDATE=$2
36
37 # Path to the SUMO software and other scripts:
38 NFS=~/.sumo/Instances
39 LOCAL=~/.sumo/swarm_hilevel_tuning
40 TUNINGPATH=$NFS/clientside/swarm_hilevel_tuning
41 SUMO=~/.bin/sumo
42 MAKEADD=$TUNINGPATH/swarm_make_additional.py
43 NETFILE=$TUNINGPATH/input_net.net.xml
44 ADDFILE=/tmp/input_additional_swarm.add.xml
45 TRIPINFO=$LOCAL/results/tripinfo_$CANDIDATE.xml
46 EVALUATOR=$TUNINGPATH/evaluator.py
47
48 # All other parameters are the candidate parameters to be
    passed to *_make_additional
49 shift 2 || exit 1
50 CAND_PARAMS=$*
51 STDOUT=c${CANDIDATE}@$CLIENT.stdout
```

```
52 STDERR=c${CANDIDATE}@${CLIENT}.stderr
53
54 # Now we can call the *_make_additional by building a
    command line with all parameters for it
55 ssh tuning@${CLIENT} python2 $MAKEADD $CAND_PARAMS 1> $STDOUT
    2> $STDERR
56
57 # Now we can call sumo
58 ssh tuning@${CLIENT} "$SUMO --net-file $NETFILE --route-files
    $INSTANCE --additional-files $ADDFILE --tripinfo-output
    $TRIPINFO --device.hbfa.probability 1 --sloppy-insert"
    1>> $STDOUT 2>> $STDERR
59
60 # Now we can call the evaluator. It will return our cost
61 ssh tuning@${CLIENT} "export MPLCONFIGDIR=/tmp ; python2
    $EVALUATOR $CANDIDATE" 2>> $STDERR
62
63 # We are done with our duty. Clean files and exit with 0 (no
    error).
64 rm -f $STDOUT $STDERR
65 ssh tuning@${CLIENT} rm -f $ADDFILE $TRIPINFO
66
67 # We free the client
68 touch ../clients/${CLIENT}
69
70 exit 0
```


Appendice B

Risultati del Parameter Tuning della Swarm policy

B.1 Regolazione di un incrocio singolo

B.1.1 Grafici comparativi tra le configurazioni migliori e peggiori

Figura B.1: Confronto tra la configurazione migliore e peggiore per la F1

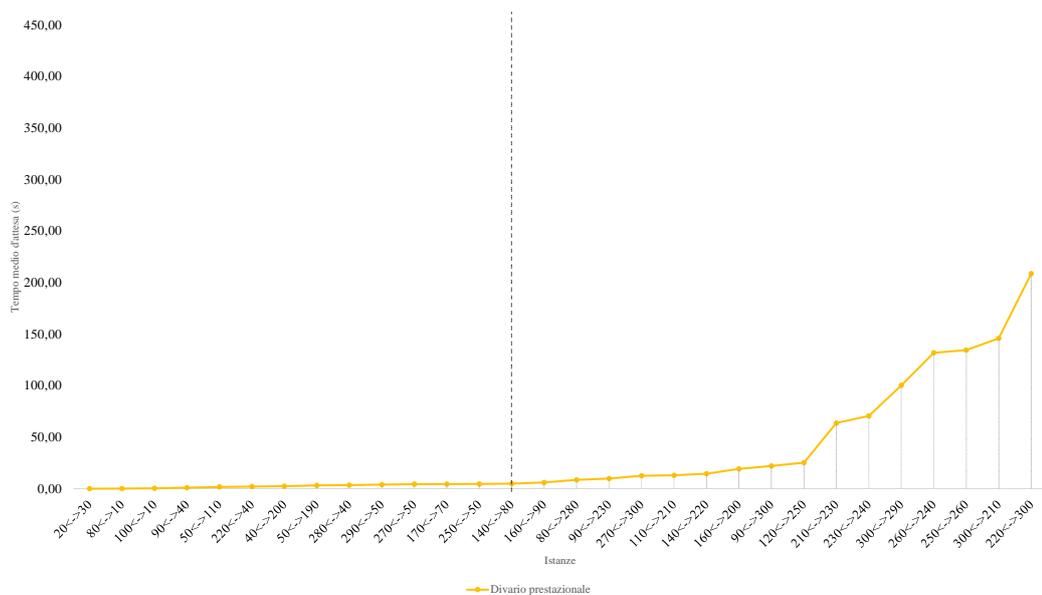


Figura B.2: Confronto tra la configurazione migliore e peggiore per la F2

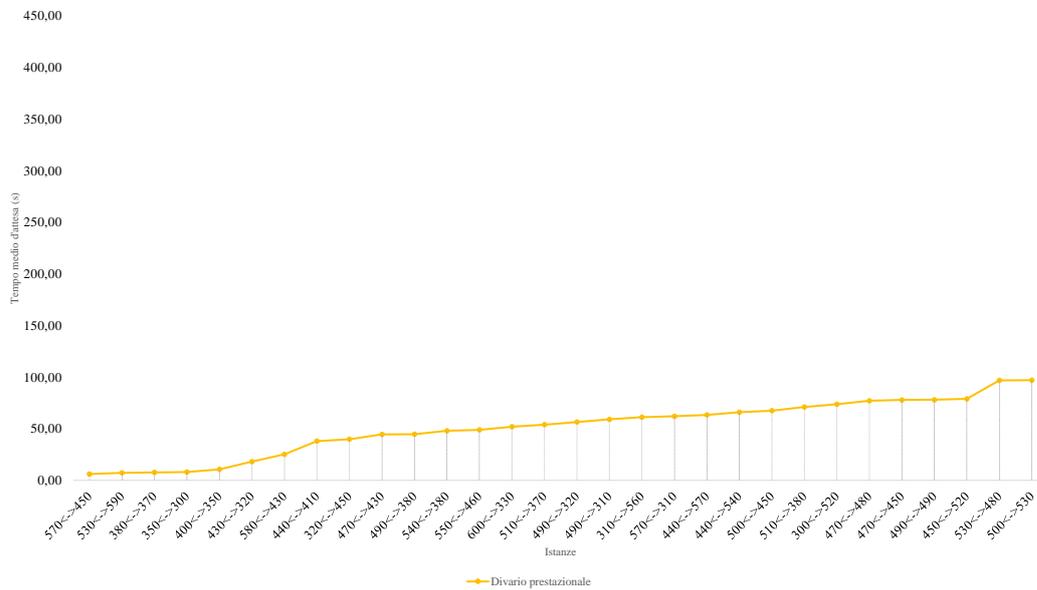


Figura B.3: Confronto tra la configurazione migliore e peggiore per la F3

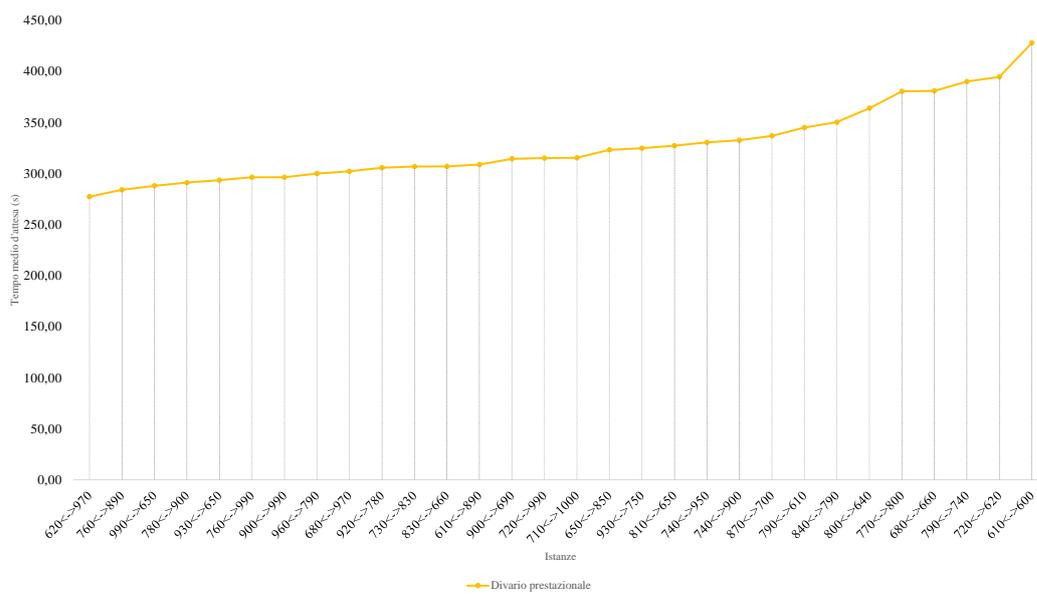


Figura B.4: Confronto tra la configurazione migliore e peggiore per la F4

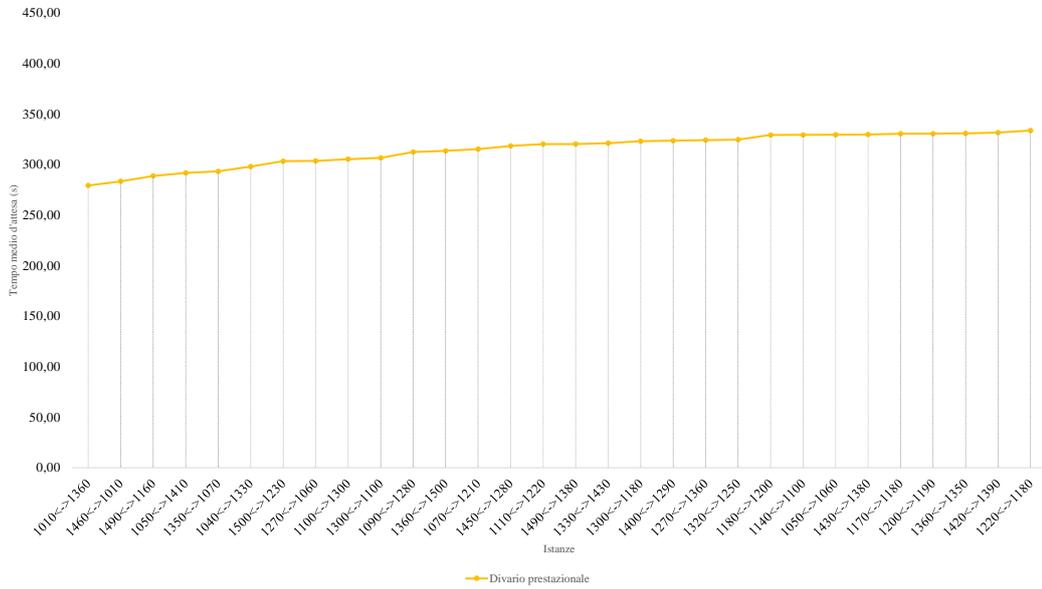
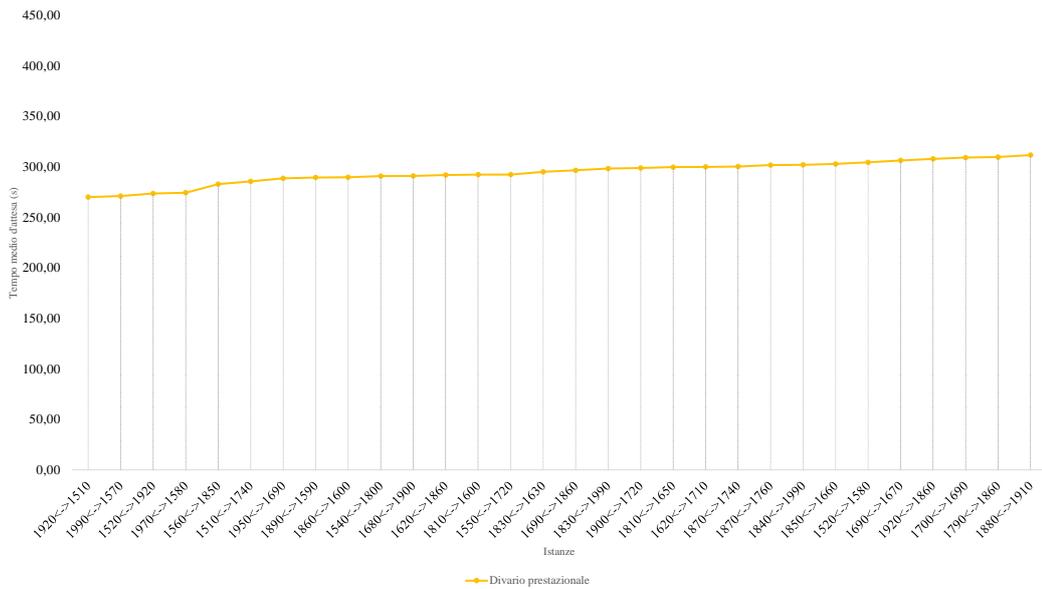


Figura B.5: Confronto tra la configurazione migliore e peggiore per la F5



B.1.2 Risultati dei confronti tra le configurazioni migliori e peggiori**Tabella B.1:** Tempi di attesa medi della configurazione migliore e peggiore per la F1

Istanze	Migliore (s)	Peggiora (s)	Divario (s)
20<->30	0,89	0,89	0,00
80<->10	0,67	0,72	0,05
100<->10	0,76	1,08	0,32
90<->40	2,24	3,20	0,96
50<->110	2,31	3,96	1,65
220<->40	2,41	4,49	2,08
40<->200	2,60	5,02	2,43
50<->190	2,29	5,46	3,16
280<->40	2,25	5,64	3,39
290<->50	2,85	6,77	3,93
270<->50	3,50	7,82	4,32
170<->70	2,75	7,16	4,41
250<->50	2,61	7,17	4,56
140<->80	3,18	7,99	4,81
160<->90	3,24	9,24	6,00
80<->280	3,71	12,23	8,51
90<->230	3,64	13,47	9,83
270<->300	8,96	21,45	12,49
110<->210	3,39	16,37	12,98
140<->220	4,57	19,02	14,45
160<->200	4,73	23,89	19,16
90<->300	4,36	26,36	22,00
120<->250	5,32	30,53	25,21
210<->230	5,54	69,22	63,68
230<->240	7,35	77,87	70,52
300<->290	10,21	110,40	100,19
260<->240	8,05	139,83	131,77
250<->260	8,15	142,44	134,29
300<->210	8,20	153,94	145,75
220<->300	8,00	216,51	208,51

Tabella B.2: Tempi di attesa medi della configurazione migliore e peggiore per la F2

Istanze	Migliore (s)	Peggiora (s)	Divario (s)
570<->450	33,12	39,20	6,08
530<->590	125,77	133,08	7,30
380<->370	16,58	24,29	7,71
350<->300	12,73	20,80	8,08
400<->350	16,26	26,86	10,59
430<->320	14,64	32,80	18,16
580<->430	42,25	67,50	25,25
440<->410	20,81	58,88	38,07
320<->450	16,83	56,66	39,83
470<->430	19,45	63,97	44,52
490<->380	20,75	65,47	44,72
540<->380	24,41	72,37	47,96
550<->460	33,44	82,35	48,91
600<->330	28,67	80,63	51,96
510<->370	21,07	75,03	53,96
490<->320	16,52	73,05	56,52
490<->310	17,51	76,64	59,13
310<->560	28,36	89,52	61,16
570<->310	26,68	88,73	62,05
440<->570	31,29	94,73	63,44
440<->540	22,64	88,70	66,06
500<->450	24,82	92,36	67,54
510<->380	18,61	89,65	71,03
300<->520	18,55	92,25	73,70
470<->480	20,27	97,32	77,04
470<->450	22,14	100,08	77,93
490<->490	38,78	116,85	78,07
450<->520	26,73	105,76	79,03
530<->480	30,21	127,06	96,85
500<->530	36,01	133,10	97,09

Tabella B.3: Tempi di attesa medi della configurazione migliore e peggiore per la F3

Istanze	Migliore (s)	Peggiora (s)	Divario (s)
620<->970	96,64	373,84	277,20
760<->890	148,03	431,89	283,86
990<->650	95,28	383,11	287,83
780<->900	155,62	446,61	290,99
930<->650	100,86	394,24	293,38
760<->990	115,05	411,25	296,20
900<->990	168,01	464,25	296,24
960<->790	132,29	432,14	299,86
680<->970	92,20	394,18	301,98
920<->780	142,55	448,02	305,47
730<->830	148,27	454,90	306,64
830<->660	115,84	422,68	306,84
610<->890	89,68	398,24	308,57
900<->690	106,03	420,27	314,24
720<->990	97,28	412,21	314,93
710<->1000	92,59	407,79	315,19
650<->850	89,26	412,21	322,96
930<->750	109,09	433,75	324,66
810<->650	104,31	431,40	327,10
740<->950	96,86	427,15	330,29
740<->900	103,25	435,66	332,41
870<->700	93,53	430,25	336,72
790<->610	71,85	416,73	344,88
840<->790	113,64	463,80	350,15
800<->640	69,83	433,64	363,81
770<->800	86,22	466,51	380,29
680<->660	86,59	467,28	380,68
790<->740	74,78	464,67	389,90
720<->620	51,58	446,02	394,44
610<->600	38,43	466,18	427,76

Tabella B.4: Tempi di attesa medi della configurazione migliore e peggiore per la F4

Istanze	Migliore (s)	Peggiora (s)	Divario (s)
1010<->1360	126,69	405,96	279,27
1460<->1010	124,70	408,02	283,32
1490<->1160	130,51	419,22	288,71
1050<->1410	128,15	419,81	291,65
1350<->1070	128,88	422,12	293,24
1040<->1330	131,03	429,01	297,97
1500<->1230	133,63	436,93	303,30
1270<->1060	134,00	437,58	303,58
1100<->1300	137,38	442,68	305,31
1300<->1100	136,01	442,54	306,53
1090<->1280	138,77	451,12	312,36
1360<->1500	143,80	457,34	313,54
1070<->1210	139,57	454,81	315,24
1450<->1280	140,60	458,98	318,38
1110<->1220	141,96	462,09	320,13
1490<->1380	140,15	460,40	320,25
1330<->1430	146,02	467,10	321,08
1300<->1180	142,19	465,24	323,05
1400<->1290	142,41	466,01	323,60
1270<->1360	146,18	470,35	324,17
1320<->1250	146,62	471,29	324,67
1180<->1200	147,82	477,04	329,22
1140<->1100	147,37	476,73	329,36
1050<->1060	147,45	476,87	329,42
1430<->1380	144,60	474,24	329,64
1170<->1180	147,27	477,77	330,50
1200<->1190	147,13	477,70	330,58
1360<->1350	148,18	478,96	330,79
1420<->1390	147,50	479,07	331,57
1220<->1180	143,29	477,01	333,72

Tabella B.5: Tempi di attesa medi della configurazione migliore e peggiore per la F5

Istanze	Migliore (s)	Peggiora (s)	Divario (s)
1920<->1510	131,59	401,59	270,00
1990<->1570	131,02	402,02	271,00
1520<->1920	132,09	405,61	273,52
1970<->1580	132,84	407,16	274,32
1560<->1850	136,16	418,96	282,80
1510<->1740	137,38	422,90	285,51
1950<->1690	139,36	427,89	288,53
1890<->1590	138,85	428,17	289,32
1860<->1600	139,63	429,21	289,58
1540<->1800	137,92	428,67	290,75
1680<->1900	140,73	431,64	290,91
1620<->1860	137,61	429,42	291,81
1810<->1600	141,56	433,73	292,17
1550<->1720	141,98	434,20	292,22
1830<->1630	142,59	437,56	294,97
1690<->1860	143,44	439,95	296,51
1830<->1990	143,81	442,03	298,22
1900<->1720	143,07	441,83	298,76
1810<->1650	142,59	442,23	299,64
1620<->1710	144,61	444,44	299,83
1870<->1740	145,40	445,64	300,23
1870<->1760	145,93	447,58	301,65
1840<->1990	145,26	447,17	301,92
1850<->1660	144,28	447,06	302,78
1520<->1580	146,30	450,68	304,37
1690<->1670	148,04	454,28	306,24
1920<->1860	148,73	456,55	307,82
1700<->1690	148,75	457,79	309,04
1790<->1860	147,42	456,99	309,58
1880<->1910	148,20	459,77	311,57

B.2 Regolazione di una rete complessa

B.2.1 Grafici comparativi tra le configurazioni migliori e peggiori

Figura B.6: Confronto tra la configurazione migliore e peggiore per la F1

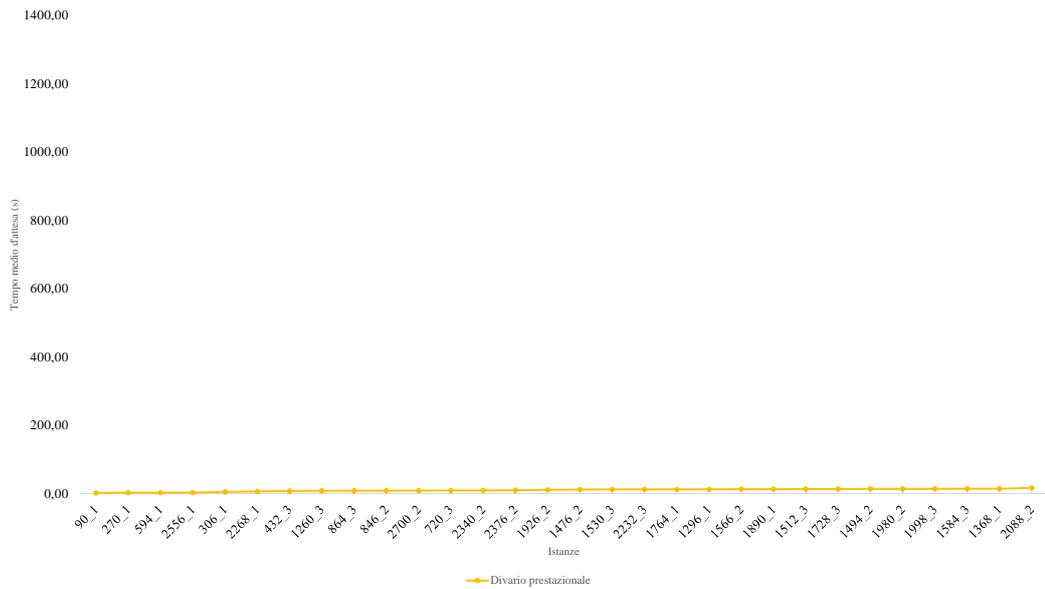


Figura B.7: Confronto tra la configurazione migliore e peggiore per la F2

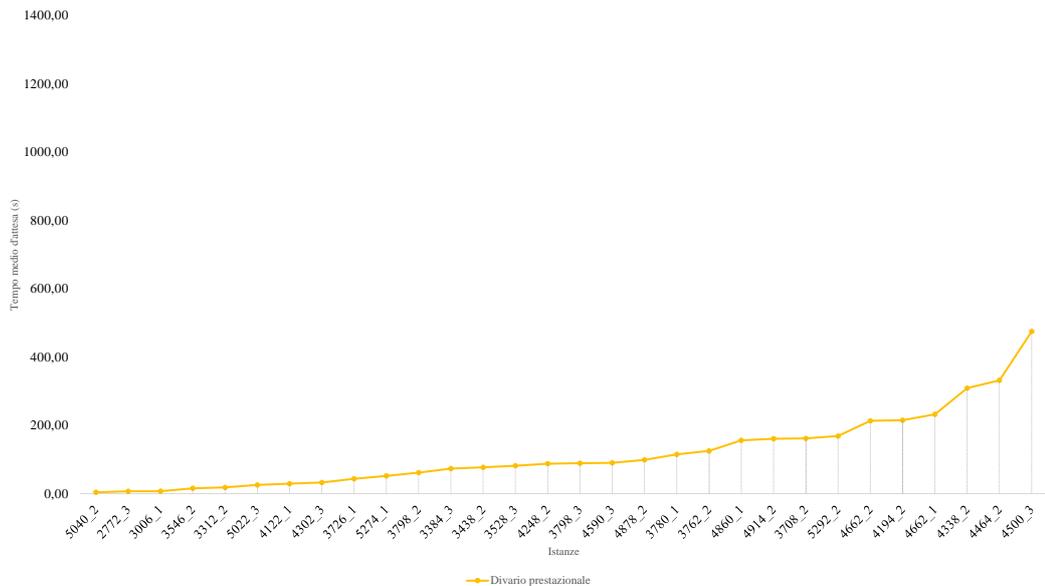


Figura B.8: Confronto tra la configurazione migliore e peggiore per la F3

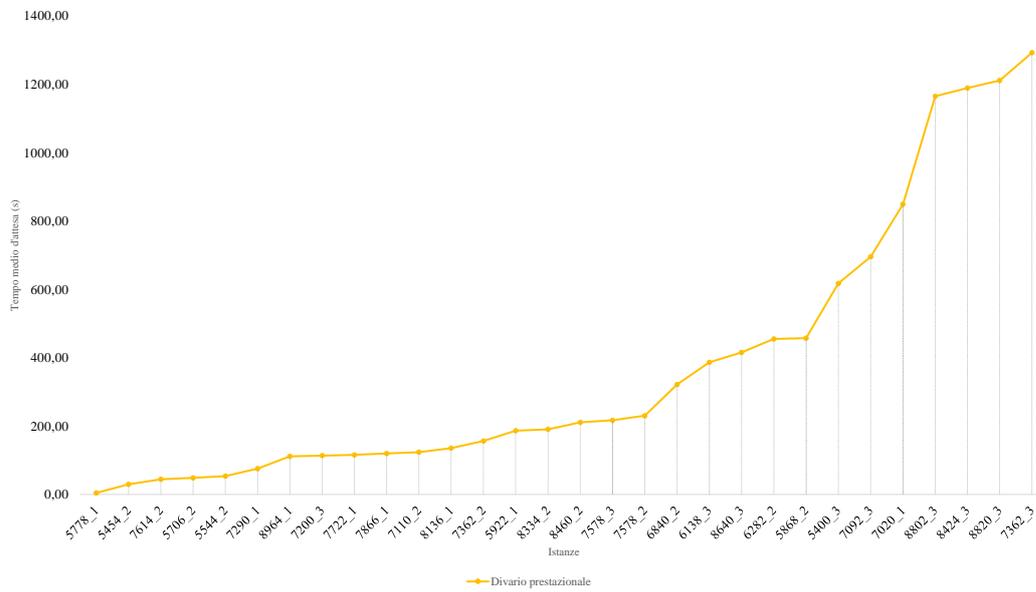


Figura B.9: Confronto tra la configurazione migliore e peggiore per la F4

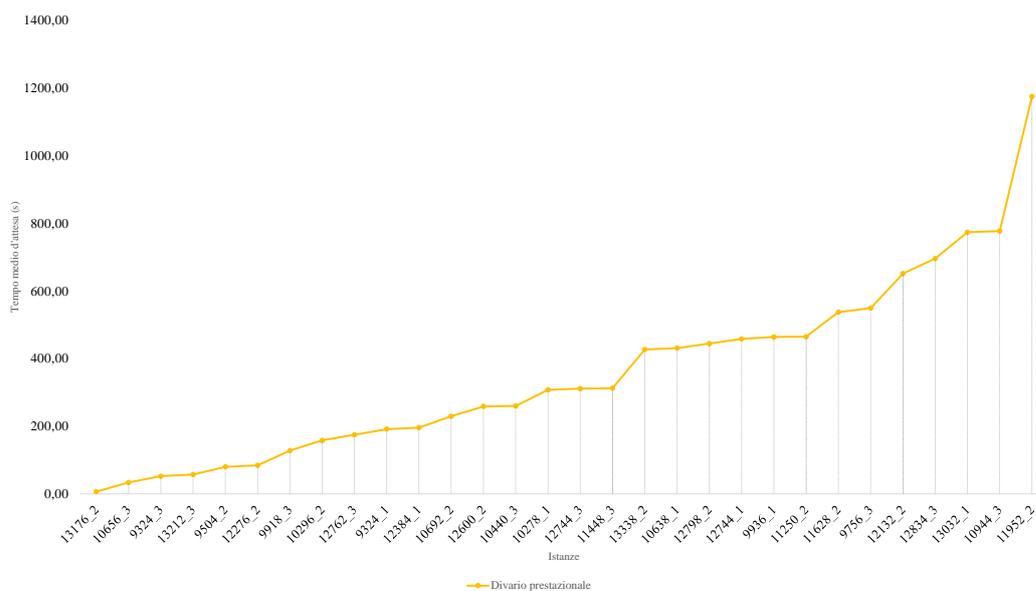
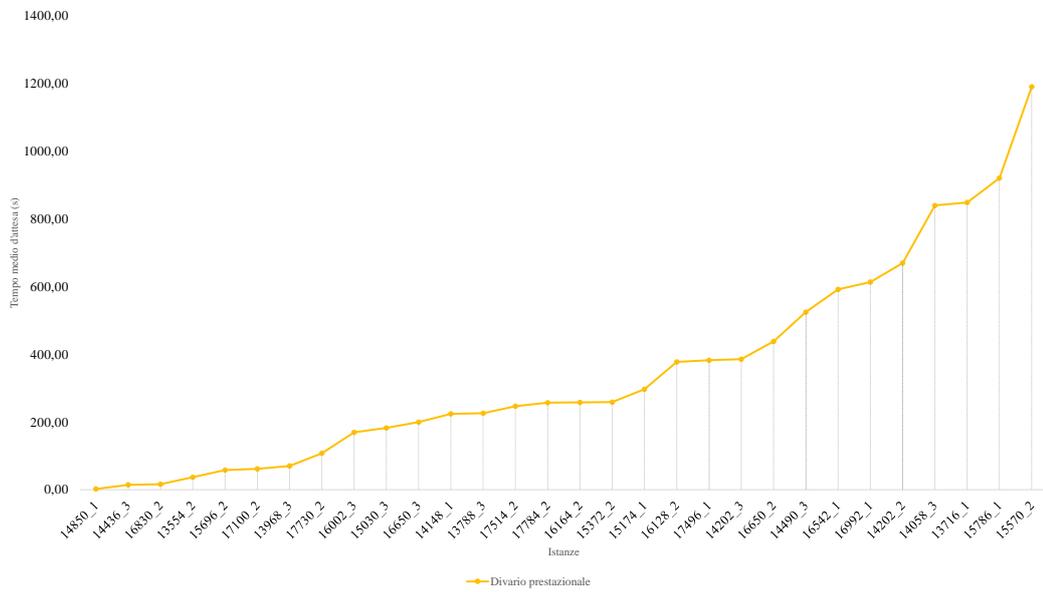


Figura B.10: Confronto tra la configurazione migliore e peggiore per la F5



B.2.2 Risultati dei confronti tra le configurazioni migliori e peggiori**Tabella B.6:** Tempi di attesa medi della configurazione migliore e peggiore per la F1

Istanze	Migliore (s)	Peggiora (s)	Divario (s)
90_1	68,54	66,70	1,84
270_1	53,21	55,63	2,42
594_1	41,05	43,59	2,54
2556_1	48,25	51,35	3,10
306_1	55,60	60,44	4,84
2268_1	45,65	52,03	6,38
432_3	48,14	55,82	7,68
1260_3	39,68	47,70	8,02
864_3	36,96	45,30	8,34
846_2	35,39	44,18	8,79
2700_2	44,39	53,28	8,89
720_3	38,58	47,59	9,01
2340_2	42,00	51,30	9,30
2376_2	41,55	51,47	9,92
1926_2	39,29	50,42	11,13
1476_2	39,91	51,70	11,79
1530_3	37,32	49,40	12,08
2232_3	40,21	52,29	12,08
1764_1	38,40	50,55	12,16
1296_1	36,58	49,37	12,79
1566_2	37,57	50,62	13,04
1890_1	38,04	51,21	13,17
1512_3	37,01	50,27	13,26
1728_3	38,06	51,52	13,46
1494_2	38,35	52,08	13,74
1980_2	39,45	53,36	13,91
1998_3	39,01	52,96	13,95
1584_3	37,23	51,38	14,15
1368_1	35,82	50,18	14,36
2088_2	38,07	54,85	16,78

Tabella B.7: Tempi di attesa medi della configurazione migliore e peggiore per la F2

Istanze	Migliore (s)	Peggiora (s)	Divario (s)
5040_2	405,61	409,80	4,19
2772_3	46,13	53,26	7,13
3006_1	46,56	53,79	7,23
3546_2	57,68	73,50	15,82
3312_2	57,63	75,90	18,26
5022_3	266,06	240,46	25,60
4122_1	84,36	113,55	29,19
4302_3	88,27	120,98	32,71
3726_1	70,17	113,84	43,67
5274_1	441,30	493,54	52,24
3798_2	69,03	130,82	61,79
3384_3	52,64	126,09	73,46
3438_2	54,52	131,76	77,24
3528_3	66,87	148,84	81,98
4248_2	85,19	173,01	87,82
3798_3	69,03	158,16	89,13
4590_3	146,22	236,47	90,25
4878_2	399,05	300,08	98,97
3780_1	69,01	184,08	115,07
3762_2	67,95	192,97	125,02
4860_1	214,38	370,41	156,02
4914_2	364,45	525,17	160,72
3708_2	68,28	230,14	161,86
5292_2	379,47	548,21	168,74
4662_2	177,26	390,64	213,38
4194_2	84,62	299,40	214,79
4662_1	130,17	362,44	232,27
4338_2	91,49	400,34	308,85
4464_2	106,76	438,58	331,81
4500_3	126,65	601,33	474,67

Tabella B.8: Tempi di attesa medi della configurazione migliore e peggiore per la F3

Istanze	Migliore (s)	Peggiora (s)	Divario (s)
5778_1	253,38	248,92	4,46
5454_2	242,92	213,43	29,49
7614_2	379,92	424,37	44,45
5706_2	285,82	334,36	48,54
5544_2	289,83	236,18	53,65
7290_1	383,59	307,95	75,63
8964_1	404,73	516,18	111,45
7200_3	338,82	225,13	113,70
7722_1	414,61	298,85	115,76
7866_1	444,24	564,24	120,00
7110_2	466,23	342,46	123,77
8136_1	494,72	359,22	135,50
7362_2	348,25	504,69	156,44
5922_1	343,57	156,82	186,75
8334_2	339,80	530,35	190,56
8460_2	545,06	334,03	211,02
7578_3	400,24	617,29	217,06
7578_2	655,75	425,56	230,19
6840_2	367,50	689,24	321,74
6138_3	328,95	715,40	386,45
8640_3	582,42	166,95	415,47
6282_2	371,38	826,37	454,99
5868_2	286,07	743,62	457,56
5400_3	233,06	850,37	617,32
7092_3	351,87	1047,39	695,53
7020_1	392,36	1241,24	848,87
8802_3	400,91	1565,87	1164,96
8424_3	383,81	1572,90	1189,09
8820_3	479,67	1690,80	1211,13
7362_3	325,87	1618,33	1292,47

Tabella B.9: Tempi di attesa medi della configurazione migliore e peggiore per la F4

Istanze	Migliore (s)	Peggiora (s)	Divario (s)
13176_2	732,79	726,10	6,69
10656_3	242,47	208,86	33,61
9324_3	342,89	290,51	52,39
13212_3	502,45	559,49	57,04
9504_2	218,89	138,96	79,93
12276_2	617,81	533,51	84,30
9918_3	222,27	350,12	127,86
10296_2	578,58	420,36	158,22
12762_3	216,68	391,41	174,74
9324_1	350,47	159,26	191,21
12384_1	626,62	430,71	195,91
10692_2	469,56	698,78	229,22
12600_2	326,77	585,13	258,35
10440_3	974,81	1234,76	259,95
10278_1	837,70	530,20	307,50
12744_3	859,31	1170,42	311,11
11448_3	697,53	385,35	312,18
13338_2	969,26	542,48	426,79
10638_1	316,52	747,22	430,70
12798_2	1082,40	638,15	444,25
12744_1	675,04	217,08	457,96
9936_1	934,79	471,14	463,66
11250_2	728,52	263,90	464,62
11628_2	745,60	208,75	536,85
9756_3	314,02	863,03	549,01
12132_2	1117,17	466,30	650,88
12834_3	654,82	1350,25	695,42
13032_1	545,22	1318,30	773,08
10944_3	545,87	1322,74	776,87
11952_2	445,57	1620,07	1174,50

Tabella B.10: Tempi di attesa medi della configurazione migliore e peggiore per la F5

Istanze	Migliore (s)	Peggiora (s)	Divario (s)
14850_1	202,64	204,81	2,18
14436_3	530,56	516,34	14,22
16830_2	654,59	638,61	15,98
13554_2	231,33	194,61	36,72
15696_2	592,76	650,66	57,90
17100_2	608,83	670,32	61,50
13968_3	991,49	921,52	69,97
17730_2	511,84	619,41	107,57
16002_3	869,13	699,81	169,32
15030_3	520,26	702,69	182,43
16650_3	753,93	554,26	199,67
14148_1	913,54	689,58	223,96
13788_3	1062,73	1288,73	226,00
17514_2	443,92	197,22	246,70
17784_2	662,54	919,48	256,94
16164_2	787,69	1045,59	257,90
15372_2	695,63	954,20	258,57
15174_1	317,22	614,01	296,79
16128_2	683,13	305,76	377,37
17496_1	959,20	576,79	382,41
14202_3	410,04	795,58	385,54
16650_2	471,50	909,61	438,11
14490_3	388,47	913,31	524,84
16542_1	209,96	801,79	591,82
16992_1	633,92	1247,47	613,55
14202_2	1205,21	535,63	669,59
14058_3	1274,78	434,93	839,85
13716_1	510,09	1358,68	848,59
15786_1	544,39	1465,01	920,62
15570_2	642,23	1832,71	1190,48

Bibliografia

- [ADL06] B. Adenso-Diaz and M. Laguna, “Fine-tuning of algorithms using fractional experimental designs and local search,” *Operations Research*, vol. 54, no. 1, pp. 99–114, 2006. [Online]. Available: <http://search.proquest.com/docview/219155307>
- [BDT99] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm intelligence: from natural to artificial systems*. Oxford university press New York, 1999, vol. 4.
- [Bir09] M. Birattari, *Tuning metaheuristics: a machine learning perspective*. Springer, 2009, vol. 197.
- [Cam01] R. Camus, “Progetto di impianti semaforici,” 02 2001. [Online]. Available: http://www.dicar.units.it/PersPage/camus/Regolazione_semaforica.pdf
- [Cap09] A. Cappelli, “Le intersezioni stradali: controllo e capacità,” 2009. [Online]. Available: <http://www.iuav.it/Ateneo1/docenti/architetto/docenti-st/Agostino-C/materiali-/ATRASPORTI/Cap-10-intersezioni.pdf>
- [CGD08] S.-B. Cools, C. Gershenson, and B. D’Hooghe, “Self-organizing traffic lights: A realistic simulation,” in *Advances in Applied Self-organizing Systems*, ser. Advanced Information and Knowledge Processing, M. Prokopenko, Ed. Springer London, 2008, pp. 41–50. [Online]. Available: http://dx.doi.org/10.1007/978-1-84628-982-8_3
- [dIedT02] M. delle Infrastrutture e dei Trasporti, “Rapporto di sintesi: norme sulle caratteristiche funzionali e geometriche delle intersezioni stradali,” Ministero delle Infrastrutture e dei Trasporti, Ispettorato generale per la circolazione e la sicurezza stradale, Tech. Rep., 2002.

- [HFB⁺13] C. Hebenstreit, M. Fellendorf, A.-C. Bellini, R. Blokpoel, M. Milano, and D. Krajzewicz, “Colombo: Deliverable 2.1 policy definition and dynamic policy selection algorithms,” COLOMBO Project Consortium, Tech. Rep., 10 2013.
- [HHS07] F. Hutter, H. H. Hoos, and T. Stützle, “Automatic algorithm configuration based on local search,” *Proceedings of the National Conference on Artificial Intelligence*, vol. 22, no. 2, p. 1152, 2007. [Online]. Available: <http://www-devel.cs.ubc.ca/~hoos/Publ/HutEtAl07.pdf>
- [LH08] S. Lämmer and D. Helbing, “Self-control of traffic lights and vehicle flows in urban road networks,” *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 04, p. P04019, 2008. [Online]. Available: <http://arxiv.org/pdf/0802.0403>
- [LIDLSB11] M. López-Ibáñez, J. Dubois-Lacoste, T. Stützle, and M. Birattari, “The irace package, iterated race for automatic algorithm configuration,” IRIDIA, Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle, Tech. Rep., 2011. [Online]. Available: <http://iridia.ulb.ac.be/IridiaTrSeries/link/IridiaTr2011-004.pdf>

Ringraziamenti

Pare impossibile, ma anche quest'avventura si è conclusa. Sono stati due anni di fuoco, costellati di momenti carichi di tensione e ansie ma, fortunatamente, non ero da solo. In tanti mi avete accompagnato, supportato e, soprattutto, supportato durante questo percorso. Di questo, ve ne sono grato.

Di certo non possono iniziare senza ringraziare chi ha reso possibile questo momento, la seconda laurea. Un sentito grazie va quindi alla mia relatrice, la professoressa Michela Milano, e il mio correlatore, Alessio Bonfietti, per la professionalità dimostrata, per avermi dato l'opportunità di misurarmi con un progetto impegnativo quale è quello in cui la mia tesi si è inserita e per avermi guidato nella realizzazione di questo prodotto.

Doveroso è il ringraziamento verso i miei genitori e mia sorella, per avermi consentito di proseguire gli studi e, soprattutto, per aver sopportato le mie innumerevoli mancanze nei loro confronti di quest'ultimo periodo, senza farmelo pesare più di tanto. Grazie Paola e Fausto, la mia famiglia "d'adozione", per farmi sentire come se fossi a casa mia, e grazie ad Elisa e Giorgio per le divertenti serate passate in vostra compagnia.

Le lezioni, i laboratori, ma in generale le ore passate all'interno dell'università non sarebbero state le stesse senza qualcuno con cui passarle assieme. Fortunatamente, c'erano i miei amici ad alleviare quei momenti, a condividere con me gioie e dolori degli esami. Simone, Gregorio e Michele, grazie per questo percorso passato assieme.

Non sarebbe però corretto limitarsi a ringraziare solo i miei compagni di università. Nel corso di questi due anni ho conosciuto nuove persone, con le quali ho instaurato nuovi rapporti di amicizia. Credo di potermi ritenere fortunato per essere stato accolto all'interno del vostro gruppo e per il divertimento che mi avete regalato. Manuel, Andrea, Antonio, Luca, grazie per il tempo che avete condiviso con me.

Giustamente, non posso terminare i ringraziamenti ai miei amici senza menzionare Ezio. Sono ormai tanti gli anni che ci conosciamo, grazie per non aver mai tradito la mia fiducia.

Ormai è una tradizione, ma non potrebbe essere diversamente. Per concludere in bellezza questi ringraziamenti, per darti il giusto risalto (perché incastrarti tra altri nomi non sarebbe rispettoso) ma soprattutto per chiudere il cerchio aperto con la seconda pagina di questa tesi, grazie Marco. Grazie perché è da tre anni che sei al mio fianco e mi hai migliorato, per il supporto che mi dai, per l'affetto di cui mi copri, per non rinfacciarmi mai le mancanze che ho nei tuoi confronti. Grazie per sostenermi, per la fiducia riponi in me, perché mi fai sentire importante. Grazie perché mi fai stare bene. Grazie perché hai scelto di condividere con me la tua vita. Grazie per i bellissimi momenti abbiamo passato assieme.