
Relazione per il corso Quantitative analysis of system

PROF. ANDREA BONDAVALLI

DOTT. PAOLO LOLLINI

DOTT. TOMMASO ZOPPI

Modellazione e Simulazione

CORSO DI LAUREA TRIENNALE IN INFORMATICA



BAJRON ISMAILAJ
MATRICOLA 2686563
ESAME DA 6 CREDITI

Indice

1	Analisi di un rete ferroviaria	1
1.1	Informazioni personali	1
1.2	Analisi del problema	1
1.3	Descrizione generale del problema	2
1.4	Descrizione atomica della linea ferroviaria	3
1.4.1	Modelli atomici	3
1.4.2	Marcatura iniziale	5
1.4.3	Probabilità e tempo di distribuzione per le attività	6
1.4.4	Condizioni d'abilitazione	6
1.5	Definizione delle variabili di guadagno	9
1.5.1	Analisi 1 : Stato della linea ferroviaria nell'istante di tempo	10
1.5.2	Analisi 1: Stato della linea ferroviaria a regime (steady-state)	10
1.5.3	Analisi 2: Numero medio dei treni che completano la corsa nell'istante di tempo	11
1.5.4	Analisi 3: Determinare i parametri ideali per i modelli che massimizzano la performance	11
1.5.5	Analisi 4: Comportamento del sistema in presenza di guasti	12
1.6	Parametri usati per lo studio	12
1.7	Simulatori di Möbius	14
1.8	Risultati ottenuti	15
1.8.1	Risultati dell'analisi uno: Sistema nell'istante di tempo	15
1.8.2	Risultati dell'analisi uno: Sistema a regime	18
1.8.3	Risultati dell'analisi due	18
1.8.4	Risultati dell'analisi tre	20
1.8.5	Risultati dell'analisi quattro	22
1.9	Conclusioni	23

Capitolo 1

Analisi di un rete ferroviaria

1.1 Informazioni personali

- Nome Cognome : Bajron Ismailaj
- Matricola : 2686563
- email : bajron.ismailaj1@stud.unifi.it

Sono un studente della laurea triennale e nel mio piano di studio ho come attività didattica la materia Modellazione e Simulazione di 6 crediti che corrisponde a la parte di modellazione di QAS.

1.2 Analisi del problema

Il progetto consiste nel valutare la performance della linea ferroviaria che collega Firenze con Viareggio. In particole in presenza di politiche diverse per la gestione della precedenza dei treni nella linea ferroviaria, si cerca di dare delle risposte e le seguenti domande:

- Qual è lo stato della linea in base ad un numero differente di treni?
- In quali casi si raggiunge la massima efficienza di transito dei treni?
- Quale politica e quali parametri massimizzano la performance della linea?
- Come si comporta la linea ferroviaria in caso di guasti?

Per dare delle risposte a tali domande si usa la teoria del l'analisi dei sistemi critici e in particolare si e interessato alla previsione dei guasti (Fault Forecasting), i mezzi dell'affidabilità e sicurezza. Si cerca di effettuare una valutazione del comportamento della linea ferroviaria. In sintesi si valuta in termini probabilistici, l'analisi quantitativa, gli attributi che soddisfano le misure che servono per determinare lo stato del sistema.

Basandoci su la teoria delle reti di Petri e in particolare le SAN (Stochastic Activity Networks) si costruisce un modello del sistema usando il tool di Möbius simuliamo il comportamento del sistema ottenendo come risultato numerico, le misure che servono a rispondere a le domande poste.

1.3 Descrizione generale del problema

La linea ferroviaria implementa tre politiche differenti per la gestione della seconda sezione, che collega Pistoia a Lucca, e sono Altrenation, PistoiaFirst e LuccaFirst. Per ognuno di queste politiche si è costruito un modello atomico SAN con lo stesso nome. I tre modelli sono simili e presentano dei cambiamenti solo nella seconda sezione. Nella figura 1.1 viene descritto il sistema atomico per la politica PistoiaFirst. Le stazioni nella linea ferroviaria sono rappresentate tramite i posti, i treni vengono rappresentati da i token nei posti e le attività con i gate di input e output descrivono il transito da una stazione all'altra.

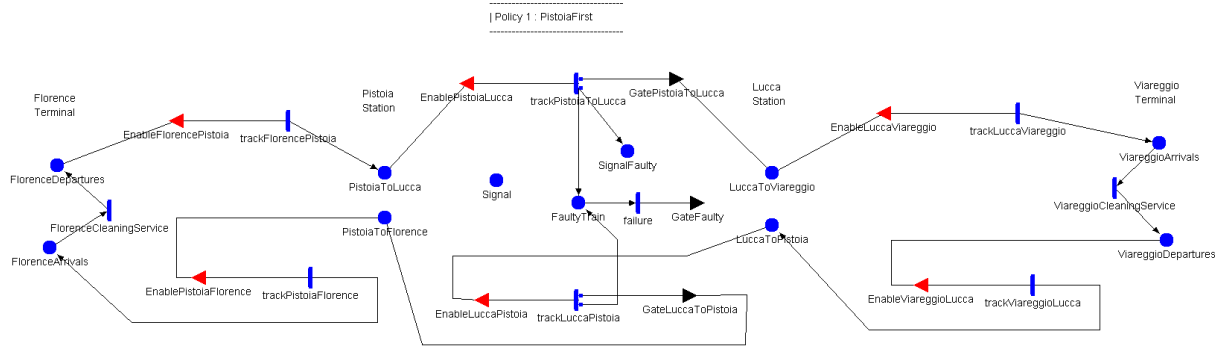


Figura 1.1: Rappresentazione della linea ferroviaria con la politica PistoiaFirst

Si ha due tipi di stazioni, i terminali Firenze e Viareggio e le stazioni minori Pistoia e Lucca. Ogni stazione viene descritto dall'insieme di due posti. I terminali Firenze e Viareggio vengono rappresentati, per ognuno, da i posti Arrivals e Departures che sono anche le piattaforme di arrivo e di partenza. I treni (token) iniziano e terminano la loro corsa nei terminali per questo si ha due sensi di marcia; il primo quello da Firenze - Viareggio e il secondo, da Viareggio - Firenze. Per creare un'idea migliore si può pensare al movimento dei treni, nei modelli, come un movimento in senso orario dei token da una stazione all'altra. La stazione minore di Pistoia è descritta da i posti, PistoiaToLucca per il primo senso di marcia e PistoiaToFlorence per il secondo senso di marcia. Stessa rappresentazione anche per la stazione di Lucca con i posti LuccaToViareggio e LuccaToPistoia.

Le attività descrivono il movimento dei treni. Per esempio un treno a FirenzeDepartures, con le condizioni di abilitazione del gate di input EnableFlorencePistoia soddisfaccibili, e con lo scatto dell'attività trackFlorencePistoia, muove un treno a PistoiaToLucca. Tutte le attività hanno come riferimento di misura del tempo l'unità di 1 ora.

Le attività sono di tre tipi

- Quelle di transito (track) che muovono i treni da una stazione all'altra.
- Quelle di pulizia, CleaningService, dove i treni che arrivano nelle piattaforme d'arrivo dei terminali vengono sottoposti al servizio di pulizia.
- E infine l'attività di failure che descrive il guasto della locomotiva nella sezione due.

Le sezioni uno e due sono identiche per tutte e tre i tipi di modelli atomici della linea ferroviaria.

Un classico percorso di un treno sarebbe il seguente:
 Il treno presente in FirenzeDepartures transita a PistoiaToLucca, condizioni e attivazioni permettendo. A Pistoia, se le condizioni imposte da la politica scelta nella sezione due sono soddisfatte e nessun altro treno transita su tale binario allora lo scatto dell'attività trackpistoiaToLucca sposta il treno a Lucca, LuccaToViareggio. Il treno prosegue la corsa, condizioni e attivazioni permettendo, verso Viareggio, ViareggioArrivals, dove viene sottoposto al servizio di pulizia per poi muoversi verso la piattaforma di partenze per proseguire la corsa con senso di marcia Viareggio - Firenze. Lo scatto di trackViareggioLucca sposta il treno a Lucca dove se le condizioni del gate di input EnableLuccaPistoia e lo scatto dell'attività trackLuccaPistoia comporta il transito del treno a Pistoia. Prosegue la corsa verso Firenze dove viene sottoposto al servizio di pulizia. Nella sezione due possono verificarsi dei guasti nella locomotiva e in questo caso il treno viene messo a FaultyTrain dove aspetta che si ripristina di nuovo la corsa.

1.4 Descrizione atomica della linea ferroviaria

La linea è composta da tre sezioni che sono Firenze - Pistoia, Pistoia - Lucca e Lucca - Viareggio. In seguito si ha la descrizione dei modelli atomici per le tre politiche considerando le sezioni individualmente. Una spiegazione più approfondita del comportamento della linea avvera nella descrizione delle attività e delle porte di input.

1.4.1 Modelli atomici

Il modello atomico San: Alternation è più semplice, infatti nella sezione due presenta meno posti, questo perché si cerca di minimizzare il numero dei stati generati 1.2.

Tutti i modelli sono stabili non presentano attività istantanee e di conseguenza non si genera una sequenza

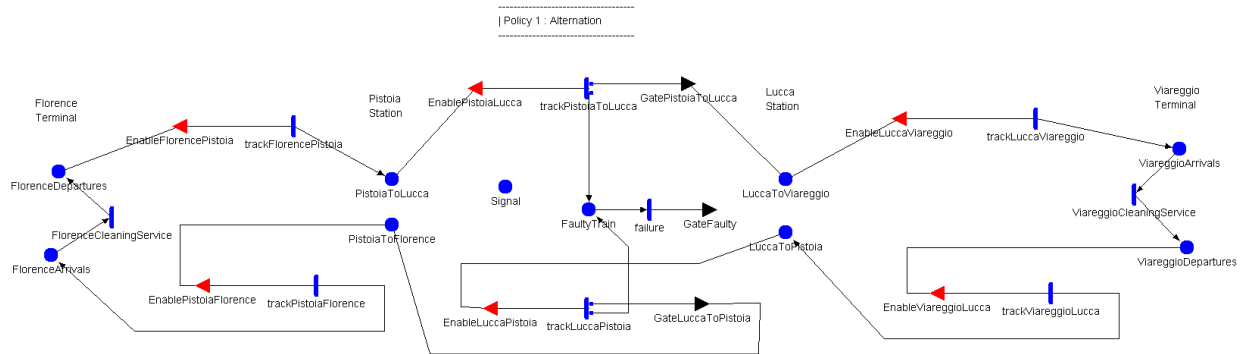


Figura 1.2: Rappresentazione della linea ferroviaria con la politica Alternation

infinita di marcature instabili. Inoltre i modelli sono ben specificati.

I processi stocastici associato ai cambiamenti delle marcatura hanno tutte le attività con distribuzione di tempo esponenziale di conseguenza i modelli sono dei HCTMC (Homogenous Continuous Time Markoc Chain). Inoltre visto che si tratta di processi generativi di Markov si può usare il tool Möbius per l'analisi analitica e per la simulazione.

Sezione uno della linea ferroviaria

La sezione uno della linea ferroviaria è identica per tutte e tre i modelli 1.3.

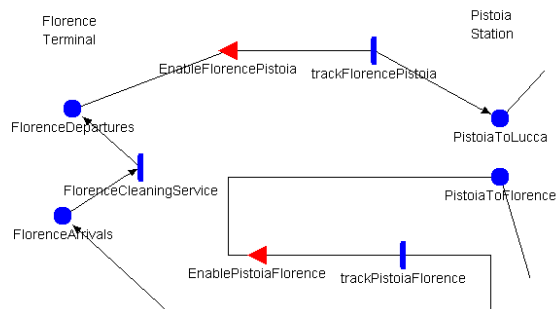


Figura 1.3: Rappresentazione della sezione 1

Un treno (token) presente nella piattaforma di partenza del terminale Firenze aspetta finché le condizioni del gate EnableFlorencePistoia sia vero e con lo scatto dell'attività trackFirenzePistoia transita da Firenze a Pistoia secondo il senso di marcia Firenze - Viareggio.

Un token nel posto PistoiaToFlorence allo scattare dell'attività trackPistoiaFlorence si sposta nella FlorenceArrivals dove viene sottoposto al servizio di pulizia.

Sezione due per il modello Alternation

Nella sezione due si ha un solo binario e nel modello Alternation i treni su le stazioni di Pistoia e Lucca alternano il passaggio 1.4.

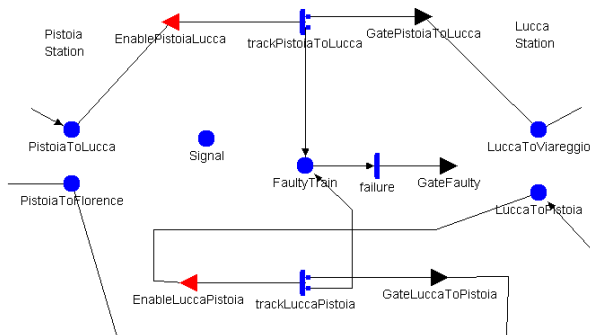


Figura 1.4: Rappresentazione della sezione 2 per il modello Alternation

La presenza di un token sul posto Signal indica il senso di marcia Firenze - Viareggio, altrimenti se nessun token si trova su Signal il senso di marcia è quello di Viareggio - Firenze.

Un token in FaultyTrain indica che la locomotiva è in guasto e il binario non è libero. Lo scatto dell'attività

failure descrive il ripristino della marcia. Il gate di output GateFaulty sposta i token rispettando il senso di marcia interrotto in base alla presenza del token su Signal, e per poi alternare il valore in esso.

Sezione due per i modelli PistoiaFirst e LuccaFirst

La sezione due per i modelli PistoiaFirst e LuccaFirst è identica esternamente però c'è sono dei cambiamenti nelle condizioni d'attivazione. 1.5

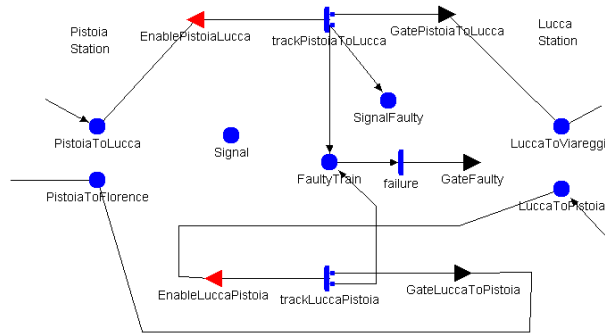


Figura 1.5: Rappresentazione della sezione 2 per i modelli PistoiaFirst e LuccaFirst

Diversamente dal esempio precedente la presenza di un token sul posto Signal significa che sul binario unico non c'è sono treni in transito o in guasto, e di conseguenza, i treni che si trovano nelle stazioni di Pistoia (PistoiaToLucca) e Lucca (LuccaToPistoia) possono transitare se la politica scelta e rispettata e le condizioni di abilitazione e scatto dell'attività sono vere.

Un token in FaultyTrain indica che la locomotiva è in guasto e il binario non è libero. La presenza di un token sul posto SignalFaulty tiene memoria del senso di marcia Firenze - Viareggio altrimenti si considera il senso opposto. Lo scatto dell'attività failure descrive il ripristino della marcia. Il gate di output GateFaulty sposta i token rispettando il senso di marcia interrotto in base alla presenza del token su SignalFaulty. Inoltre assegna un token sul posto Signal per indicare che il binario è libero.

Sezione tre della linea ferroviaria

La sezione tre della linea ferroviaria è identica per tutte e tre i modelli 1.6.

Un treno (token) presente nella stazione LuccaToViareggio aspetta lo scatto dell'attività trackLucca-Viareggio per poi spostarsi a le piattaforme d'arrivo di Viareggio. Il treno viene sottoposto al servizio di pulizia, ViareggioCleaningService per poi essere pronto per la partenza secondo il senso di marcia Viareggio - Firenze.

Se le condizioni e lo scatto di trackViareggioLucca, avvengono, il treno si sposta a LuccaToPistoia.

1.4.2 Marcatura iniziale

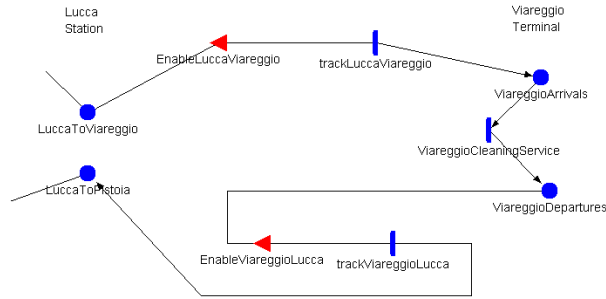


Figura 1.6: Rappresentazione della sezione 3

Dato N interno pari che rappresenta il numero dei treni presenti nel sistema allora la marcatura iniziale dei modelli è

- $N/2$ token su i posti FlorenceDepartures e ViareggioDepartures
- Un token al posto Signal che nel modello uno indica che i treni di Pistoia hanno la precedenza su Lucca. Invece per i modelli PistoiaFirst e LuccaFirst indica che il binario della sezione due è libero.

1.4.3 Probabilità e tempo di distribuzione per le attività

Le attività temporizzate presenti nel sistema sono tutte con tempo di distruzione esponenziale. L'unità di tempo selezionata è di $t = 1$ ora e le attività si sincronizzano su tale misura. La durata delle attività è identica per tutte e tre i modelli.

Attività della linea ferroviaria			
Nome dell'attività	Parametri di distribuzione Code	Primo case	Secondo case
FlorenceCleaningService	2.0	1-h	h
ViareggioCleaningService	2.0		
failure	2.0/3.0		
trackFlorencePistoia	2.0		
trackLuccaPistoia	1.5		
trackLuccaViareggio	1.2	1-h	h
trackPistoiaFlorence	2.0		
trackPistoiaToLucca	1.5		
trackViareggioLucca	1.2		

Le attività trackLuccaPistoia e trackPistoiaLucca che gestiscono il transito sulla sezione due sono ripetitivamente composte da due case. Una volta che l'attività scatta si ha la probabilità h che la locomotiva presenta un guasto.

1.4.4 Condizioni d'abilitazione

Condizioni per la sezione uno

Le condizioni di abilitazione dei gate di input per la sezione uno è identica per tutte e tre i modelli. Per il gate di input EnableFlorencePistoia si ha le seguenti condizioni

Input Gate:	EnableFlorencePistoia
Predicate	$((\text{PistoiaToLucca} \rightarrow \text{Mark}() + \text{PistoiaToFlorence} \rightarrow \text{Mark}()) < P)$ $\&\& (\text{FlorenceDepartures} \rightarrow \text{Mark}() > 0)$
Function	$\text{FlorenceDepartures} \rightarrow \text{Mark}() --;$

Se il numero dei treni presenti a Pistoia è minore della della variabile globale P e se a Firenze c'i sono dei treni che aspettano per partire allora l'attività trackFlorencePistoia è abilitata e una volta che scatta sposta un treno da Firenze a Pistoia.

Per le condizioni di abilitazione per l'attività trackPistoiaFlorence è la seguente

Input Gate:	EnablePistoiaFlorence
Predicate	$(\text{PistoiaToFlorence} \rightarrow \text{Mark}() > 0)$
Function	$\text{PistoiaToFlorence} \rightarrow \text{Mark}() --;$

L'attività è abilitata con la sola presenza di un treno sul posto PistoiaToFlorence e quando scatta sposta il treno nella piattaforma dei arrivi di Firenze.

Condizioni della sezione due con la politica Alternation

L'unico binario della sezione due alterna il passaggio dei treni tra Pistoia e Lucca cominciando da Pistoia. La presenza di un token a Signal gestisce questo comportamento.

Input Gate:	EnablePistoiaLucca
Predicate	$(\text{FaultyTrain} \rightarrow \text{Mark}() == 0) \&\& (\text{PistoiaToLucca} \rightarrow \text{Mark}() > 0)$ $\&\& ((\text{LuccaToPistoia} \rightarrow \text{Mark}() + \text{LuccaToViareggio} \rightarrow \text{Mark}()) < L + 1) \&\& (\text{Signal} \rightarrow \text{Mark}() == 1)$
Function	$\text{PistoiaToLucca} \rightarrow \text{Mark}() --;$

Se il treno non è in guasto cioè si ha un binario libero, e il numero dei treni in Lucca è al massimo L , e un treno si trova a Pistoia secondo il senso di marcia Firenze- Viareggio allora l'attività trackPistoiaLucca è abilitata e se scatta sposta un treno a LuccaToViareggio.

Se non si ha un token in Signal vuol dire che è attiva il senso di marcia Viareggio - Firenze.

Input Gate:	EnableLuccaPistoia
Predicate	$((\text{PistoiaToLucca} \rightarrow \text{Mark}() + \text{PistoiaToFlorence} \rightarrow \text{Mark}()) < P + 1) \&\& (\text{FlorenceDepartures} \rightarrow \text{Mark}() > 0)$
Function	$\text{FlorenceDepartures} \rightarrow \text{Mark}() --;$

Se il binario è libero da treni in guasto e se i treni a Pistoia sono al massimo P e su LuccaToPistoia si ha un treno allora l'attività trackLuccaViareggio è abilitata e se scatta transita il treno a Pistoia rispettando il secondo senso di marcia.

I gate di output sono semplici, aggiungono il token secondo il senso di marcia e alternano il valore del posto Signal per dare la precedenza al senso di marcia opposto.

Condizioni della sezione due con la politica PistoiaFirst

La presenza di un token sul posto Signal significa che sul binario unico non c'è sono treni in transito o in guasto. I treni presenti su la stazione di Pistoia che devono raggiungere Viareggio hanno la priorità su i treni che sono a Lucca e devono raggiungere Pistoia.

Input Gate:	EnablePistoiaLucca
Predicate	$(\text{PistoiaToLucca} \rightarrow \text{Mark}() > 0) \ \&\& \ ((\text{LuccaToPistoia} \rightarrow \text{Mark}() + \text{LuccaToViareggio} \rightarrow \text{Mark}()) < L+1) \ \&\& (\text{Signal} \rightarrow \text{Mark}() == 1)$
Function	$\text{PistoiaToLucca} \rightarrow \text{Mark}() --; \text{Signal} \rightarrow \text{Mark}() = 0;$

Se sul posto PistoiaToLucca è presente almeno un treno allora si controlla che a Lucca si ha posto per un nuovo treno, allora si attiva l'attività e nel caso che scatta rimuove il treno da Pistoia e indica che il binario non è più libero poiché c'è un treno in transito.

Invece se su Lucca si ha un treno pronto per Pistoia, si controlla che su Pistoia non ci siano treni presenti e allora si attiva l'attività trackLuccaPistoia che se scatta esegue la regola di completamento aggiungendo un token a PistoiaToFlorence.

Input Gate:	EnableLuccaPistoia
Predicate	$(\text{PistoiaToLucca} \rightarrow \text{Mark}() == 0) \ \&\& (\text{LuccaToPistoia} \rightarrow \text{Mark}() > 0) \ \&\& (\text{Signal} \rightarrow \text{Mark}() == 1)$
Function	$\text{LuccaToPistoia} \rightarrow \text{Mark}() --; \text{Signal} \rightarrow \text{Mark}() = 0;$

Visto che a PistoiaToLucca non c'è sono dei treni ci saranno abbastanza piattaforme per ospitare i treni in arrivo e di conseguenza non c'è bisogno di un controllo per tale condizione.

Se su la sezione due si ha un guasto della locomotiva vuol dire che si è scelto il case due e si ha un token sul FaultyTrain. In caso che a scattare sia l'attività trackPistoiaToLucca del senso di marcia Firenze - Viareggio si avrà anche un token in SignalFaulty per tener conto del senso di corsa. Nel gate di output quando il treno ripristina la corsa dopo il guasto si ha:

Output Gate:	GateFaulty
Function	$\text{if } (\text{SignalFaulty} \rightarrow \text{Mark}() == 1) \{ \text{LuccaToViareggio} \rightarrow \text{Mark}() ++; \text{Signal} \rightarrow \text{Mark}() = 1; \text{SignalFaulty} \rightarrow \text{Mark}() = 0; \} \text{else} \{ \text{PistoiaToFlorence} \rightarrow \text{Mark}() ++; \text{Signal} \rightarrow \text{Mark}() = 1; \}$

Se siamo al senso di marcia Firenze - Viareggio cioè un token in SignalFaulty, spostiamolo in LuccaToViareggio e mettiamo un token su Signal per indicare il fatto che il binario libero.

Condizioni nella sezione due con la politica LuccaFirst

Se non si ha un treno in guasto e se il binario è libero allora i treni che partono da Lucca verso Pistoia hanno la precedenza su i treni che partono da Pistoia verso Lucca.

Input Gate:	EnablePistoiaLucca
Predicate	(PistoiaToLucca->Mark() > 0) && (LuccaToPistoia->Mark()==0) && (Signal->Mark()==1)
Function	PistoiaToLucca->Mark() --; Signal->Mark();

Infatti se non si ha un treno che aspetta in LuccaToPistoia allora i treni che aspettano ad Pistoia possono procedere al transito verso Lucca.

Altrimenti, la presenza di un treno a LuccaToPistoia ha priorità maggiore e se il binario è libero, abilita l'attività.

Input Gate:	EnableLuccaPistoia
Predicate	(PistoiaToLucca->Mark() < P + 1) && (LuccaToPistoia->Mark() > 0) && (Signal->Mark()==1)
Function	LuccaToPistoia->Mark() --; Signal->Mark();

Se l'attività trackLuccaPistoia scatta allora la funzione del gate viene eseguita, cambiando lo stato della sezione. Muove il treno da la stazione di Lucca e rimuove il token dal Signal in questo modo indica che il binario è occupato, il treno è in transito.

I gate di output sono identici della sezione due del modello PistoiaFirst.

Condizioni per la sezione tre

Le condizioni dei gate di input per la sezione tre è identica per tutte e tre i modelli. Per il gate di input EnableLuccaViareggio si ha le seguenti condizioni

Input Gate:	EnableLuccaViareggio
Predicate	(LuccaToViareggio->Mark() > 0)
Function	LuccaToViareggio->Mark() --;

L'attività è abilitata se c'è la presenza di un treno in LuccaToViareggio.

Per le condizioni di abilitazione per l'attività trackViareggioLucca si ha la seguente

Input Gate:	EnablePistoiaFlorence
Predicate	(PistoiaToFlorence->Mark() > 0)
Function	PistoiaToFlorence->Mark() --;

Se il numero dei treni a Lucca è minore del numero della variabile globale L e se nella piattaforma di partenza di Viareggio sono presenti dei treni allora l'attività trackViareggioLucca è abilitata e una volta che scatta sposta un treno da Viareggio a Lucca.

1.5 Definizione delle variabili di guadagno

Per ogni modello atomico (Alternation, PistoiaFirst e LuccaFirst) si considera il processo stocastico sottostante e si costruisce l'insieme RS delle marcature raggiungibili da la marcatura iniziale. In questo modo si costruisce la variabile casuale V_t che rappresenta la struttura di guadagno.

$$V_t = \sum_{m \in RS} \mathcal{R}(m) I_t^m + \sum_{a \in \mathcal{T}} \mathcal{C}(a) I_t^a$$

Si hanno le seguenti variabili di guadagno.

1.5.1 Analisi 1 : Stato della linea ferroviaria nell'istante di tempo

La definizione della variabile di performance deve rispondere alla seguente domanda. qual è lo stato del sistema? Più precisamente, qual è il numero dei treni nelle stazioni in determinati istanti di tempo?

Dati otto differenti istanti di tempo $t = 1, 6, 12, 24, 48, 96, 192, 384$ in ore, c'i si domanda quanti treni mediamente si trovano in ognuna delle stazioni della linea ferroviaria istanti di tempo definiti e per ogni politica diversa sul sistema?

Per ogni politica sul sistema (selezioniamo Alternation), selezionato l'istante scelto e selezionata la stazione da analizzare (per esempio Firenze) allora si definisce la variabile di performance A che determina il numero dei treni nella stazione cioè il numero dei token presenti nei posti FlorenceArrivals e FlorenceDepartures nei istanti di tempo t .

$$E[A] = E[V_t] \text{ con struttura di guadagno basato su guadagno a tasso (rate reward)}$$

dove il tasso è della forma:

$$\mathcal{R}(m) = \begin{cases} \text{Somma del numero dei token presenti nei posti FlorenceArrivals e FlorenceDepartures} \\ 0 & \text{altrimenti} \end{cases} \quad (1.1)$$

Per il modello Alternation cioè la politica Alternation si ha la seguente struttura in Möbius.

Performance Variable : TrainToFlorence		
Affecting Models	Alternation	
Impulse Functions		
Reward Function	(Reward is over all Available Models) return (Alternation->FlorenceDepartures->Mark() + Alternation->FlorenceArrivals->Mark());	
Simulator Statistics	Type	Instant of Time
	Options	Estimate Mean
		Include Lower Bound on Interval Estimate
		Include Upper Bound on Interval Estimate
		Estimate out of Range Probabilities
		Confidence Level is Relative
	Parameters	Start Time 1,6,12,24,48,96,192,384,
Confidence	Confidence Level	0.95
	Confidence Interval	0.1

Figura 1.7: Struttura di guadagno per definire il numero dei treni nella stazione di Firenze

La stessa rappresentazione per le strutture di guadagno per la determinazione del numero medio dei treni nelle altre stazioni. Per le stazione di Pistoia descritta da i posti PistoiaToLucca e PistoiaToFlorence, per la stazione di Lucca rappresentata da i posti LuccaToViareggio e LuccaToPistoia e infine per la stazione di Viareggio descritta da i posti ViareggioArrivals e ViareggioDepartures. Figura 1.8

1.5.2 Analisi 1: Stato della linea ferroviaria a regime (steady-state)

Visto che si tratta di modelli che sono dei processi Markoviani rigenerativi allora sono ben definiti il comportamento del sistema a regime.

Simile all'analisi 1 definita nell'istante ti tempo, questa analisi determina lo stato della linea ferroviaria a

Performance Variable : TrainToPistoia				Performance Variable : TrainToLucca				Performance Variable : TrainToViareggio			
Affecting Models	Alternation			Affecting Models	Alternation			Affecting Models	Alternation		
Impulse Functions				Impulse Functions				Impulse Functions			
Reward Function	<i>(Reward is over all Available Models)</i>			Reward Function	<i>(Reward is over all Available Models)</i>			Reward Function	<i>(Reward is over all Available Models)</i>		
	return (Alternation->PistoiaToLucca->mark() + Alternation->PistoiaToFlorence->mark());				return (Alternation->LuccaToViareggio->mark() + Alternation->LuccaToPistoia->mark());				return (Alternation->ViareggioDepartures->mark() + Alternation->ViareggioArrivals->mark());		
Simulator Statistics	Type	Instant of Time		Type	Instant of Time		Type	Instant of Time			
	Options	Estimate Mean		Options	Estimate Mean		Options	Estimate Mean			
		Include Lower Bound on Interval Estimate			Include Lower Bound on Interval Estimate			Include Lower Bound on Interval Estimate			
		Include Upper Bound on Interval Estimate			Include Upper Bound on Interval Estimate			Include Upper Bound on Interval Estimate			
		Estimate out of Range Probabilities			Estimate out of Range Probabilities			Estimate out of Range Probabilities			
	Confidence Level is Relative		Confidence Level is Relative		Confidence Level is Relative						
	Parameters	Start Time	1,6,12,24,48,96,192,384,	Parameters	Start Time	1,6,12,24,48,96,192,384,	Parameters	Start Time	1,6,12,24,48,96,192,384,		
	Confidence	Confidence Level 0.95		Confidence	Confidence Level 0.95		Confidence	Confidence Level 0.95			
Confidence Interval 0.1		Confidence Interval 0.1			Confidence Interval 0.1						

Figura 1.8: Struttura di guadagno per definire il numero dei treni nelle stazione di Pistoia, Lucca e Viareggio

regime. Più precisamente per ogni modello (scegliamo Alternation), per ogni stazione desiderata (nel esempio Firenze) allora si definisce la variabile di performance B che determina il numero dei treni che sostano a Firenze quando il sistema è a regime.

$$E[B] = E[V_{t,t \rightarrow \infty}] \text{ con struttura di guadagno basato su guadagno a tasso (rate reward)}$$

La struttura di guadagno è identica all'analisi 1 con tempo all'istante.

Per il modello atomico Alternation si ha le seguenti definizioni di variabili di guadagno a tasso 1.9.

1.5.3 Analisi 2: Numero medio dei treni che completano la corsa nell'istante di tempo

Per ogni dei tre modelli atomici si vuole determinare il numero dei treni che completano la corsa nei intervalli di tempo da $[0, t], t \in \{1, 6, 12, 24, 48, 96, 192, 384\}$.

Un treno termina la corsa se arriva nei terminali Firenze o Viareggio, cioè se scatta l'attività trackLuccaViareggio o trackPistoiaFlorence. Più precisamente si vuole definire la variabile di performance C che accumula guadagno ogni volta che le due attività scattano nel determinato intervallo.

$$E[C] = E[V_{0,t}] \text{ con struttura di guadagno basato su guadagno a impulso (impulse reward)}$$

$$C(a) = \begin{cases} 1 & \text{Se scatta l'attività trackLuccaViareggio o trackPistoiaFlorence} \\ 0 & \text{altrimenti} \end{cases} \quad (1.2)$$

Sul tool Möbius si ha la seguente definizione per la politica Alternation 1.10.

Stessa definizione per gli modelli atomici PistoiaFirst e LuccaFirst.

Ovviamente è desiderabile avere un numero maggiore di treni che terminano la corsa, e se questa lo associamo all'efficienza della linea ferroviaria diciamo che si studia l'efficienza del sistema. Si cerca di determinare quando si ha una linea più efficiente in presenza di un numero differente di treni.

1.5.4 Analisi 3: Determinare i parametri ideali per i modelli che massimizzano la performance

Per ogni modello atomico si vuole determinare il valore minimo dei parametri P e L che massimizza il numero dei treni che terminano la corsa, cioè determinare per quali valori di P e L la linea ferroviaria è più efficiente

L'analisi tre è identica all'analisi due con la differenza che si ha uno solo intervallo temporaneo $[0, 384]$ ore. La struttura di guadagno è la stessa dell'analisi due, differenziandosi della definizione dei casi di studio del tool Möbius (study). Si definisce una variabile di performance per ognuno dei tre modelli con la struttura di guadagno uguale alla variabile C , cioè con guadagno a impulso su le attività trackPistoiaFlorence e trackLuccaViareggio.

Performance Variable : SteadyStateFlorence			
Affecting Models	Alternation		
Impulse Functions			
Reward Function	(Reward is over all Available Models) return (Alternation->FlorenceDepartures->Mark() + Alternation->FlorenceArrivals->Mark());		
Simulator Statistics	Type	Steady State	
	Options	Estimate Mean	
		Include Lower Bound on Interval Estimate	
		Include Upper Bound on Interval Estimate	
		Estimate out of Range Probabilities	
		Confidence Level is Relative	
	Parameters	Initial Transient	0.0
		Batch Size	1.0
	Confidence	Confidence Level	0.95
Confidence Interval		0.1	

Performance Variable : SteadyStateLucca			
Affecting Models	Alternation		
Impulse Functions			
Reward Function	(Reward is over all Available Models) return (Alternation->LuccaToViareggio->Mark() + Alternation->LuccaToPistoia->Mark());		
Simulator Statistics	Type	Steady State	
	Options	Estimate Mean	
		Include Lower Bound on Interval Estimate	
		Include Upper Bound on Interval Estimate	
		Estimate out of Range Probabilities	
		Confidence Level is Relative	
	Parameters	Initial Transient	0.0
		Batch Size	1.0
	Confidence	Confidence Level	0.95
Confidence Interval		0.1	

Performance Variable : SteadyStatePistoia			
Affecting Models	Alternation		
Impulse Functions			
Reward Function	(Reward is over all Available Models) return (Alternation->PistoiaToLucca->Mark() + Alternation->PistoiaToFlorence->Mark());		
Simulator Statistics	Type	Steady State	
	Options	Estimate Mean	
		Include Lower Bound on Interval Estimate	
		Include Upper Bound on Interval Estimate	
		Estimate out of Range Probabilities	
		Confidence Level is Relative	
	Parameters	Initial Transient	0.0
		Batch Size	1.0
	Confidence	Confidence Level	0.95
Confidence Interval		0.1	

Performance Variable : SteadyStateViareggio			
Affecting Models	Alternation		
Impulse Functions			
Reward Function	(Reward is over all Available Models) return (Alternation->ViareggioDepartures->Mark() + Alternation->ViareggioArrivals->Mark());		
Simulator Statistics	Type	Steady State	
	Options	Estimate Mean	
		Include Lower Bound on Interval Estimate	
		Include Upper Bound on Interval Estimate	
		Estimate out of Range Probabilities	
		Confidence Level is Relative	
	Parameters	Initial Transient	0.0
		Batch Size	1.0
	Confidence	Confidence Level	0.95
Confidence Interval		0.1	

Figura 1.9: Variabili di performance che determinano lo stato della linea ferroviaria a regime

Sul tool Möbius si ha la seguente definizione per la politica Alternation 1.11.
Stessa definizione per gli modelli atomici PistoiaFirst e LuccaFirst.

1.5.5 Analisi 4: Comportamento del sistema in presenza di guasti

Con l'analisi 4 si cerca di capire l'efficienza della linea ferroviari, numero medio dei treni che terminano la corsa, in presenza di percentuali differenti di guasti considerando istanti differenti di tempo.
Visto che l'insieme degli istanti di tempo e la variabile di performance è identica all'analisi due si fa uso di tale variabile di guadagno. Le differenze sono nei parametri, esattamente nel definire i diversi valori di h . Nella figura 1.10 si descrive tale variabile che è identica ad ognuno dei tre modelli atomici definiti.

1.6 Parametri usati per lo studio

Performance Variable : TravelCompleted		
Affecting Models	Alternation	
Impulse Functions	Alternation->trackPistoiaFlorence	
	(Reward is over all Available Models)	
	return (1.0);	
	Alternation->trackLuccaViareggio	
Reward Function	(Reward is over all Available Models)	
	return (1.0);	
Simulator Statistics	Type	Interval of Time
	Options	Estimate Mean
		Include Lower Bound on Interval Estimate
		Include Upper Bound on Interval Estimate
		Estimate out of Range Probabilities
	Parameters	Confidence Level is Relative
		Start Time
	Confidence	0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
		Stop Time
	1,6,12,24,48,96,192,384,	
	Confidence Level	
	0.95	
	Confidence Interval	
	0.1	

Figura 1.10: Variabile di performance che determina il numero dei treni che terminano la corsa

I tre modelli atomici hanno lo stesso insieme di variabili globali.

Variabili globali della linea ferroviaria		
Nome della variabile	Tipo di valore	Descrizione
N	int	$2 * N$ Numero totale dei treni nella linea ferroviaria
P	int	$P + 1$ Piattaforme presenti nella stazione di Pistoia
L	int	$L + 1$ Piattaforme presenti nella stazione di Lucca
h	double	Probabilità che la locomotiva si ferma

Analisi 1

Il primo studio (study) è utilizzato per l'analisi 1, in Möbius, è descritto nella tabella 1.12:

Analisi 2

Il secondo studio (Analysis2) è identica all'analisi Analysis1 ed è usato per tutte e tre i modelli.

Analisi 3

Il terzo studio, Analysis3 usato per l'analisi 3 si descrive nella figura 1.13.

Per il modello Alternation la forbice dei parametri P e L è più larga per questo si è usato i parametri $L, P \in \{1, 2, 3, 4, 5, 6, 7, 8\}$ e per un numero di treni $N \in \{10, 14, 18, 22\}$.

Per il problema dell'esplosione degli stati, per gli altri modelli si è utilizzato $N \in \{10, 14\}$.

Analisi 4

Per l'analisi 4, con riferimento al modello Alternation, si usa lo studio AlternationAnalysis4 descritto nella figura 1.14.

Performance Variable Model: AlternationAnalysis3		
Top Level Model Information	Child Model Name	Alternation
	Model Type	SAN Model

Performance Variable : TrainCompleted			
Affecting Models	Alternation		
Impulse Functions	RailWay4->trackPistoiaFlorence (Reward is over all Available Models)		
	return (1.0);		
	Alternation->trackPistoiaFlorence (Reward is over all Available Models)		
	return (1.0);		
	Alternation->trackLuccaViareggio (Reward is over all Available Models)		
	return (1.0);		
Reward Function	(Reward is over all Available Models)		
Simulator Statistics	Type	Interval of Time	
	Options	Estimate Mean	
		Include Lower Bound on Interval Estimate	
		Include Upper Bound on Interval Estimate	
		Estimate out of Range Probabilities	
		Confidence Level is Relative	
	Parameters	Start Time	0.0 ,
		Stop Time	384,
	Confidence	Confidence Level	0.95
Confidence Interval		0.1	

Figura 1.11: Variabile di performance che determina il numero dei treni che terminano la corsa per il modello Alternation

Per l'analisi 4, con riferimento al modello PistoiaFirst, si fa uso dello studio PistoiaFirstAnalysis4 descritto nella figura 1.15.

Per l'analisi 4, in riferimento al modello LuccaFirst, si fa uso dello studio LuccaFirstAnalysis4 descritto nella figura 1.16.

1.7 Simulatori di Möbius

Per avere i risultati desiderati si usano le seguenti combinazioni dei passi del programma Möbius.

Range Study Variable Assignments for Study *Analysis1* in Project *Railway* :

Variable	Type	Range Type	Range	Increment	Increment Type	Function	n
L	int	Fixed	6	-	-	-	-
N	int	Manual	[1, 3, 5, 7, 9, 11]	-	-	-	-
P	int	Fixed	4	-	-	-	-
h	double	Fixed	0.001	-	-	-	-

Figura 1.12: Valori dei parametri usati nel primo studio (Analysis1)

Range Study Variable Assignments for Study *Analysis3* in Project *Railway* :

Variable	Type	Range Type	Range	Increment	Increment Type	Function	n
L	int	Incremental	[Incremental Range,1]	1	Additive	-	-
N	int	Fixed	7	-	-	-	-
P	int	Incremental	[Incremental Range,1]	1	Additive	-	-
h	double	Fixed	0.001	-	-	-	-

Figura 1.13: Valori dei parametri usati per l'analisi 3 (Analysis3)

Simulatori di Möbius			
Modello atomico	Reward (Struttura di Guadagno)	Study	Solver
Alternation	AlternationAnalysis1	Analysis1	AlternationAnalysis1Transient (Transient Solver)
	AlternationAnalysis1	Analysis1	AlternationSteady (Iterative Steady State Solver)
	AlternationAnalysis2	Analysis2	AlternationAnalysis2 (Accumulated Reward Solver)
	AlternationAnalysis3	Analysis3	AlternationAnalysis3 (Accumulated Reward Solver)
	AlternationAnalysis2	AlternationAnalysis4	AlternationAnalysis4 (Accumulated Reward Solver)
PistoiaFirst	PistoiaFirstAnalysis1	Analysis1	PistoiaFirstAnalysis1Transient (Transient Solver)
	PistoiaFirstAnalysis1	Analysis1	PistoiaFirstSteady (Iterative Steady State Solver)
	PistoiaFirstAnalysis2	Analysis2	PistoiaFirstAnalysis2 (Accumulated Reward Solver)
	PistoiaFirstAnalysis3	Analysis3	PistoiaFirstAnalysis3 (Accumulated Reward Solver)
	PistoiaFirstAnalysis2	PistoiaFirstAnalysis4	PistoiaFirstAnalysis4 (Accumulated Reward Solver)
LuccaFirst	LuccaFirstAnalysis1	Analysis1	LuccaFirstAnalysis1Transient (Transient Solver)
	LuccaFirstAnalysis1	Analysis1	LuccaFirstSteady (Iterative Steady State Solver)
	LuccaFirstAnalysis2	Analysis2	LuccaFirstAnalysis2 (Accumulated Reward Solver)
	LuccaFirstAnalysis3	Analysis3	LuccaFirstAnalysis3 (Accumulated Reward Solver)
	LuccaFirstAnalysis2	LuccaFirstAnalysis4	LuccaFirstAnalysis4 (Accumulated Reward Solver)

1.8 Risultati ottenuti

Visto che i dati prodotti da le varie simulazioni sono in quantità enormi e la loro elaborazione in uno stato grezzo risulta di difficili comprensione, tramite il programma *Matlab* si è cercato una rappresentazione grafica più espressiva.

1.8.1 Risultati dell'analisi uno: Sistema nell'istante di tempo

Nella simulazione si fa uso dei seguenti parametri:

Range Study Variable Assignments for Study *AlternationAnalysis4* in Project Railway :

Variable	Type	Range Type	Range	Increment	Increment Type	Function	n
L	int	Fixed	6	-	-	-	-
N	int	Fixed	7	-	-	-	-
P	int	Fixed	4	-	-	-	-
h	double	Incremental	[Incremental Range,10.0]	10.0	Multiplicative	-	-

Figura 1.14: Valori dei parametri usati per l'analisi 4 per il modello Alternation

Range Study Variable Assignments for Study *PistoiaFirstAnalysis4* in Project Railway :

Variable	Type	Range Type	Range	Increment	Increment Type	Function	n
L	int	Fixed	8	-	-	-	-
N	int	Fixed	5	-	-	-	-
P	int	Fixed	4	-	-	-	-
h	double	Incremental	[Incremental Range,10.0]	10.0	Multiplicative	-	-

Figura 1.15: Valori dei parametri usati per l'analisi 4 per il modello PistoiaFirst

- Numero dei treni nella linea ferroviaria $N \in \{2, 6, 10, 14, 18, 22\}$.
- Istanti di tempo $t \in \{1, 6, 12, 24, 48, 96, 192, 384\}$.

I parametri $P = 4, L = 6$ sono fissati.

Scegliendo il modello atomico Alternation, si cerca di definire il numero dei treni che è in sosta nel terminale di Firenze al variare di N e negli istanti t di tempo. La figura 1.17 presenta due graffi, uno bidimensionale e l'altro tridimensionale, che descrivono l'evoluzione del numero dei treni a Firenze.

Si nota che per $N = 2$ il comportamento è costante invece al crescere del numero dei treni nella linea ferroviaria si notano le seguenti osservazioni:

- Dall'inizio fino al secondo istante di osservazione, cioè dopo 6 ore dalla marcatura iniziale si ha un decremento dei numeri di treni spiegato dal fatto che i treni cominciano a muoversi verso le altre stazioni, lasciando Firenze.
- Al seguito del decremento iniziale il numero dei treni nel terminale di Firenze diventa stabile nella progressione in tempo del sistema. Infatti negli ultimi istanti d'osservazione tale differenza si appiattisce incredibilmente, infatti il numero dei treni dopo 192 ore e dopo 384 ore è identica.

Nella figura 1.18 si descrive l'evoluzione della sosta dei treni nelle stazioni minori di Pistoia e Lucca e nel terminale di Viareggio.

Da quello che si osserva si distinguono due tipi di comportamenti nella distribuzione dei treni nel modello atomico Alternation;

- Nei terminali Firenze e Viareggio si ha una diminuzione drastica iniziale della quantità dei treni, proseguendo nell'evoluzione del sistema tale quantità si stabilizza.
- Nelle stazioni minori che inizialmente sono vuote con il trascorrere del tempo si riempiono arrivando a sfiorare i limiti delle piattaforme. Infatti in un numero sempre maggiore di treni nel sistema risulta che la sezione due con il binario unico che collega le stazioni di Pistoia e Lucca funge da collo di bottiglia della linea Ferroviaria. Si ha un concentramento di treni in tali stazioni finché si raggiungono i limiti delle piattaforme. Quando si raggiungono tali limiti definiti da P e L i treni cominciano a sostare nei terminali.

Analisi uno nell'istante di tempo per i modelli atomici PistoiaFirst e LuccaFirst

Nella figura 1.19 si descrivono l'andamento dello stato della linea ferroviaria per le politiche PistoiaFirst e LuccaFirst.

Nel modello PistoiaFirst si osservano le seguenti particolarità;

Range Study Variable Assignments for Study *LuccaFirstAnalysis4* in Project *Railway* :

Variable	Type	Range Type	Range	Increment	Increment Type	Function	n
L	int	Fixed	6	-	-	-	-
N	int	Fixed	7	-	-	-	-
P	int	Fixed	4	-	-	-	-
h	double	Incremental	[Incremental Range,10.0]	10.0	Multiplicative	-	-

Figura 1.16: Valori dei parametri usati per l'analisi 4 per il modello LuccaFirst

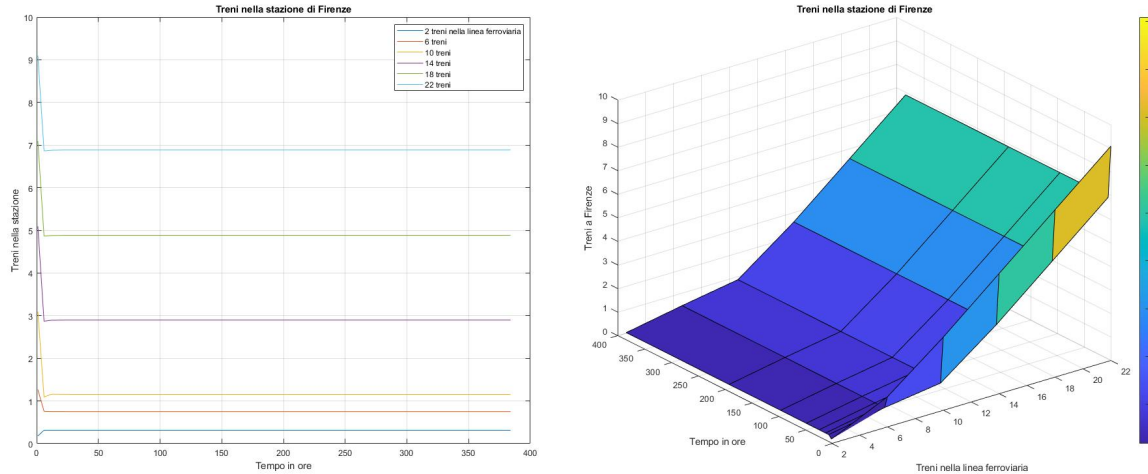


Figura 1.17: Quantità dei treni che aspettano a Firenze con la politica Alternation

- Nel terminale di Firenze inizialmente sono presenti meta dei treni della linea ferroviaria ma con l'evoluzione del sistema questo numero decrementa velocemente per poi stabilizzarsi. In ogni caso rimane sempre in un numero molto inferiore in confronto al terminale di Viareggio.
- Nella stazione di Pistoia il numero dei treni, dopo un scostamento iniziale rimane basso riflettendo il fatto che nella linea ferroviaria viene applicata la politica Pistoia First cioè i treni da Pistoia hanno la precedenza su quelli da Lucca nel utilizzo del binario unico.
- Nella stazione di Lucca il numero dei treni raggiunge il limite delle piattaforme.
- Al terminale di Viareggio il numero dei treni presenti è grande indicando il fatto che una volta che le piattaforme della stazione di Lucca sono occupate i treni attendono a Viareggio il momento di transito.
- Concludendo, nella sezione tre della linea ferroviaria sono presenti la maggior parte dei treni invece nella sezione uno sostano pochi treni.

Nel modello LuccaFirst si ha un comportamento opposto in confronto del modello PistoiaFirst. I treni sono concentrati nella sezione uno e nella sezione tre sono presenti pochi treni. Visto che i treni a Lucca hanno la precedenza in confronto a quelli di Pistoia, in questa stazione sostano pochi treni, stesso risultato per il terminale di Viareggio. Invece nella stazione di Firenze sostano la stragrande parte dei treni presenti nel sistema poiché le piattaforme di Pistoia sono occupate.

Per tutte e tre le politiche dopo le prime 12 ore si nota che il numero dei treni nelle stazioni rimane stabile con dei scostamenti insignificanti questo significa che la linea ferroviaria diventa a regime velocemente.

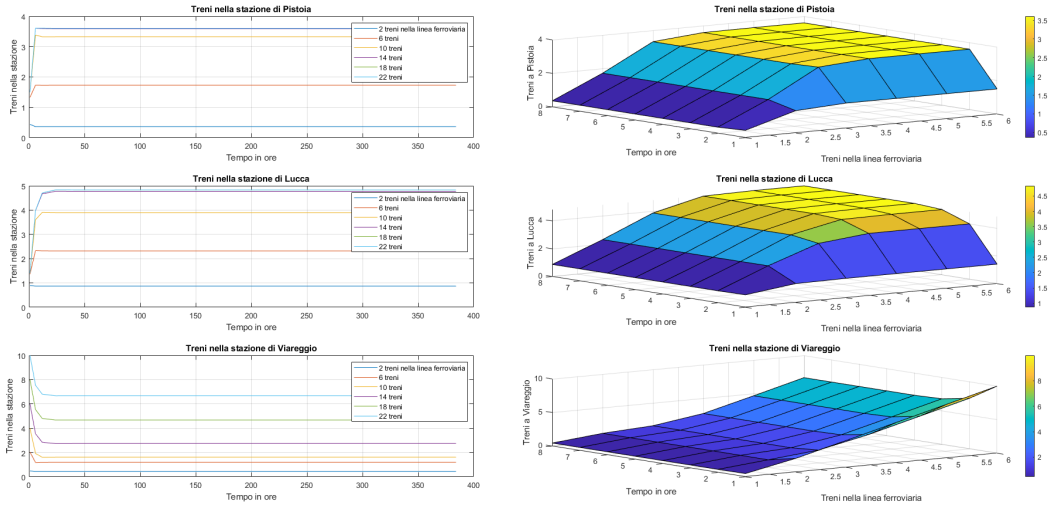


Figura 1.18: Quantità dei treni che aspettano nelle stazioni di Pistoia, Lucca e Viareggio con la politica Alternation

1.8.2 Risultati dell'analisi uno: Sistema a regime

La linea ferroviaria per tutte e tre le politiche non presenta dei stati assorbenti di conseguenza e ben definita il comportamento a regime indipendente dalla marcatura iniziale.

Come si è accennato nell'analisi precedente per tutte e tre i modelli dopo le prime 12 ore di evoluzione il numero dei treni nelle stazioni rimane quasi invariato e con il passare del tempo la differenza si appiattisce ulteriormente. Infatti negli ultimi istanti di osservazione, precisamente dopo 192 e 384 ore la media del numero dei treni presenti nelle stazioni è identico. Questo fatto viene confermato anche dall'analisi della linea ferroviaria a regime (steady state). Infatti i dati della simulazione a regime sono identici ai dati del sistema dopo 192 ore.

Concludendo, la linea ferroviaria ha uguale numero di treni quando sostano su le stazioni con sistema a regime e quando il sistema è regredito dopo 192 o 384 ore.

La linea ferroviaria raggiunge lo stato a regime dopo 192 ore. Questo risultato è valido indipendentemente dalla politica applicata.

1.8.3 Risultati dell'analisi due

Nella simulazione si utilizzano i seguenti parametri:

- Numero dei treni nella linea ferroviaria $N \in \{2, 6, 10, 14, 18, 22\}$.
- Istanti di tempo $t \in \{1, 6, 12, 24, 48, 96, 192, 384\}$.

Purtroppo non si ha la possibilità di presentare lo studio per l'analisi due per gli modelli atomici PistoiaFirst e LuccaFirst con un numero di treni maggiore di 14 poiché si ha una esplosione dei stati per questi modelli il calcolatore non è in grado di proseguire nella simulazione. Per il modello Alternation si riesce a completare la simulazione con i parametri in ingresso.

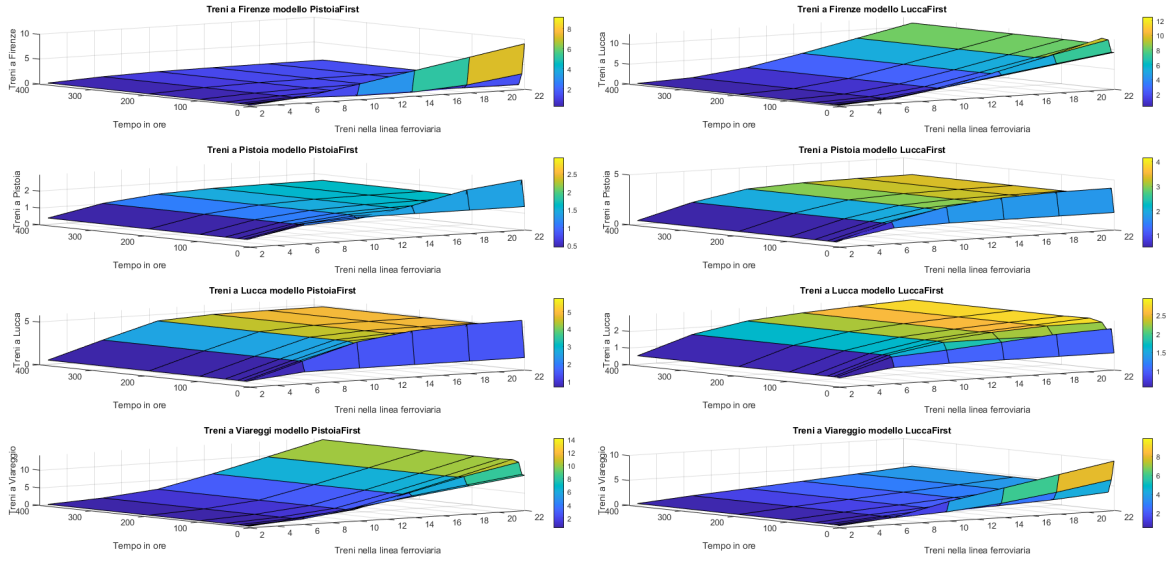


Figura 1.19: Quantità dei treni nelle stazioni per i modelli PistoiaFirst e LuccaFirst

Nella figura 1.20 si descrivono l'andamento della media dei treni che terminano la corsa nella linea ferroviaria con le politiche Alternation, PistoiaFirst e LuccaFirst.

Si nota subito che i tre modelli hanno lo stesso andamento. Per un intervallo di tempo più lungo si ha un numero sempre maggiore di treni che terminano la corsa.

Con pochi treni presenti nel sistema si ha, come c'è si aspetta, un numero piccolo di treni che terminano la corsa. Man mano che il numero dei treni presenti cresce, lo stesso accade anche per il numero dei treni che terminano la corsa.

Quando il linea regredisce nel tempo si nota che anche se nel sistema iniettiamo un numero sempre maggiore di treni questo non comporta una crescita significativa del numero dei treni che terminano la corsa. Anzi la differenza del risultato in presenza di un numero sempre maggiore di treni, si appiattisce. Più precisamente, considerando un numero di treni presenti nel sistema, maggiore di 10, $N \in \{14, 18, 22\}$ il numero medio dei treni che termina la corsa è quasi identico, con differenze insignificanti. Addirittura nel modello PistoiaFirst terminano la corse più treni quando nel sistema sono presenti 10 treni invece che 14, portando a un decremento del servizio.

Concludendo, nella linea ferroviaria per tutte e tre i modelli avere un numero maggiore di 14 di treni presenti nella stazione non migliora le prestazioni del servizio, anzi porta ridondanza.

Per un confronto fra le tre politiche della linea ferroviaria nel intervallo massimamente temporaneo $[0, 384]$ ore si ha le seguenti considerazioni:

- Per un $N \in \{2, 6\}$ numero di treni presenti sul sistema, il modello PistoiaFirst è il più efficiente, cioè ha una media più alta di treni che terminano la corsa.
- Per un $N \in \{10, 14, 18, 22\}$ numeri di treni presenti sul sistema, il modello Alternation è il più efficiente seguito dal modello LuccaFirst.

Passare da 10 a 14 treni in servizio su i modelli Alternation e LuccaFirst si ha una crescita meno di 3% il numero medio dei treni che terminano la corsa, invece su PistoiaFirst si ha un decremento.

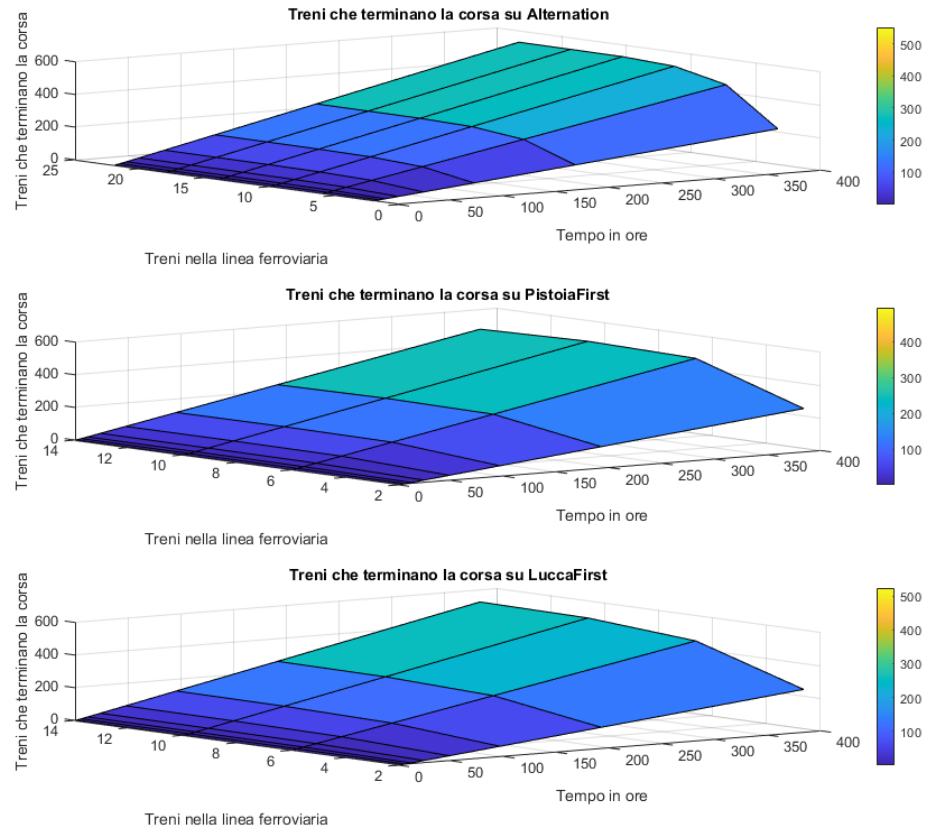


Figura 1.20: Quantità dei treni che terminano la corsa per i modelli Alternation, PistoiaFirst e LuccaFirst

Passare da 6 a 10 treni in servizio su i modelli Alternation e LuccaFirst fa crescere di quasi 10% il numero medio dei treni che terminano la corsa e di 2% per PistoiaFirst.

1.8.4 Risultati dell'analisi tre

Si cerca di determinare il minimo dei valori P e L che massimizzano l'efficienza della linea ferroviaria. Usando $P, L \in \{1, 2, 3, 4, 5, 6, 7, 8\}$ si determina il numero medio dei treni che terminano la corsa. Prima di procedere con l'analisi si cerca di capire qual è il numero N ideale di treni presenti nella ferrovia per una determinata politica.

Prendendo come modello di studio Alternation si vuole determinare l'efficienza della linea ferroviaria per un numero di treni nella linea ferroviaria $N \in \{10, 14, 18, 22\}$ al variare di P e L .

Dalla simulazione si ha i risultati raggruppati nella figura 1.21 da qui si possono fare le seguenti constatazioni.

- I grafici sono molto simili e il fatto di avere più treni nella linea ferroviaria comporta un miglioramento trascurabile per quanto riguarda efficienza.
- Le differenze fra $N = 14$ e $N = \{18, 22\}$ sono insignificanti, di fatto si può scartare la scelta di $N = \{18, 22\}$.

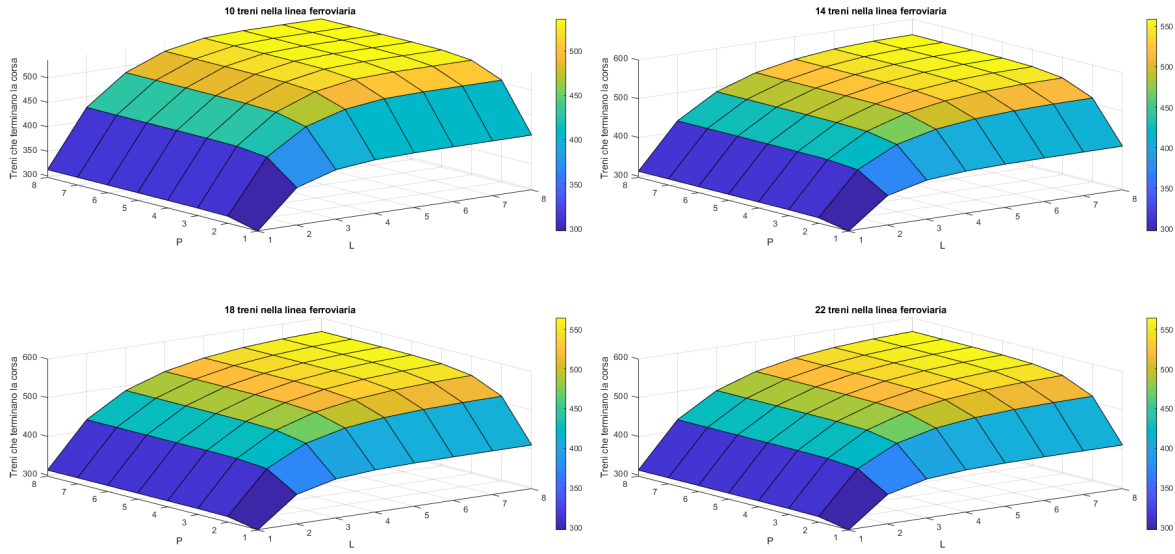


Figura 1.21: Quantità dei treni che terminano la corsa per i modello Alternation con rispettivamente 10, 14, 18, 22 treni presenti nella linea ferroviaria

- Nell'analisi viene chiesto di massimizzare l'efficienza della linea, allora la scelta di $N = 14$ di treni presenti è la scelta migliore in confronto a $N = 10$.
- Per $N = 14$ i valori di $P = 4$ e $L = 6$ sono i valori minimi che massimizzano l'efficienza. Infatti si vede che per $P = 4$ e $L = 6$ sono le coordinate d'inizio del tetto nel graffo e in tale punto si ha il valore 550,039.

Concludendo, per il modello Alternation i parametri che massimizza l'efficienza del sistema sono $N = 14$, $P = 4$, $L = 6$.

Per determinare i parametri degli modelli PistoiaFirst e LuccaFirst si fa uso della simulazione con i parametri $N = 10$ con $L, P \in \{1, 2, 3, 4, 5, 6, 7, 8\}$ e per $N = 14$ con $L, P \in \{1, 2, 3, 4, 5, 6\}$. Questo cambiamento di parametri è dovuto anche alla potenza di calcolo che purtroppo l'elaboratore non riesce ad affrontare. Il risultato dell'analisi tre con questi parametri è rappresentato nella figura 1.22.

Per il modello PistoiaFirst si nota, come accennato nell'analisi due, che si ha una maggior performance della linea ferroviaria quando sono presenti 10 treni invece di 14, per tale modello, un numero maggiore di treni porta a ridondanze nel sistema e per questo che le prestazioni calano. Come risultato i parametri che massimizzano l'efficienza sono $N = 10$, $P = 5$, $L = 8$ e in tale punto si ha il risultato 534.111.

Per il modello LuccaFirst si ha un comportamento simile al modello Alternation, e per un numero di treni presenti nella linea ferroviaria di $N = 14$ con parametri $P = 4$, $L = 6$ si ha una media maggiore di treni che terminano la corsa. In tale punto si ha il risultato 523.1821

Confrontando i tre modelli atomici con i parametri scelti si deduce che il modello Alternation massimizza il numero medio dei treni che terminano la corsa seguito da PistoiaFirst e infine LuccaFirst. Con i parametri $P = 4$, $L = 6$ LuccaFirst ha una performance migliore di PistoiaFirst.

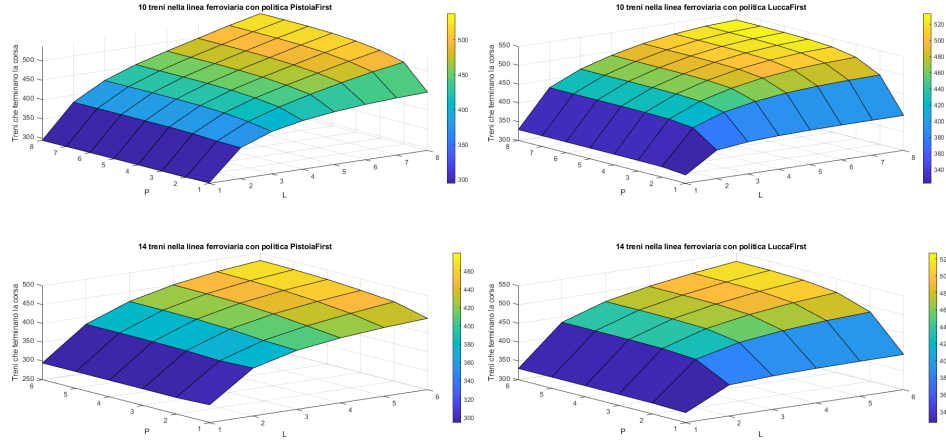


Figura 1.22: Quantità dei treni che terminano la corsa per i modelli PistoiaFirst e LuccaFirst con rispettivamente 10 e 14 treni nella linea ferroviaria

1.8.5 Risultati dell'analisi quattro

Nella figura 1.23 con i parametri tempo $t \in \{1, 6, 12, 24, 48, 96, 192, 384\}$ e $h \in \{0.00001, 0.0001, 0.001, 0.01, 0.1\}$ si determina l'andamento dell'efficienza della linea ferroviaria.

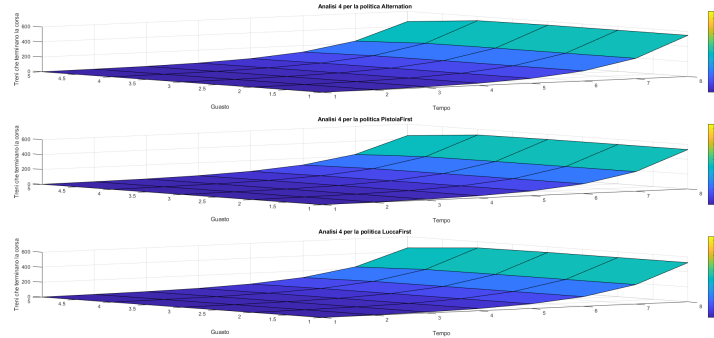


Figura 1.23: Quantità dei treni che terminano la corsa per i modelli Alternation, PistoiaFirst e LuccaFirst nel analisi quattro

Si osserva che

- I tre modelli hanno lo stesso comportamento. Inoltre più il sistema evolve nel tempo, più grande è il numero dei treni che terminano la corsa.
- Al cresce della probabilità di guasto della locomotiva si ha una diminuzione del numero dei treni che terminano la corsa.
- I valori $h \in \{0.00001, 0.0001, 0.001\}$ influenzano in modo insignificante l'efficienza della linea ferroviaria.

- Per un $h \in \{0.1\}$ l'efficienza del sistema diminuisce di 17% in confronto al andamento normale per i modelli atomici Alternation e PistoiaFirst.
- Per un $h \in \{0.1\}$ l'efficienza del sistema diminuisce di 15% in confronto al andamento normale per i modello atomico LuccaFirst infatti in questo caso si ha un numero di treni che terminano la corsa più grande di PistoiaFirst. Concludendo in presenza di importanti guasti della locomotiva la politica LuccaFirst offre più resistenza nel preservare l'efficienza della linea ferroviaria.

1.9 Conclusioni

Questo progetto di modellazione e simulazione si propone come obbiettivo quello di analizzare la performance della linea ferroviaria che collega le città di Firenze e Viareggio. Considerando tre tipi di politiche differenti di priorità nel transito dei treni si cerca di capire lo stato e l'efficienza della linea. In particolare si ha interesse di definire lo stato della linea, la quantità dei treni che sostano su le stazioni, in modo tale da capire quali siano i punti critici del sistema. Si è arrivato al risultato che la linea ferroviaria raggiunge velocemente lo stato a regime (steady state).

La conoscenza dello stato della linea sarà utile per capire quale delle tre politiche è più performante, precisamente si cerca di trovare su quali condizioni della linea ferroviari si ha il maggior numero di transito dei treni che collegano Firenze e Viareggio. Da i risultati si è concluso che la politica di alterazione dei treni che collega Pistoia e Lucca è la migliore.

Inoltre si è interessati a determinare il numero delle piattaforme ideali per una migliore performance concludendo che per Pistoia e Lucca sono rispettivamente i parametri sono $P = 4$ e $L = 6$. Si è analizzato il fatto che l'introduzione di un numero maggiore di 14 di treni comporta solo della ridondanza nel sistema. Inoltre si è scoperto che, per la politica Pistoia First, introdurre una grande quantità di treni nella linea ferroviaria decresce l'efficienza.

Introducendo un numero importante di guasti nella sezione due si è determinato che l'efficienza dell'intera linea diminuisce, inoltre, si è scoperto che con la politica Lucca First si ha una resistenza maggiore nel deterioramento della performance.

Concludendo che se alla linea ferroviaria Firenze - Viareggio viene applicata la politica di alterazione dei treni nella sezione due con parametri di $P = 4$ e $L = 6$ per le stazioni di Pistoia e Lucca con un numero $N = 14$ di treni, si avrà la migliore performance, numero medio dei treni che terminano la corsa.

Elenco delle figure

1.1	Rappresentazione della linea ferroviaria con la politica PistoiaFirst	2
1.2	Rappresentazione della linea ferroviaria con la politica Alternation	3
1.3	Rappresentazione della sezione 1	4
1.4	Rappresentazione della sezione 2 per il modello Alternation	4
1.5	Rappresentazione della sezione 2 per i modelli PistoiaFirst e LuccaFirst	5
1.6	Rappresentazione della sezione 3	6
1.7	Struttura di guadagno per definire il numero dei treni nella stazione di Firenze	10
1.8	Struttura di guadagno per definire il numero dei treni nelle stazione di Pistoia, Lucca e Viareggio	11
1.9	Variabili di performance che determinano lo stato della linea ferroviaria a regime	12
1.10	Variabile di performance che determina il numero dei treni che terminano la corsa	13
1.11	Variabile di performance che determina il numero dei treni che terminano la corsa per il modello Alternation	14
1.12	Valori dei parametri usati nel primo studio (Analysis1)	15
1.13	Valori dei parametri usati per l'analisi 3 (Analysis3)	15
1.14	Valori dei parametri usati per l'analisi 4 per il modello Alternation	16
1.15	Valori dei parametri usati per l'analisi 4 per il modello PistoiaFirst	16
1.16	Valori dei parametri usati per l'analisi 4 per il modello LuccaFirst	17
1.17	Quantità dei treni che aspettano a Firenze con la politica Alternation	17
1.18	Quantità dei treni che aspettano nelle stazioni di Pistoia, Lucca e Viareggio con la politica Alternation	18
1.19	Quantità dei treni nelle stazioni per i modelli PistoiaFirst e LuccaFirst	19
1.20	Quantità dei treni che terminano la corsa per i modelli Alternation, PistoiaFirst e LuccaFirst	20
1.21	Quantità dei treni che terminano la corsa per i modello Alternation con rispettivamente 10, 14, 18, 22 treni presenti nella linea ferroviaria	21
1.22	Quantità dei treni che terminano la corsa per i modelli PistoiaFirst e LuccaFirst con rispettivamente 10 e 14 treni nella linea ferroviaria	22
1.23	Quantità dei treni che terminano la corsa per i modelli Alternation, PistoiaFirst e LuccaFirst nel analisi quattro	22