

Matteo Pallagrosi – 0338469

Relazione progetto PMCSN

Studio di simulazione di una
stazione della metropolitana di Roma

Relazione progetto PMCSN - Matteo Pallagrosi

Sommario

1. Descrizione del sistema
2. Problemi del sistema
3. Obiettivi
4. Modello concettuale
 - 4.1. Descrizione dei centri
 - 4.2. Stato
 - 4.3. Eventi
5. Modello delle specifiche
 - 5.1. Definizione delle probabilità di routing
 - 5.2. Modellazione dei centri
 - 5.3. Tassi di arrivo e fasce orari
6. Modello computazionale
 - 6.1. Generazione dei processi stocastici
7. Verifica
8. Validazione
9. Progettazione degli esperimenti
 - 9.1. Analisi del collo di bottiglia
 - 9.2. Analisi del transitorio
 - 9.3. Simulazione a orizzonte infinito
 - 9.4. Simulazione a orizzonte finito
 - 9.5. Conclusioni

1. Descrizione del sistema

Il sistema oggetto dell'analisi è quello di una stazione della metropolitana, nello specifico della città di Roma, dove i servizi di mobilità pubblica sono gestiti da ATAC. Queste stazioni, in modo particolare quelle situate nelle zone centrali e presso i capolinea, sono spesso caratterizzate da un'affluenza significativa, specialmente nelle fasce orarie caratterizzate da maggiori spostamenti da parte dei cittadini, ad esempio negli orari in cui si recano o escono da lavoro, così come a ridosso dell'ingresso e in corrispondenza dell'uscita da scuola da parte degli studenti. La scelta di considerare le stazioni metropolitane di Roma, piuttosto che di altre città, è dipendente dalla disponibilità di dati precisi sull'affluenza (biglietti vidimati giornalmente e abbonamenti sottoscritti), secondo quanto contenuto nel report annuale sui dati della mobilità pubblicato da ATAC e reperibile online (1), nonché dettato dall'esperienza personale che ha permesso di modellare al meglio il sistema preso in esame. Nello specifico si è scelto di prendere in considerazione la stazione di San Giovanni della linea A, poiché risulta tra quelle a maggiore affluenza secondo quanto riportato in (1).

I fruitori del servizio hanno la possibilità di acquistare un biglietto presso le biglietterie automatiche collocate all'ingresso delle stazioni stesse, oppure presso esercenti fisici, come tabaccherie o appositi desk sempre collocati dentro le stazioni. Una volta muniti di biglietto, oppure di un abbonamento valido (che evita loro di volta in volta le attese presso le biglietterie avendo validità annuale), possono recarsi presso i tornelli, dove avviene la vidimazione del titolo di viaggio in loro possesso. Come gli abbonati, anche coloro che desiderano pagare con carta di credito, possono recarsi direttamente ai tornelli senza passare per le biglietterie in quanto tale metodo di pagamento è messo a disposizione grazie ad un lettore presente su ogni tornello. Prima di raggiungere la banchina gli utenti potrebbero essere sottoposti ad ulteriori controlli (di validità dell'abbonamento o del biglietto) da parte di appositi ispettori. Anche se non sempre presenti, nella modellazione del sistema si è scelto di considerare la presenza dei controllori, figure che risultano sempre più necessarie dato l'aumento nell'ultimo periodo di persone che superano i tornelli senza essere muniti di un titolo valido. Lo studio presentato vuole quindi anche valutare se l'effettiva presenza costante di controllori possa essere possibile, senza avere un impatto negativo troppo ingente sui tempi di attesi sperimentati dai fruitori del servizio di trasporti. Infine per coloro che ne hanno necessità, è possibile raggiungere la banchina tramite appositi ascensori.

2. Problematiche del sistema

Poiché la metropolitana rappresenta uno dei principali mezzi di spostamento utilizzati a Roma, questa risulta spesso caratterizzata da lunghe attese, che si verificano in modo particolare nelle stazioni situate nel centro cittadino, maggiormente affollate, con code che tipicamente riguardano le biglietterie (per lo più utilizzate da turisti che acquistano titoli di viaggio per singola corsa, e che talvolta faticano nel destreggiarsi con l'interfaccia poco intuitiva che caratterizza le biglietterie automatiche), e che potrebbero quindi scoraggiare i nuovi utenti arrivati dall'utilizzo del servizio, portandoli ad abbandonare la stazione e utilizzare differenti mezzi di trasporto come ad esempio il taxi. Inoltre nelle fasce orarie caratterizzate da maggiore affluenza, l'arrivo degli utenti presso i tornelli potrebbe condurre rapidamente alla saturazione delle capacità di tale centro, comportando la

creazione di lunghe code e l'insoddisfazione dei cittadini. Come già indicato in precedenza, al fine di ridurre l'incrementarsi di un fenomeno sempre più presente, ossia il transito presso il tornello senza validare alcun biglietto o abbonamento, ma passando rapidamente alle spalle di un ignaro utente quando le porte dei tornelli sono ancora aperte, si è valutata la possibilità di intensificare le attività di controllo del biglietto da parte di appositi ispettori, che ovviamente conseguono in un aumento dei tempi necessari agli utenti per raggiungere la banchina, ma, limitando l'aumento di questi tempi con un adeguato dimensionamento del numero di addetti al controllo, ciò può condurre ad una forte limitazione di tale fenomeno del "salto del tornello". Infine a supporto di coloro che necessitano di un ausilio per raggiungere la banchina (per motivi di comodità o perché persone caratterizzate da una mobilità ridotta), la disponibilità degli ascensori risulta un altro aspetto critico da tenere in considerazione.

3. Obiettivi

Alla luce delle problematiche evidenziate nella sezione precedente, lo studio condotto è stato finalizzato al conseguimento dei seguenti obiettivi:

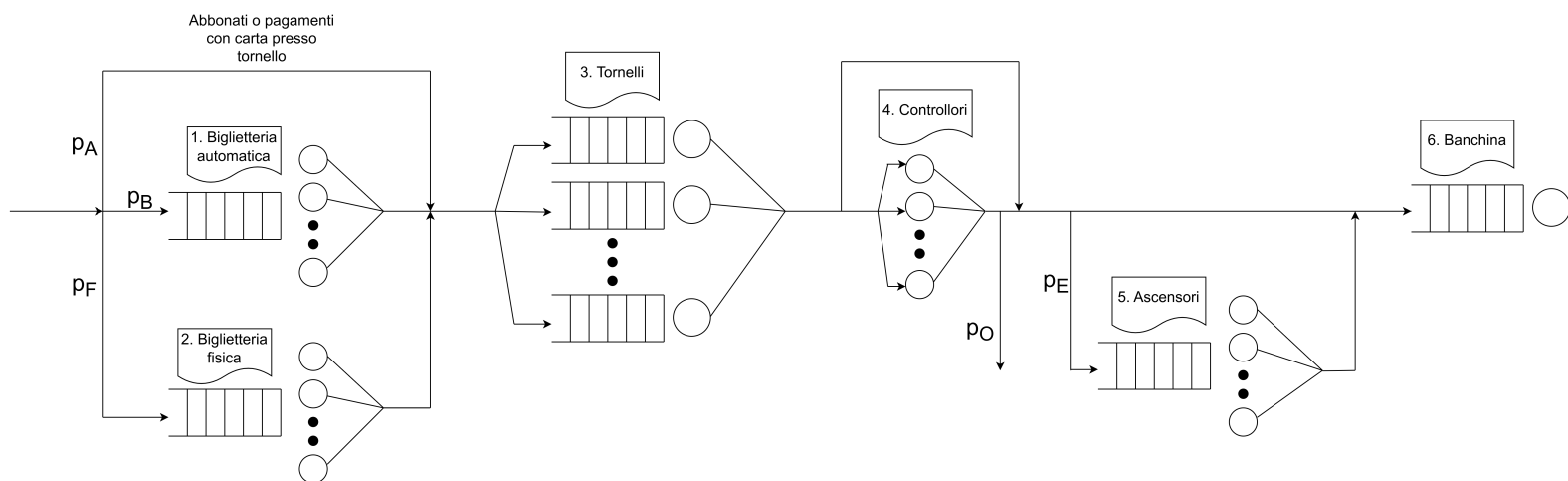
- Il tempo medio affinché un qualunque utente raggiunga la banchina deve essere inferiore a 6 minuti.
- La percentuale di persone che viene sottoposta ad ulteriori controlli da parte degli appositi controllori deve essere almeno pari al 30%. Alla base di questo obiettivo vi è la considerazione secondo cui è necessario evitare che la pressione sul centro controllori sia troppo elevata (dato da un tasso di arrivo presso questo centro troppo alto) altrimenti molte persone riescono a evitare questi controlli, situazione che si vuole limitare il più possibile.
- Il tempo medio di attesa in coda per le persone che vogliono usufruire degli ascensori per raggiungere la banchina deve essere inferiore a 30 secondi.
- Il tempo che un utente impiega per acquistare un biglietto dalla cassa automatica (dal momento in cui si mette in coda) deve essere inferiore a 5 minuti.

Il conseguimento di tali obiettivi deve tenere in conto del fatto che la banchina presenta una capacità massima prevista di 880 passeggeri, di cui si deve evitare il superamento per evitare il sovraffollamento, che potrebbe mettere a rischio la sicurezza dei fruitori del servizio. Tale capacità è stata definita facendo riferimento alla normativa UNI 7508, la cui sintesi è riportata in (2), per il dimensionamento delle banchine delle stazioni della metropolitana, considerando un rapporto tra passeggeri e area disponibile al più pari a $N/A = 2$, e considerando le dimensioni medie di una banchina della metro di Roma, pari a 110 x 4 m. Lo studio propone quindi di simulare anche il passaggio dei treni, modellato secondo un processo stocastico opportuno, con una media coerente con il tempo medio di interarrivo dei treni secondo quanto riportato in (3), ossia 3 minuti nelle ore di punta. Essendo stata presa in considerazione una stazione intermedia, quale San Giovanni, il treno raggiunge tipicamente la stazione con un certo numero di passeggeri già a bordo, che sarà anch'esso modellato in modo opportuno. Tali considerazioni pongono

quindi dei vincoli alla configurazione ottimale dei vari centri della stazione che lo studio persegue, poiché con l'obiettivo di raggiungere i QoS fissati, è necessario evitare di ridurre i tempi eccessivamente, poiché ciò potrebbe produrre un'affluenza alla banchina che il transito dei treni non riesce a smaltire adeguatamente. Il processo di transito dei treni viene quindi modellato in modo opportuno, ma non si prevede una sua manipolazione (ossia non si incrementa la frequenza di arrivo dei treni nel tentativo di smaltire il traffico poiché ciò renderebbe il modello irrealistico). Nel rispettare i Qos, l'obiettivo è quello di minimizzare il numero di serventi utilizzati in ogni configurazione, così da ridurre i costi previsti per l'operatività della stazione.

4. Modello concettuale

Il diagramma del sistema considerato risulta il seguente:



Gli utenti in arrivo presso la stazione, ciascuno modellato come un job nella simulazione, possono decidere se acquistare un biglietto presso le biglietterie elettroniche oppure presso le casse fisiche. La probabilità con cui è stato modellato questo comportamento sono state definite sulla base dell'esperienza personale, secondo cui si è preferito favorire l'affluenza di utenti presso le biglietterie elettroniche, maggiormente utilizzate rispetto alla casse fisiche. Coloro che possiedono un abbonamento valido oppure che intendono pagare tramite carta di credito possono invece recarsi direttamente ai tornelli.

Per il centro dei tornelli si è scelto di non modellare una probabilità di uscita, poiché tipicamente chi si reca presso il tornello è munito di un titolo di viaggio valido, oppure è intenzionato ad attraversare il tornello in modo illecito, rendendo quindi irrealistica la possibilità di un utente che abbandona il sistema in corrispondenza di questo centro.

Il centro successivo, caratterizzato dalla presenza dei controllori, prevede il controllo del possesso di un titolo di viaggio o abbonamento in corso di validità. Qualora tale validità non dovesse essere riscontrata l'utente viene multato e deve abbandonare la stazione. Da qui la necessità di modellare una probabilità di uscita p_o superati questi controlli.

Gli utenti possono poi recarsi direttamente presso la banchina oppure usufruire degli ascensori messi a disposizione nella stazione.

La banchina non rappresenta un “centro” nel senso stretto del termine, sebbene sia stato disegnato come tale per semplicità di rappresentazione, e poichè comunque rappresenta un processo stocastico, quello dell’arrivo del treno, che preleva un certo numero di utenti in attesa sulla banchina. Tale centro fittizio non sarà quindi oggetto di manipolazioni nella ricerca della configurazione ottimale al fine di perseguire i QoS, ma rappresenta un vincolo (l’obbligo di non superare la capacità massima della banchina) di cui tenere in conto nella configurazione del sistema. L’inserimento di questo vincolo è stato reso necessario al fine di rendere il sistema modellato il più realistico possibile, rendendo quindi lo studio di simulazione significativo ed evitando che un dimensionamento esagerato nel numero di serventi presso ciascun centro potesse rappresentare la soluzione finale, dato che questo comporterebbe un afflusso di utenti sulla banchina non sostenibile dalla disponibilità di treni e dalla frequenza con cui essi transitano presso la stazione e prelevano passeggeri.

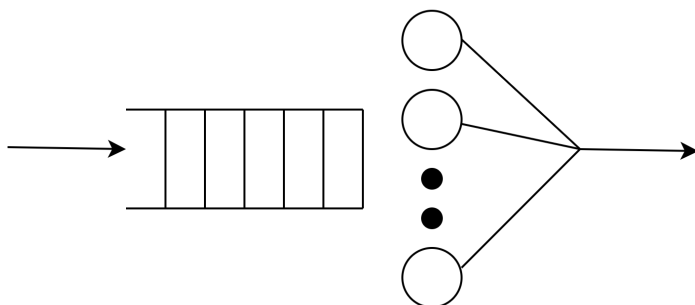
Sono definite le seguenti probabilità:

- $p_A \rightarrow$ probabilità di avere un utente abbonato, che non deve comprare il biglietto, ma si reca direttamente ai tornelli.
- $p_B \rightarrow$ utenti che acquistano il biglietto presso le casse automatiche
- $p_F \rightarrow$ utenti che acquistano il biglietto presso le casse fisiche
- $p_E \rightarrow$ utenti che utilizzano gli ascensori per raggiungere la banchina
- $p_O \rightarrow$ probabilità di non superare i controlli da parte degli appositi ispettori, e di uscire quindi dal sistema

4.1 Descrizione dei centri

Il sistema presenta tre tipologie di centri, qui descritte.

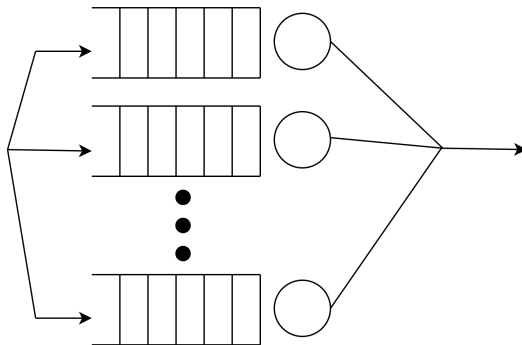
(1) Multi-server single-queue (Mssq):



questa tipologia di centri prevede una coda singola, in cui vengono accodati gli utenti in arrivo presso il centro. Se al momento dell’arrivo dell’utente almeno uno dei server è libero, l’utente può essere immediatamente servito, altrimenti l’utente viene messo in coda secondo uno scheduling di tipo FIFO. Nel momento in cui un utente completa presso un server, se è presente

almeno un utente in coda questo sarà servito, altrimenti il server torna ad essere libero. Una volta serviti gli utenti lasciano il centro e raggiungono il centro successivo. Appartengono a questa tipologia i centri 1, 2 e 5.

(2) Multi-server multi-queue (Msmq):



questa tipologia di centri prevede code multiple, ciascuna delle quali presenta un server che processa i job prelevati dalla rispettiva coda. Tale sistema risulta simile ad un meccanismo di frequency-division multiplexing, ma con la fondamentale differenza per cui la ripartizione dei job in arrivo al centro tra le varie code non risulta essere uniforme, bensì basata sullo stato corrente delle code stesse, il che risulta in una probabilità di routing verso le diverse coppie code-server che varia dinamicamente in funzione della popolazione correntemente presente in ogni coda. Da ciò deriva la reale necessità dello studio di simulazione, poiché la presenza di un routing probabilistico dipendente dallo stato impedisce di modellare il sistema mediante una catena di Markov, e di risolverlo quindi analiticamente nella sua interezza. Il meccanismo di accodamento è qui spiegato nel dettaglio:

- se un utente arriva presso il centro e trova una coda vuota (il che implica che anche il server deve essere vuoto), si dirige verso quella coppia coda-server, ed entra immediatamente in servizio. Se più di una coppia coda-server risulta vuota, l'utente seleziona uniformemente una tra queste, e si dirige verso tale coppia.
- Se nessuna delle coppie code-server risulta vuota, l'utente identifica quale tra le code presenta il minor numero di utenti accodati, e si accoda a questi. Se più code presentano lo stesso numero di utenti, il nuovo utente seleziona uniformemente una tra queste.

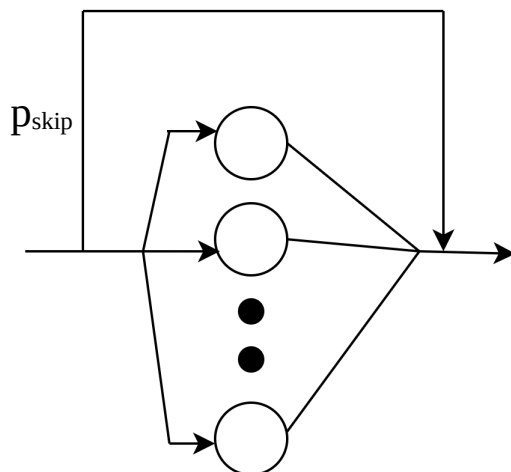
Gli utenti vengono serviti dal server di ogni coda secondo un ordinamento FIFO. Una volta serviti lasciano il centro per raggiungere il centro successivo. Appartiene a questa tipologia il centro 3.

La modellazione di questo centro così strutturato è stata dettata dalla volontà di rispecchiare il più possibile un comportamento reale messo in atto dagli utenti che si dirigono verso i tornelli, ossia quello di scegliere se possibile un tornello libero. Qualore questo non fosse disponibile, scelgono di accodarsi dietro ad uno dei tornelli con la coda più corta.

Questo centro è di fatto costituito da diversi centri single-server single-queue. La scelta di modellare questi singoli centri insieme, piuttosto che individualmente, è dettata dalla volontà di studiarne il comportamento nella

loro totalità, portando a variare il numero di code coppie-server al fine di raggiungere i QoS fissati.

(3) Multi-server loss system (Msls):



questa tipologia di centri prevede molteplici server, ma non dispone di alcuna coda. Un utente in arrivo presso tale centro potrà quindi essere servito se almeno un server risulta libero, altrimenti l'utente viene "scartato", ossia supera il centro senza attraversarlo. Quando un utente completa il servizio, il server corrispondente torna ad essere libero.

La probabilità di skip dipende quindi dallo stato del centro, poiché in ogni istante di tempo potrà essere nulla se almeno un servente è libero, altrimenti è pari a 1.

A questa tipologia appartiene il centro 4. Tale centro è stato modellato in questo modo per riflettere un comportamento reale che si sperimenta quando gli utenti si accingono ad effettuare il controllo del biglietto o dell'abbonamento. Se al momento del transito dell'utente uno dei controllori è libero (ossia non impegnato ad effettuare alcun controllo), l'utente verrà controllato, altrimenti, se tutti i controllori sono già impegnati, l'utente può superare i controlli senza che venga verificato il possesso di un titolo di viaggio valido. L'inserimento di una coda non rispecchierebbe quindi la realtà, e per tale motivo si è deciso di evitarla.

4.2 Stato

In ogni istante lo stato del sistema è dato dallo stato di ciascun centro. In ogni centro ogni coda può essere vuota o non vuota (nei centri che possiedono code), e ogni server può essere libero oppure occupato. Nei centri con coda (o code multiple), un server può essere libero solo se non ci sono job in coda. Se il server è libero allora la coda deve essere vuota. Se la coda è non vuota, il server deve essere occupato.

4.3 Eventi

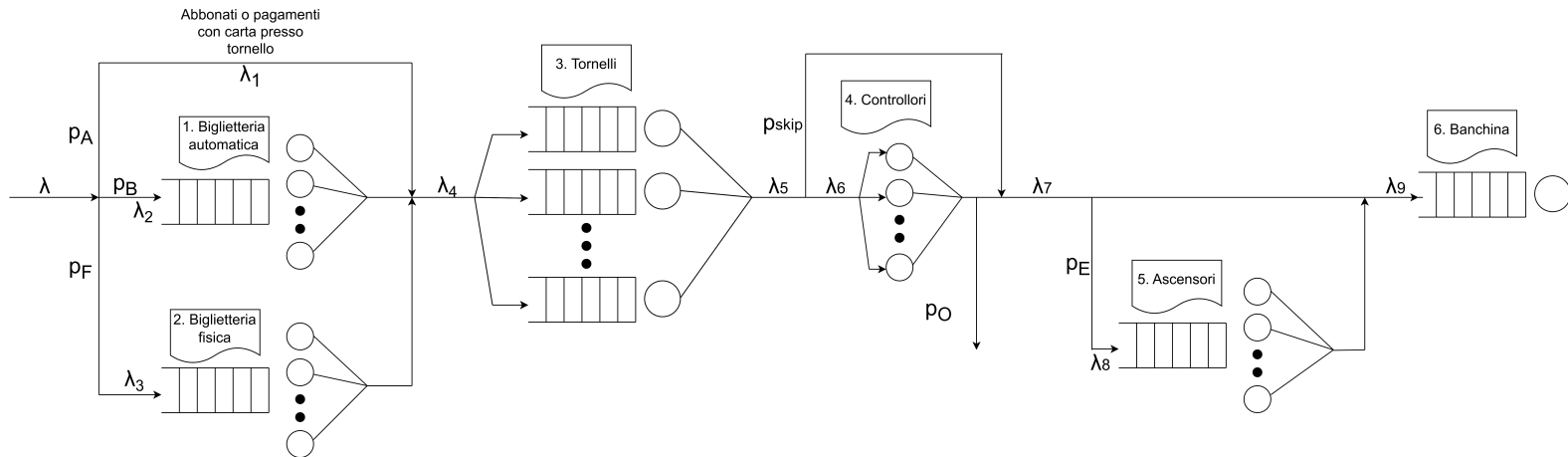
Si definiscono le seguenti tipologie di eventi:

- arrivo di un job
- completamento del servizio di un job

- cambiamento di fascia oraria (che implica un cambiamento del tasso di arrivo dei job dall'esterno)
- campionamento (quindi raccolta delle statistiche)

In particolare le ultime due tipologie rappresentano eventi “artificiali”, schedulati in istanti di tempo prefissati.

5. Modello delle specifiche



$$\lambda_1 = p_A \cdot \lambda$$

$$\lambda_2 = p_B \cdot \lambda$$

$$\lambda_3 = p_F \cdot \lambda$$

$$\lambda_4 = \lambda_1 + \lambda_2 + \lambda_3 = (p_A + p_B + p_F) \cdot \lambda = \lambda$$

$$\lambda_5 = \lambda_4 = \lambda$$

$$\lambda_6 = \lambda_5 \cdot (1 - p_{\text{skip}}) = \lambda \cdot (1 - p_{\text{skip}})$$

$$\lambda_7 = [\lambda_6 \cdot (1 - p_O)] + (\lambda_5 \cdot p_{\text{skip}}) = [\lambda \cdot (1 - p_{\text{skip}}) \cdot (1 - p_O)] + (\lambda \cdot p_{\text{skip}})$$

$$\lambda_8 = p_E \cdot \lambda_7$$

$$\lambda_9 = \lambda_8 + (1 - p_E) \cdot \lambda_7 = p_E \cdot \lambda_7 + (1 - p_E) \cdot \lambda_7 = \lambda_7$$

La matrice che descrive il routing nel sistema è la seguente:

	Est	1	2	3	4	5	6
Est	0	p_B	p_F	p_A	0	0	0
1	0	0	0	1	0	0	0
2	0	0	0	1	0	0	0
3	0	0	0	0	$1 - p_{\text{skip}}$	$(p_{\text{skip}} \cdot p_E)$	$p_{\text{skip}} \cdot (1 - p_E)$
4	p_O	0	0	0	0	$(1 - p_O) \cdot p_E$	$(1 - p_O) \cdot (1 - p_E)$
5	0	0	0	0	0	0	1
6	1	0	0	0	0	0	0

EST = esterno, 1 = biglietteria automatica, 2 = biglietteria fisica, 3 = tornelli, 4 = controllori, 5 = ascensori, 6 = banchina

L'elemento (i,j) della matrice rappresenta la probabilità di andare dal centro i al centro j . Si sottolinea ancora una volta come la banchina rappresenta un centro "fittizio" e di fatto il raggiungimento di essa determina l'uscita dal sistema, a seguito del suo completo attraversamento. Inoltre coerentemente con la volontà di modellare il centro 3 come un unico centro (suddiviso poi in diversi sottocentri singola coda singolo servente), la probabilità di routing verso questo centro si scompone in realtà nella somma di n probabilità, dove n è il numero di coppie server-coda considerate nella configurazione scelta, e ciascuna di queste probabilità, ad ogni istante, dipende dallo stato (numero di job) nella coda corrispondente. Nelle figure successive sono riportati degli esempi che mostrano come vengono modellate queste probabilità in un certo istante di tempo, secondo la politica di accodamento indicata precedentemente, e nel caso di 4 coppie server-coda.

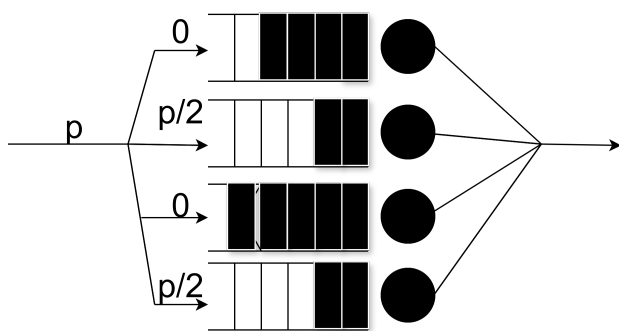


Figura 1: Nessuno dei server è libero. Probabilità distribuita uniformemente tra i server con numero minimo di utenti in coda

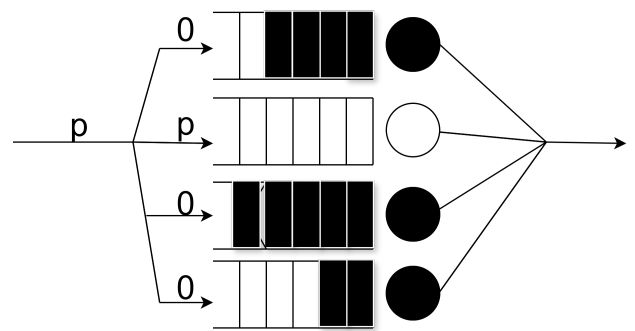


Figura 2: Un server è libero. L'utente si dirige verso di esso

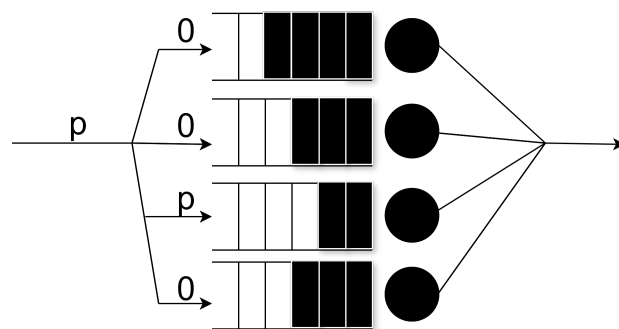


Figura 3: Nessun server risulta libero. L'utente si dirige verso la coda con numero minimo di utenti accodati.

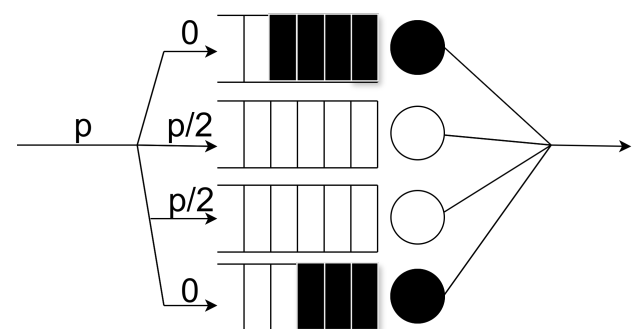


Figura 4 Più di un server risulta libero. Probabilità di routing distribuita uniformemente tra i server liberi.

5.1 Definizione delle probabilità di routing

In (1) sono contenuti i dati riguardanti il numero di abbonati annuali della metropolitana di Roma, che risultano essere (sommando coloro che acquistano un abbonamento mensile con chi acquista direttamente l'abbonamento annuale) pari a 275955. Supponendo che gli abbonati utilizzino quotidianamente la metropolitana,

e che siano uniformemente distribuiti tra le 76 diverse stazioni, il numero di abbonati che giornalmente si reca alla stazione di San Giovanni è di circa 3631. Poichè il numero di vidimazioni giornaliere medio presso la stazione di San Giovanni risulta essere pari a 10920, gli abbonati ne costituiscono il 33%. da queste considerazioni vengono quindi ricavate le seguenti probabilità:

- $p_A = 33\% \rightarrow$ percentuale di utenti abbonati
- $p_B + p_F = 67 \rightarrow$ percentuale di utenti che acquistano un biglietto presso la biglietteria elettronica o le casse fisiche.

Poichè dall'esperienza personale si ricava che gli utenti prediligono le casse automatiche, piuttosto che quelle fisiche, si deriva che:

- $p_B = 50\% \rightarrow$ utilizzatori casse automatiche
- $p_F = 17\% \rightarrow$ utilizzatori casse fisiche

Infine gli ascensori risultano utilizzati da pochi utenti (sembra ragionevole considerare un utente ogni 20), da cui ricaviamo la seguente probabilità:

- $p_E = 5\%$

Anche il mancato possesso di un titolo di viaggio valido è un fenomeno sempre più frequente. Da qui la volontà di considerare la seguente probabilità di non superare i controlli:

- $p_o = 10\%$

5.2 Modellazione dei centri

Gli arrivi degli utenti presso la stazione possono essere ritenuti "random", ossia l'arrivo ha pari probabilità di avvenire in un qualunque istante di tempo all'interno di un periodo $(0,t)$, e costituiscono perciò un processo di Poisson. Per tale motivo i tempi di interarrivo sono ben caratterizzati mediante una distribuzione esponenziale, la cui proprietà di "assenza di memoria" descrive correttamente la casualità e l'indipendenza con cui l'arrivo di un nuovo utente avviene rispetto agli utenti che hanno raggiunto la stazione precedentemente.

Inoltre data la disponibilità di dati reali riguardanti l'affluenza giornaliera degli utenti presso le diverse stazioni della metro di Roma, presenti in (1), è possibile definire dei tassi di arrivo coerenti con tali dati, come mostrato in seguito.

Per motivi analoghi a quelli presentati per gli arrivi, anche i tempi di servizio sono modellati con una distribuzione esponenziale. In questo caso non sono disponibili dati reali sul tempo effettivo impiegato dagli utenti per realizzare operazioni quali l'acquisto dei biglietti, il superamento dei tornelli o dei controlli. Tale tempi però possono essere ricavati dall'esperienza personale, in quanto utente abbonato per molti anni e quindi assiduo frequentatore della metropolitana di Roma, così da modellare il sistema nel modo più realistico possibile.

Da tali considerazioni si ottengono quindi le seguenti distribuzioni per i diversi centri che costituiscono il sistema:

1. Biglietteria automatica

- arrivi markoviani con tasso λ_2

- tempi di servizio esponenziali con media $E[S_i] = 90 \text{ s} = 1,5 \text{ min}$

Si sceglie di considerare un tempo medio di servizio maggiore rispetto a quello della biglietteria fisica poichè la biglietteria automatica richiede la stampa del biglietto in quel momento nonché la selezione della tipologia di biglietto desiderato mediante l'interfaccia della macchina, operazioni che

tipicamente impegnano l'utente per un tempo superiore a quello necessario all'operatore della cassa fisica a fornire un biglietto, tipicamente già stampato. Anche l'operazione di pagamento stesso potrebbe richiedere più tempo (talvolta il lettore di banconote della cassa automatica richiede l'inserimento più volte della stessa banconota per mancata lettura). La distribuzione considerata richiede un troncamento inferiore. Infatti da calcoli effettuati risultati che la $P\{t < 20 \text{ s}\} = \text{cdf}(t = 20) = 0.199263$, ossia circa il 20 %. Un tempo inferiore a 20 secondi per l'acquisto del biglietto presso le casse automatiche risulta però irrealistico, poiché la sola stampa del biglietto richiede circa 10 s (se devono essere stampati più biglietti in una transazione il tempo aumenta), che sommato al tempo necessario alla macchina per processare il denaro inserito e l'erogazione del resto, rende impossibile effettuare tale operazione in meno di 20 s. Da qui la volontà di inserire un lower bound a 20 per i valori che tale distribuzione può generare. Non è ritenuto necessario un upper bound invece, poiché la $P\{t > 4 \text{ min (240 s)}\} = 1 - \text{cdf}(t = 240) = 0.0694835$, ossia circa il 6%, valore ritenuto accettabile, poiché caratterizza realisticamente quei rari casi in cui l'utente riscontra problemi nell'acquisto del biglietto (banconote che non vengono accettate, problemi con l'interfaccia della cassa automatica, temporaneo malfunzionamento della cassa stessa), e impiega quindi un tempo maggiore.

2. Biglietteria fisica

- arrivi markoviani con tasso λ_3
- tempi di servizio esponenziali con media $E[S_i] = 40 \text{ s}$

Con considerazioni simili al caso della biglietteria automatica, anche per la cassa fisica si ritiene opportuno introdurre un lower bound al valore di 5 s, poiché un tempo inferiore a questa (che ha una probabilità pari a $P\{t < 5 \text{ s}\} = \text{cdf}(t = 5) = 0.117503$, ossia circa l'11%) risulterebbe non rappresentativo del sistema reale. Anche in questo caso non è stato ritenuto necessario introdurre vincoli superiori (rari casi in cui il tempo per l'acquisto può aumentare di molto, ad esempio per difficoltà nel fornire il resto al cliente, per mancanza di moneta, o problemi nell'effettuare un pagamento elettronico con carta).

3. Tornelli

- arrivi markoviani con tasso λ_4
- tempi di servizio esponenziali con media $E[S_i] = 5 \text{ s}$

Anche in questo caso, come nei precedenti, risulta necessario introdurre un lower bound, al valore di 1 s. La probabilità che il tempo sia inferiore a 1 s, considerando la distribuzione esponenziale, è pari $P\{t < 1 \text{ s}\} = \text{cdf}(t = 1 \text{ s}) = 0.181269$, ossia circa il 18%, valore di probabilità troppo elevato considerando che un tempo inferiore ad 1 s per l'attraversamento del tornello risulta impossibile (la sola operazione di validazione del biglietto o lettura dell'abbonamento richiede circa un secondo). Per quanto riguarda vincoli superiori, non sono considerati necessari in quanto tempi superiori (anche di molto) sono possibili in rari casi, ad esempio per malfunzionamenti del tornello o del lettore del biglietto (ad esempio $P\{t > 20 \text{ s}\} = 1 - \text{cdf}(t = 20) =$

0.181269, inferiore al 2%, data la velocità con cui la coda dell'esponenziale si avvicina allo 0).

4. Controllori

- arrivi markoviani con tasso λ_6
- tempi di servizio esponenziali con media $E[S_i] = 10$ s

Il tempo medio di servizio è stato scelto tenendo in considerazione le attività che un controllore deve tipicamente realizzare per validare un titolo di viaggio, ossia controllare il possesso e la data di validazione in caso di biglietto per singola corsa, oppure la validità dell'abbonamento e un documento con dati coerenti con quelli sulla tessera della metro nel caso di utenti abbonati. Si presuppone che per entrambe le attività un tempo medio di 10 secondi sia ragionevole.

Anche in questo caso risulta ragionevole collocare un lower bound a 2 s (la probabilità che $t < 2$ s risulterebbe altrimenti troppo elevata, pari a $\text{cdf}(t = 2 \text{ s}) = 0.181269$, circa il 18%). Un controllo che avviene con tempi inferiori ai 2 s risulta infatti non coerente con il sistema reale (in cui il controllore deve svolgere, anche se talvolta velocemente, diverse operazioni quali quelle indicate in precedenza). Non risulta necessario considerare un upperbound, poichè ad esempio nel caso di titolo di viaggio o abbonamento non valido il controllare rimane impegnato con l'utente per un tempo che può essere anche molto maggiore rispetto al valor medio (per la stesura della multa ad esempio).

5. Ascensori

- arrivi markoviani con tasso λ_8
- tempi di servizio con distribuzione normale con media $E[S_i] = 20$ s e varianza $\sigma^2 = 1$.

La scelta della distribuzione normale per questi tempi di servizio deriva dal fatto che il servizio fornito dal centro degli ascensori è caratterizzato dall'ingresso nell'ascensore e la selezione del piano, che prevedono tempi pressochè molto vicini tra loro (con una piccola variabilità), e lo spostamento dell'ascensore per raggiungere il piano desiderato (nel nostro contesto la banchina del treno), che rappresenta un tempo pressochè costante. La distribuzione normale modella bene questo comportamento, poichè avendo definito una varianza molto piccola (pari a 1) produce valori che con elevata probabilità (praticamente la totalità dei valori generati) sono compresi nell'intervallo [18,20], che si ritiene rappresenti realisticamente i tempi di servizio per tale centro.

Poichè la distribuzione normale ammette anche valori negativi, nonostante la generazione di questi abbia una probabilità trascurabile ($P\{t < 0 \text{ s}\} = \text{cdf}(t = 0) = 2.75362 \cdot 10^{-89}$), solo per motivi implementativi (evitare la generazione di bug nel sistema di simulazione), si sceglie di troncare inferiormente la distribuzione al valore 15 (a livello teorico la $P\{t < 15 \text{ s}\} = \text{cdf}(t = 15) = 2.86652 \cdot 10^{-7}$, quindi trascurabile) e superiormente in 25.

6. Banchina (passaggio del treno)

Inerentemente a tale centro "fittizio", è necessario modellare i tempi di interarrivo della metropolitana alla stazione oggetto dello studio. Secondo quanto riportato in (3), la frequenza media del passaggio dei treni nelle ore di punta risulta pari a un treno ogni tre minuti. Anche dall'esperienza personale si deriva che tempi compresi tra 3 e 5 minuti risultano realistici, sebbene la variabilità di questi tempi sia piuttosto elevato (non è raro dover aspettare anche 6 minuti per il prossimo treno). Per tale motivo si sceglie di modellare tali tempi secondo una distribuzione normale con media $E[s] = 240 \text{ s} = 4 \text{ min}$ e varianza pari a $\sigma^2 = 30$, troncata nell'intervallo $[120 \text{ s}, 360 \text{ s}]$ (ossia in minuti pari a $[2, 6]$). Inoltre il treno che raggiunge la stazione, non trattandosi di un capolinea, bensì una stazione intermedia, ha un certo numero di posti già occupati, che quindi comportano un numero di posti disponibili per i passeggeri che devono salire inferiore rispetto alla capacità massima del treno (1200 passeggeri). Non disponendo però di ulteriori informazioni sull'andamento di tale processo, si sceglie di modellare il numero di posti già occupati sul treno al momento del suo arrivo (considerando anche i passeggeri che scendono presso la stazione considerata) come una distribuzione di Poisson di media $E[n] = 400$ (valore che sembra ragionevole per una stazione del centro di Roma, dove il treno giunge con molti passeggeri già a bordo). Tale soluzione si preferisce rispetto a quella di modellare il numero di posti disponibili ad ogni passaggio del treno con un valore costante, che sarebbe risultata non coerente con il sistema considerato.

5.3 Tassi di arrivo e fasce orarie

Da (1) si ricava che il numero medio di passeggeri che giornalmente si recano presso la stazione di san giovanni (ossia numero medio di vimidazioni di biglietti/abbonamenti giornaliere) è pari a 10920. Si considerano quindi le seguenti fasce orarie e distribuzione dei passeggeri durante le 18 ore di operatività della linea A (5.30-23.30):

1. 5.30-7.30: 3% $\rightarrow \lambda = 0.045 \text{ utenti/s}$
2. 7.30-10.30: 30% $\rightarrow \lambda = 0.303 \text{ utenti/s}$
3. 10.30-14.30: 10% $\rightarrow \lambda = 0.076 \text{ utenti/s}$
4. 14.30-15.30: 20% $\rightarrow \lambda = 0.606 \text{ utenti/s}$
5. 15.30-18.30: 5% $\rightarrow \lambda = 0.05 \text{ utenti/s}$
6. 18.30-21.30: 27% $\rightarrow \lambda = 0.273 \text{ utenti/s}$
7. 21.30-23.30: 5% $\rightarrow \lambda = 0.076 \text{ utenti/s}$

Si evidenzia che le fasce orarie con l'affluenza maggiore sono quelle in corrispondenza degli orari di ingresso/uscita da scuole e uffici (fasce 2, 4, 6), orari in cui si intensifica l'utilizzo della linea metropolitana ad opera dei cittadini di Roma.

6. Modello computazionale

Le simulazioni del sistema sono realizzate mediante un programma ad oggetti in Java. I centri sono rappresentati dalla classe astratta *Center*, che definisce le proprietà necessarie a caratterizzare lo stato e l'evoluzione dei centri di ciascuna tipologia, tramite attributi che descrivono il numero di server che caratterizza il centro nella configurazione desiderata, il numero di job correntemente in quel centro, il numero di server occupati e un ID univoco necessario ad identificare il centro stesso. Tale classe definisce inoltre, senza fornire un'implementazione, i metodi che ogni centro "concreto" dovrà implementare, in funzione della sua specifica modellazione. Tra questi troviamo la generazione dei tempi di servizio, il routing dei job in uscita verso i centri successivi, le funzioni necessarie a processare l'arrivo e la partenza di un job dal centro, l'aggiornamento delle statistiche e la politica di assegnamento dei job ai server. Ciascuna delle diverse tipologie per i centri (descritte in precedenza) è rappresentata da una sottoclasse astratta di *Center*, che fornisce l'implementazione dei metodi indicati in precedenza, specifica per quella tipologia. Troviamo quindi le classi *MssqCenter* (Multi-server single-queue), *MsmqCenter* (Multi-server multi-queue) e *MsIsCenter* (Multi-server loss system). I centri fisici trovano invece la loro rappresentazione nelle classi (istanziabili, quindi non astratte), il cui nome fa riferimento al centro corrispondente (*ElectronicTicketCenter*, *TicketCenter*, *TurnstilesCenter*, *TicketInspectorsCenter*, *ElevatorsCenter*), ciascuna delle quali implementa la generazione dei tempi di servizio, secondo la distribuzione definita per quel centro, e la definizione del centro successivo, in base alla topologia del sistema precedentemente descritta. La scelta di modellare le superclassi astratte conferisce al sistema elevata flessibilità e modularità, in quanto l'aggiunta di nuovi centri, che modifica quindi la topologia del sistema, si realizza semplicemente estendendo la classe corrispondente alla tipologia desiderata per quel centro con una classe che rappresenta il centro fisico (tutte le funzionalità del tipo di centro sono infatti già incapsulate nelle superclassi, e comuni a tutti i centri fisici che le estendono). Inoltre ogni *Center* mantiene il riferimento alla lista dei server che lo costituiscono, ciascuno dei quali è rappresentato dalla classe *Server*, che mantiene informazioni sullo stato di occupazione del server (idle o busy), tempo totale di processamento dei job, numero di job processati, id univoco all'interno del centro di appartenenza e istante relativo all'ultimo completamento (necessario a realizzare la politica di assegnamento dei job, che, salvo casi particolari specificati in precedenza, prevede la scelta del server libero da più tempo). Le simulazioni sono realizzate secondo l'approccio di "next event simulation" descritto in (5), che prevede di realizzare l'evoluzione del sistema estraendo di volta in volta da una lista ordinata temporalmente il prossimo evento, modificare conseguentemente lo stato del sistema, e produrre gli eventuali eventi successivi. Ogni evento è rappresentato nel programma mediante la classe *Event*, che tiene traccia della tipologia dell'evento (*ARRIVAL*, *DEPARTURE*, *SAMPLING*, *SLOTCHANGE*, *TRAINARRIVAL*), l'istante del clock di simulazione in cui si realizza l'evento, il server e il centro a cui fa riferimento. Ogni centro inoltre mantiene un riferimento ad un oggetto della classe *Area*, responsabile di raccogliere incrementalmente le statistiche riguardanti la popolazione nel nodo, in coda, e nei server (tramite la legge di Little dalle statistiche sulle popolazioni si ricavano quelle sui tempi). Infine la classe *Time* rappresenta il clock di simulazione, tenendo traccia del suo valore corrente, che

viene aggiornato ogni qualvolta si verifica un nuovo evento. Le simulazioni, ad orizzonte finito e infinito, vengono realizzate mediante le rispettive classi *FiniteHorizonSimulation* e *InfiniteHorizonSimulation*, che permettono di configurare i parametri (tassi di arrivo, configurazione di ogni centro, numero e size dei batch, numero di repliche) utilizzati nelle due tipologie di simulazione.

6.1 Generazione dei processi stocastici

Per la generazione dei valori randomici, utilizzati per realizzare le diverse distribuzioni che modellano i processi stocastici del sistema, si è utilizzata un'implementazione del generatore di Lehmer, disponibile nella libreria *Rngs* fornite dagli autori di (5). In particolare per permettere il totale disaccoppiamento tra i diversi processi (degli arrivi, tempi di servizio di ogni centro e probabilità di routing), il generatore è stato utilizzato in modalità multistream, andando ad assegnare un diverso stream a ciascun processo, così da permettere in qualunque momento di manipolare uno di questi (ad esempio modificare il processo degli arrivi), senza impattare sugli altri processi stocastici e rendendo ripetibili i risultati delle simulazioni. Inerentemente alla simulazione a orizzonte finito, dove si realizzano diverse repliche della simulazione, variando esclusivamente lo stato iniziale del generatore da una replica all'altra, poiché ogni replica ha come obiettivo quello di produrre una stima indipendente della medesima statistica transiente, è necessario garantire la totale assenza di sovrapposizione tra una replica e l'altra dei valori randomici generati. L'approccio adottato per conseguire tale garanzia è quello proposto in (5), che prevede di utilizzare lo stato finale di ogni stream come stato iniziale del medesimo stream nella replica successiva. Questo meccanismo è realizzato collocando la funzione *PlantSeeds*, dedicata all'inizializzazione di ciascuno stream del generatore, al di fuori del ciclo che realizza le diverse repliche della simulazione a orizzonte finito.

Per quanto riguarda il troncamento delle distribuzioni, che è stato descritto nella sezione precedente, l'approccio utilizzato a livello implementativo è quello proposto in (5), secondo una tecnica nota come "*Constrained Inversion*", che prevede l'utilizzo della funzione di inversione (specifica della distribuzione desiderata) per la generazione dei valori delle distribuzioni troncate. Le funzioni di densità, di ripartizione e di inversione utilizzate sono contenute nella libreria *Rvms*, sempre fornita dagli autori di (5).

Nelle diverse simulazioni realizzate sono stati calcolati intervalli di confidenza per le statistiche desiderate utilizzando la libreria *Estimate*, che fa utilizzo dell'algoritmo "one-pass" di Welford per il calcolo della media e della deviazione standard del campione passato in input.

Il programma che realizza la simulazione è disponibile in:

<https://github.com/matteopallagrosi/progettoPMCSN>.

7. Verifica

Nel processo di verifica è stato scelto il tasso di arrivo di una delle fasce orarie, nello specifico $\lambda = 0.05$ utenti/s, e la configurazione (4,1,2,2,2), che permette di avere una utilizzazione inferiore ad 1 in ciascun centro analizzato. I valori ottenuti analiticamente sono stati quindi confrontati con i valori alla stazionarietà ottenuti mediamente la simulazione, al fine verificando la correttezza del modello computazionale costruito. Per garantire l'applicabilità delle formule analitiche, la

politica di routing verso le code del centro 3 (tornelli) è stata modificata, così da avere delle probabilità di routing distribuite uniformemente sulle diverse coppie sever-coda (nella configurazione considerata, essendoci due tornelli, $p = \frac{1}{2}$ che l'utente si diriga verso ciascuno di questi due tornelli), quindi indipendenti dallo stato del centro stesso. Inoltre senza perdita di generalità sono state considerate le distribuzioni senza troncamento (l'intervallo di valori che assume ogni distribuzione è infatti parametrico, quindi la correttezza del programma è indipendente dalla scelta di tali parametri, con l'assunzione di correttezza della generazione di ogni distribuzione ad opera della libreria *Rvms* utilizzata). Di seguito sono riportati i calcoli realizzati con le formule analitiche, e i risultati della simulazione, messi a confronto, per ogni centro:

- **1. Biglietteria automatica (Mssq):**

Si considera $\lambda_2 = p_B \cdot \lambda = 0.025$ utenti/s, da cui $\bar{r} = 40$ s, e una distribuzione esponenziale per il tempo di servizio con $E(S_i) = 90$ s, da cui $\mu = 0.011$ utenti/s. Numero di server $m = 4$.

$$\rho = \frac{\lambda}{m \cdot \mu} = 0.5625$$

$$p(0) = \left[\sum_{i=0}^{m-1} \frac{(m\rho)^i}{i!} + \frac{(m\rho)^m}{m! \cdot (1-\rho)} \right]^{-1} = 0.0988008$$

$$P_Q = \frac{(m\rho)^m}{m! \cdot (1-\rho)} \cdot p(0) = 0.2411777$$

$$E(T_Q) = \frac{P_Q \cdot E(S)}{1-\rho} = 12.403428 \text{ s}$$

$$E(T_S) = E(T_Q) + E(S_i) = 102.403428 \text{ s}$$

$$E(N_Q) = \frac{P_Q \cdot \rho}{1-\rho} = \lambda \cdot E(T_Q) = 0.3100857 \text{ utenti}$$

$$E(N_S) = E(N_Q) + m \cdot \rho = 2.5600857 \text{ utenti}$$

Statistica	Risultato analitico	Risultato simulazione
ρ	0.5625	0.562864 +/- 0.003453
\bar{r}	40	39.964680 +/- 0.134478
$E(T_Q)$	12.403428	12.312980 +/- 0.440024
$E(T_S)$	102.403428	102.317189 +/- 0.678768
$E(N_Q)$	0.3100857	0.308306 +/- 0.011424
$E(N_S)$	2.5600857	2.560789 +/- 0.020768

- **2. Biglietteria fisica (Mssq):**

Si considera $\lambda_3 = p_F \cdot \lambda = 0.0085$ utenti/s, da cui $\bar{r} = 117.647058$ s, e una distribuzione esponenziale per il tempo di servizio con $E(S) = 40$ s, da cui $\mu = 0.025$ utenti/s. Numero di server $m = 1$.

$$\rho = \frac{\lambda}{\mu} = 0.34$$

$$E(T_Q) = \frac{\rho * E(S)}{1 - \rho} = 20.606060 \text{ s}$$

$$E(T_S) = E(T_Q) + E(S) = 60.606060 \text{ s}$$

$$E(N_Q) = \frac{\rho^2}{1 - \rho} = \lambda * E(T_Q) = 0.175151 \text{ utenti}$$

$$E(N_S) = E(N_Q) + \rho = 0.515151 \text{ utenti}$$

Statistica	Risultato analitico	Risultato simulazione
ρ	0.34	0.338511 +/- 0.002412
\bar{r}	117.647058	118.081327 +/- 0.703539
$E(T_Q)$	20.606060	20.540762 +/- 0.513639
$E(T_S)$	60.606060	60.496107 +/- 0.649553
$E(N_Q)$	0.175151	0.174071 +/- 0.004532
$E(N_S)$	0.515151	0.512582 +/- 0.006252

- **3. Tornelli (Msmq):**

Si considera $\lambda_4 = \lambda = 0.05$ utenti/s. Poichè consideriamo $m = 2$ (con routing uniforme) verso ogni centro è diretto un flusso $\lambda/2 = 0.025$ utenti/s, da cui $\bar{r} = 40$ s, e una distribuzione esponenziale per il tempo di servizio con $E(S) = 5$ s, da cui $\mu = 0.2$ utenti/s. Essendo le probabilità di routing uniformi le due coppie server-coda esibiscono lo stesso comportamento.

$$\rho = \frac{\lambda/2}{\mu} = 0.125$$

$$E(T_Q) = \frac{\rho * E(S)}{1 - \rho} = 0.714285 \text{ s}$$

$$E(T_S) = E(T_Q) + E(S) = 5.714285 \text{ s}$$

$$E(N_Q) = \frac{\rho^2}{1 - \rho} = \lambda * E(T_Q) = 0.017857 \text{ utenti}$$

$$E(N_S) = E(N_Q) + \rho = 0.142857 \text{ utenti}$$

Statistica	Risultato analitico	Risultato simulazione
ρ	0.125	0.125122 +/- 0.000569
\bar{r}	40	39.980332 +/- 0.125592

E(T_Q)	0.714285	0.717810 +/- 0.013197
E(T_S)	5.714285	5.719595 +/- 0.027229
E(N_Q)	0.017857	0.017959 +/- 0.000342
E(N_S)	0.142857	0.143080 +/- 0.000818

- 4. Controllori (Msls):**

Si considera $\lambda_s = \lambda = 0.05$ utenti/s, e una distribuzione esponenziale per il tempo di servizio con $E(S_i) = 10$ s, da cui $\mu = 0.1$ utenti/s. Numero di server $m = 2$.

$$p_{loss} = \Pi_2 = \frac{\left(\frac{\lambda}{\mu}\right)^m / m!}{\sum_0^m \left(\frac{\lambda}{\mu}\right)^j / j!} = 0.076923$$

$$\lambda' = \lambda * (1 - p_{loss}) = 0.0461538$$

$$\bar{r} = \frac{1}{\lambda'} = 21.666681 \text{ s}$$

$$E(N_s) = \Pi_1 * 1 + \Pi_2 * 2 = 0.461538 \text{ utenti}$$

$$E(T_s) = \frac{E(N_s)}{\lambda'} = 10 \text{ s}$$

$$E(N_Q) = 0 \text{ utenti}$$

$$E(T_Q) = 0 \text{ utenti}$$

$$\rho = \frac{\lambda'}{m * \mu} = 0.230769$$

Statistica	Risultato analitico	Risultato simulazione
ρ	0.230769	0.230234 +/- 0.000832
\bar{r}	21.666681	21.680097 +/- 0.046071
E(T_Q)	0	0.000000 +/- 0.000000
E(T_S)	10	9.982570 +/- 0.021989
E(N_Q)	0	0.000000 +/- 0.000000
E(N_S)	0.461538	0.460470 +/- 0.001399
p_{loss}	0.076923	0.077016 +/- 0.000698

- 5. Ascensori (Mssq):**

Si considera $\lambda_7 = [\lambda \cdot (1 - p_{skip}) \cdot (1 - p_o)] + (\lambda \cdot p_{skip}) = 0.045384$ utenti/s, da cui $\lambda_8 = p_E \cdot \lambda_7 = 0.002269$ utenti/s (quindi $r = 440.677966$ s) e una distribuzione normale per il tempo di servizio con $E(S_i) = 20$ s, da cui $\mu = 0.05$ utenti/s, e varianza $\sigma^2 = 1$. Numero di server $m = 2$. Per il calcolo del tempo di attesa in

coda si utilizza una generalizzazione della formula di Erlang-C (usata nei casi precedenti), per una distribuzione generica del tempo di servizio, nota come legge di congestione di Kingman.

$$\rho = \frac{\lambda}{m * \mu} = 0.0226923$$

$$p(0) = \left[\sum_{i=0}^{m-1} \frac{(m\rho)^i}{i!} + \frac{(m\rho)^m}{m! * (1-\rho)} \right]^{-1} = 0.955622$$

$$P_Q = \frac{(m\rho)^m}{m! * (1-\rho)} * p(0) = 0.001007029$$

$$E(T_Q)^{M/M/m} = \frac{P_Q * E(S)}{1-\rho} = 0.01030411 \text{ s}$$

$$E(T_Q)^{M/G/m} = \frac{(C^2 + 1)}{2} * E(T_Q)^{M/M/m} = 0.005164938 \text{ s}$$

$$E(T_S) = E(T_Q) + E(S_i) = 20.005164 \text{ s}$$

$$E(N_Q) = \lambda * E(T_Q) = 0.000011720 \text{ utenti}$$

$$E(N_S) = \lambda * E(T_S) = 0.045396 \text{ utenti}$$

Statistica	Risultato analitico	Risultato simulazione
ρ	0.0226923	0.022666 +/- 0.000248
\bar{r}	440.677966	441.945584 +/- 4.774138
$E(T_Q)$	0.005164938	0.007836 +/- 0.003625
$E(T_S)$	20.005164	20.014298 +/- 0.010863
$E(N_Q)$	0.000011720	0.000018 +/- 0.000008
$E(N_S)$	0.045396	0.045361 +/- 0.000496

Questo centro è stato testato anche con una diversa configurazione, ossia $m = 1$. In questo caso si ottiene:

$$\rho = \frac{\lambda}{\mu} = 0.0453846$$

$$E(T_Q) = \frac{\rho}{1-\rho} * \frac{(C^2 + 1)}{2} * E(S) = 0.476611 \text{ s}$$

$$E(T_S) = E(T_Q) + E(S) = 20.476611 \text{ s}$$

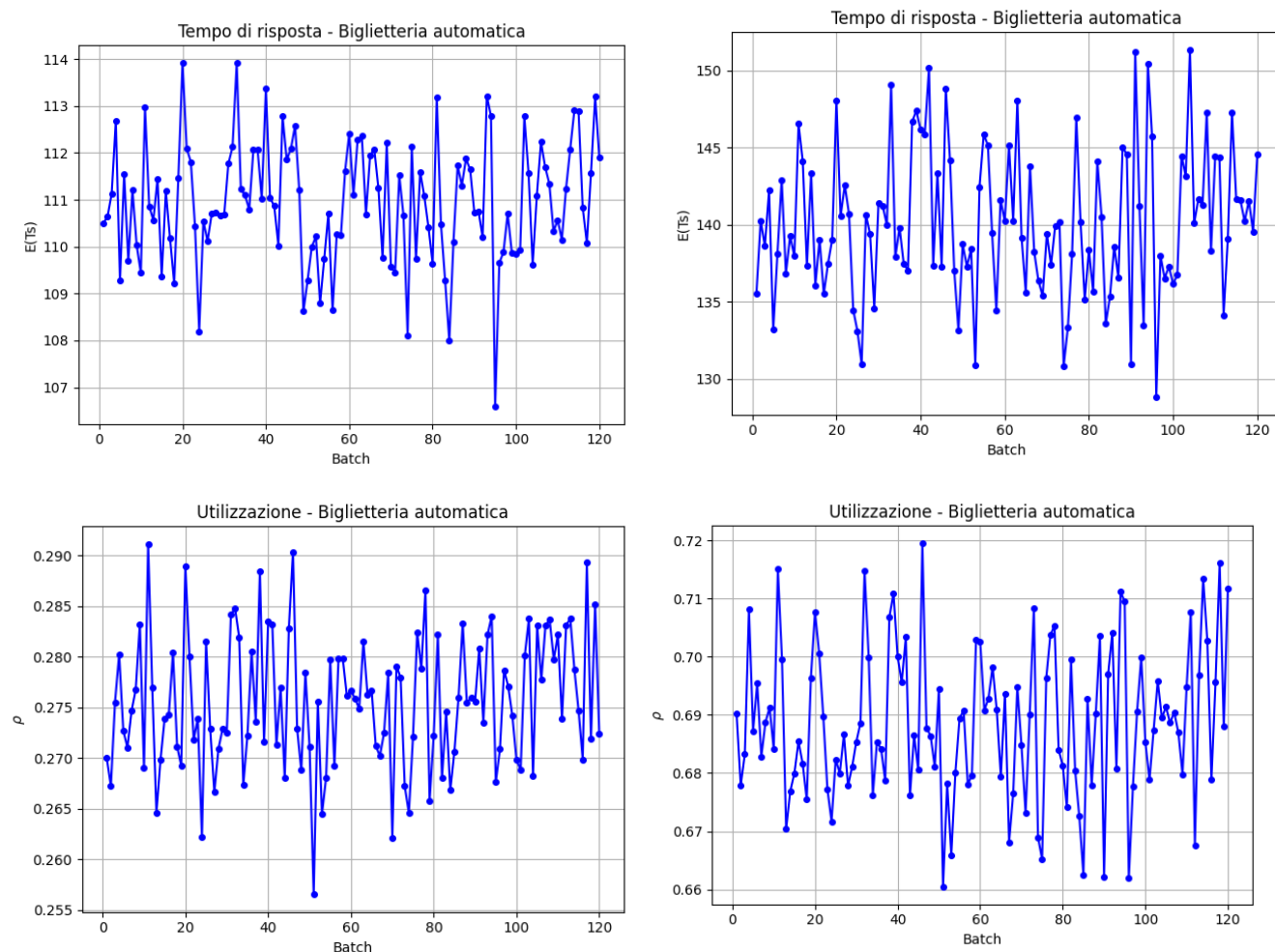
$$E(N_Q) = \frac{\rho^2}{2 * (1-\rho)} * (1 + C^2) = 0.00108154 \text{ utenti}$$

$$E(N_S) = E(N_Q) + \rho = 0.0464661 \text{ utenti}$$

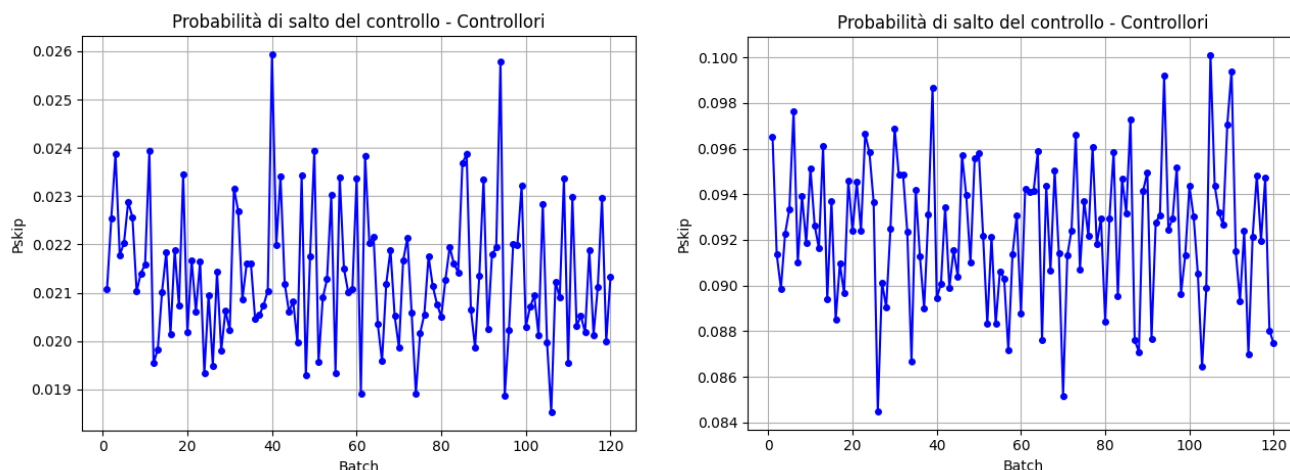
Statistica	Risultato analitico	Risultato simulazione
ρ	0.0453846	0.045343 +/- 0.000494
\bar{r}	440.677966	441.945584 +/- 4.774138
$E(T_Q)$	0.476611	0.461380 +/- 0.034006
$E(T_S)$	20.476611	20.014298 +/- 0.010863
$E(N_Q)$	0.00108154	0.001048 +/- 0.000081
$E(N_S)$	0.0464661	0.046392 +/- 0.000527

8. Validazione

Nella fase di validazione, non avendo a disposizione dati reali sulle performance del sistema di riferimento, e quindi non potendo confrontare l'output del modello computazione con quello del sistema reale, sono stati invece eseguiti dei controlli di consistenza, con l'obiettivo di dimostrare che il modello costruito si comportasse coerentemente al sistema rappresentato. Si è quindi considerata una configurazione dei centri (4,1,2,2,2), studiando il comportamento del sistema per due diversi tassi d'ingresso, in particolare $\lambda_1 = 0.02$ utenti/s e $\lambda_2 = 0.05$ utenti/s. Il risultato atteso è che, dato l'aumento dei tassi di ingresso, i tempi di risposta e le utilizzazioni dei centri debbano aumentare. Si è perciò preso in considerazione il tempo di risposta e le utilizzazioni ottenute con i due tassi di arrivo, in riferimento al centro della biglietteria elettronica. Le figure sottostanti mostrano i risultati ottenuti.



Nelle figure a sinistra sono riportate tempo di risposta e utilizzazione del centro, considerando $\lambda_1 = 0.02$ utenti/s, mentre a destra con $\lambda_2 = 0.05$ utenti/s. Ad un aumento di λ coerentemente un aumento di tempi e utilizzazioni. Infine come ulteriore statistica di cui studiare il comportamento al fine di validare il modello si considera la probabilità di salto del controllo del centro controllori. Con i medesimi tassi di ingresso sono eseguite le simulazioni, con i seguenti risultati:



Come evidenziato dai grafici, ad un maggior tasso di arrivo corrisponde una maggiore probabilità per gli utenti di saltare il controllo di un titolo di viaggio valido presso i controllori, dato che, aumentando la pressione su quel centro, si hanno maggiori probabilità di trovare i controllori già occupati all'arrivo di un utente, che quindi potrà superarli senza sottoporsi al controllo.

9. Progettazione degli esperimenti

La progettazione degli esperimenti prevede tre fasi:

1. analisi del collo di bottiglia
2. simulazione all'orizzonte infinito
3. simulazione all'orizzonte finito

Con l'analisi del collo di bottiglia l'obiettivo è quello di determinare il centro caratterizzato dalla domanda di servizio maggiore, che tiene in conto del carico su quel centro e della sua capacità di smaltire tale carico. Questo permette negli esperimenti di simulazione che seguono di avere una guida nel dimensionare adeguatamente il sistema per sostenere il flusso in ingresso.

La simulazione a orizzonte infinito prevede di simulare il sistema per un tempo teoricamente infinito, ossia nella pratica sufficientemente lungo affinché questo possa raggiungere la stazionarietà (se raggiungibile). Nel nostro contesto viene quindi simulata ogni fascia oraria con una configurazione scelta, e la simulazione è condotta con il metodo dei batch means, descritto in (5). Nell'ambito di questo studio di simulazione si è scelto di utilizzare un numero di batch $k = 120$, ciascuno di dimensione $b = 10000$. Nella scelta della dimensione del batch si è seguito il criterio proposto da Banks, Carson, Nelson, e Nicol, che suggeriscono di aumentare la dimensione del batch finché l'autocorrelazione lag one tra i batch non è inferiore a 0.2. Tale calcolo è realizzato utilizzando la libreria Acs, fornita dagli

autori di (5). L'obiettivo della simulazione a orizzonte infinito è stato quello di individuare il numero minimo di serventi (che implica minori costi per la stazione), che permettesse di rispettare i QoS.

Alla luce dei risultati ottenuti con la simulazione all'orizzonte infinito viene quindi realizzata una simulazione a orizzonte finito, che simula il comportamento del sistema sulle 18 ore di operatività della stazione, introducendo dei cambiamenti nel tasso di ingresso, al fine di caratterizzare la diversa affluenza dei passeggeri presso la stazione che si sperimenta durante la giornata. Tale simulazione è replicata 128 volte, così da poter mediare le statistiche ottenute sui vari centri sull'ensemble di repliche. La generazione dei processi stocastici segue l'approccio per l'utilizzo dei generatori già descritto nella descrizione del modello computazionale.

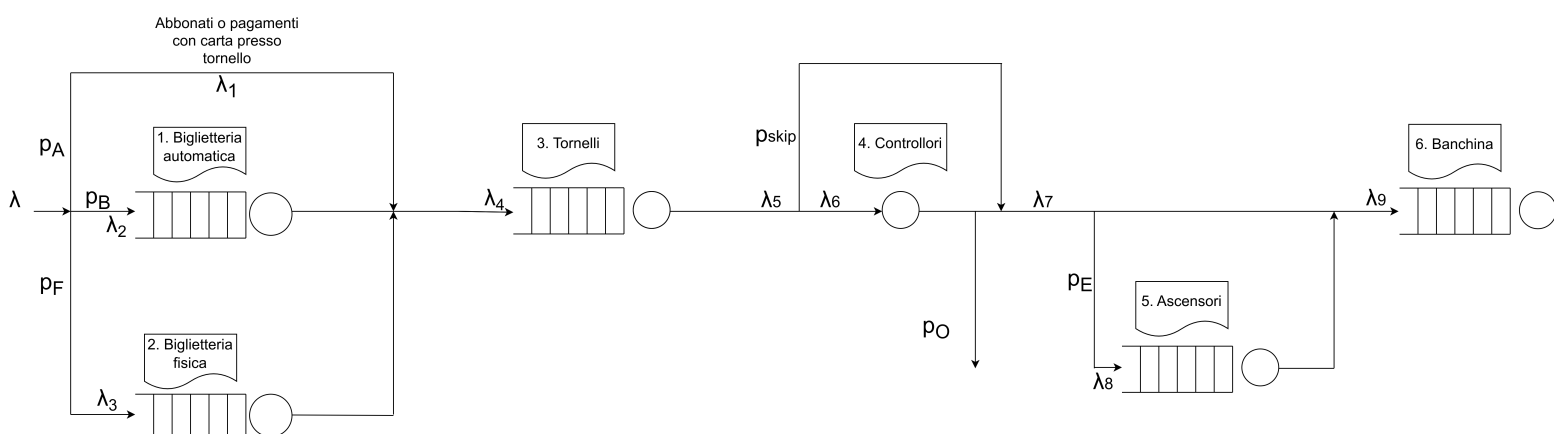
Al fine di realizzare i cambiamenti di fascia oraria, è stato introdotto l'evento di **SLOTCHANGE**, in corrispondenza del quale viene cambiata la configurazione di ogni centro. Il cambiamento di configurazione avviene nel modo seguente:

- se il numero di server deve essere aumentato rispetto a quello corrente, vengono creati nuovi server così da raggiungere il numero desiderato. Se sono presenti job in coda, ogni nuovo server preleva (finché sono disponibili) job dalla coda e diventa immediatamente occupato. Questo meccanismo tenta di emulare il comportamento di utenti reali (quando ad esempio viene attivata una nuova cassa, gli utenti si dirigono immediatamente verso di essa).
- Se il numero di server deve diminuire, qualora siano disponibili sufficienti server correntemente liberi, questi vengono eliminati. Altrimenti si elimina il maggior numero possibile di server tra quelli liberi, e si attende che server attualmente occupati terminino per poterli eliminare e raggiungere la configurazione voluta.

Oltre all'evento di **SLOTCHANGE**, viene introdotto anche l'evento di **SAMPLING**, con cui vengono calcolate le statistiche dei diversi centri in quello slot di campionamento, con conseguente azzeramento di quest'ultime.

Con la simulazione all'orizzonte finito è possibile valutare se il dimensionamento del sistema individuato alla stazionarietà permetta di rispettare i QoS anche su un periodo limitato di operatività. Si studia quindi il comportamento transiente del sistema, che permette di comprendere come i cambiamenti di fascia oraria e la conseguente variazione del tasso di ingresso possano essere adeguatamente gestiti. Nelle simulazioni tutte le statistiche mostrate sono ottenute utilizzando intervalli di confidenza al 95%.

9.1 Analisi del collo di bottiglia



Al fine di individuare il collo di bottiglia del sistema, ossia il centro su cui ricade la domanda di servizio maggiore, consideriamo ogni centro caratterizzato da un singolo servente. I tassi di arrivo presso ogni centro sono denominati come mostrato in figura. Il tempo di servizio richiesto presso l'i-esimo centro è indicato con S_i . Denotiamo con γ il flusso degli arrivi dall'esterno, nel nostro caso $\gamma = \lambda$. I tempi di servizio presso ciascun centro risultano:

$$S_1 = 90 \text{ s}$$

$$S_2 = 40 \text{ s}$$

$$S_3 = 5 \text{ s}$$

$$S_4 = 10 \text{ s}$$

$$S_5 = 20 \text{ s}$$

Consideriamo quindi le medesime equazioni di traffico già individuate nella modellazione delle specifiche, di cui ci interessano i tassi in ingresso ai centri, in funzione del flusso dall'esterno:

$$\lambda_1 = p_A \cdot \lambda$$

$$\lambda_2 = p_B \cdot \lambda$$

$$\lambda_3 = p_F \cdot \lambda$$

$$\lambda_4 = \lambda_1 + \lambda_2 + \lambda_3 = (p_A + p_B + p_F) \cdot \lambda = \lambda$$

$$\lambda_5 = \lambda_4 = \lambda$$

$$\lambda_6 = \lambda_5 \cdot (1 - p_{\text{skip}}) = \lambda \cdot (1 - p_{\text{skip}})$$

$$\lambda_7 = [\lambda_6 \cdot (1 - p_O)] + (\lambda_5 \cdot p_{\text{skip}}) = [\lambda \cdot (1 - p_{\text{skip}}) \cdot (1 - p_O)] + (\lambda \cdot p_{\text{skip}})$$

$$\lambda_8 = p_E \cdot \lambda_7 = p_E \cdot \{[\lambda \cdot (1 - p_{\text{skip}}) \cdot (1 - p_O)] + (\lambda \cdot p_{\text{skip}})\}$$

$$\lambda_9 = \lambda_8 + (1 - p_E) \cdot \lambda_7 = p_E \cdot \lambda_7 + (1 - p_E) \cdot \lambda_7 = \lambda_7 = [\lambda \cdot (1 - p_{\text{skip}}) \cdot (1 - p_O)] + (\lambda \cdot p_{\text{skip}})$$

Dalle equazioni di traffico possiamo ottenere il numero medio di visite presso il centro i-esimo, indicate con v_i , calcolate come $v_i = (\text{tasso ingresso centro } i / \lambda)$:

$$v_1 = \lambda_1 / \lambda = p_A$$

$$v_2 = \lambda_2 / \lambda = p_B$$

$$v_3 = \lambda_3 / \lambda = p_F$$

$$v_4 = \lambda_4 / \lambda = 1$$

$$v_5 = \lambda_5 / \lambda = (1 - p_{\text{skip}})$$

$$v_5 = \lambda_8 / \lambda = p_E \cdot [(1 - p_{\text{skip}}) \cdot (1 - p_O) + p_{\text{skip}}]$$

Considerando come p_{skip} quella che si ottiene avendo un singolo servente presso il centro 4 e tasso d'ingresso $\lambda = 0.02$ utenti/s (quindi $p_{\text{skip}} = 0.16666$ utenti/s), si ottengono le seguenti visite (le probabilità di routing utilizzate sono quelle definite nel modello delle specifiche):

$$v_1 = 0.5$$

$$v_2 = 0.17$$

$$v_3 = 1$$

$$v_4 = 0.83334$$

$$v_5 = 0.04583$$

Otteniamo quindi la seguente domanda media di servizio presso ciascun centro:

$$D_1 = v_1 * S_1 = 45$$

$$D_2 = v_2 * S_2 = 6.8$$

$$D_3 = v_3 * S_3 = 5$$

$$D_4 = v_4 * S_4 = 8.3334$$

$$D_5 = v_5 * S_5 = 0.9166$$

Avendo precedentemente verificato la correttezza del modello computazione, questo può essere utilizzato per ottenere, come mostrato nella tabella sottostante, risultati analoghi a quelli ottenuti analiticamente:

Centro	Visite	Servizio
1. Biglietteria automatica	0.500584 +/- 0.001219	90.028101 +/- 0.251242
2. Biglietteria fisica	0.169643 +/- 0.000795	39.976247 +/- 0.147615
3. Tornelli	1.000173 +/- 0.001662	4.998571 +/- 0.008200
4. Controllori	0.832841 +/- 0.001409	9.988917 +/- 0.019153
5. Ascensori	0.045941 +/- 0.000395	20.001242 +/- 0.007671

Poichè visite e tempi di servizio ottenuti sono coerenti con quelli ottenuti analiticamente, risultati analoghi valgono per le domande medie di servizio, date dal prodotto $D_i = v_i * S_i$.

Dai risultati ottenuti si conclude che il centro 1 (biglietteria automatica) risulta il collo di bottiglia del sistema, risultato coerente con il fatto che tale centro è interessato dalla maggior parte del flusso in ingresso al sistema (quindi numero maggiore di visite) e presenta il tempo medio di servizio maggiore. Tale risultato riflette il comportamento del sistema reale di riferimento, poiché le code presso le casse automatiche si rivelano essere, per esperienza personale, l'aspetto critico del sistema. La corretta configurazione di tale centro è quindi un aspetto fondamentale di cui tenere in conto nelle simulazioni presentate in seguito.

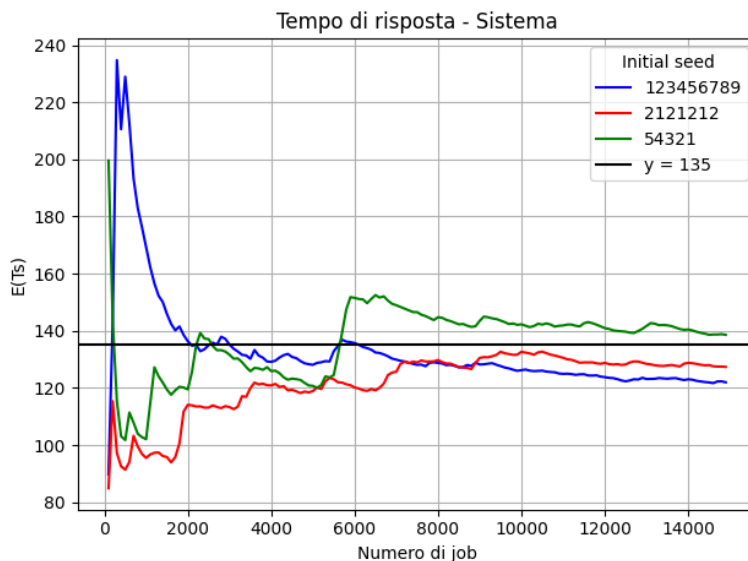
9.2 Analisi del transitorio

Questa fase di analisi ha come obiettivo lo studio del comportamento transitorio del sistema, al fine di comprendere se il sistema raggiunge la stazionarietà, ed in caso positivo, quando questa viene conseguita e come si comporta prima di raggiungerla (oppure come si comporta prima di divergere nel caso il sistema non fosse stabile). A tale scopo il sistema viene simulato con diversi seed iniziali, andando a campionare le statistiche medie ogni n job processati (nel nostro caso $n = 100$), in modo da poter studiare come si comporta il sistema al crescere del numero di job processati (e quindi conseguentemente del tempo simulato). Al fine di rendere i risultati di tale analisi significativi, si sceglie di realizzare la simulazione con due diversi tassi di ingresso (delle prime due fasce orarie), in particolare un tasso di ingresso molto basso ($\lambda = 0.045$ utenti/s) e un tasso di ingresso elevato ($\lambda = 0.303$ utenti/s). Si scelgono inoltre configurazioni del sistema tali per cui le utilizzazioni dei

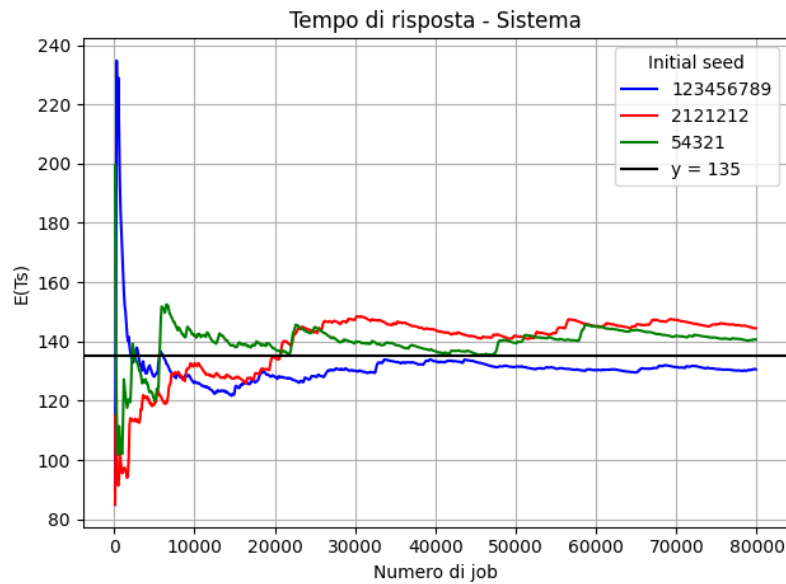
centri siano inferiori ad 1 (altrimenti è assicurata la divergenza del sistema, dato l'aumento all'infinito di tempi e popolazioni medie), rispettivamente (3,1,1,1,1) e (20,3,3,2,1). Le considerazioni che vengono fatte per questi casi possono essere estese anche alle altre fasce orarie. I due casi indicati sono mostrati di seguito:

Caso 1 ($\lambda = 0.045$ utenti/s):

Si va quindi a studiare il comportamento del tempo di risposta del sistema, con tre diversi seed iniziali, al crescere del numero di job processati, fino ad un limite superiore di 15000 job processati. Gli andamenti ottenuti per i diversi seed sono mostrati in figura con diversi colori.



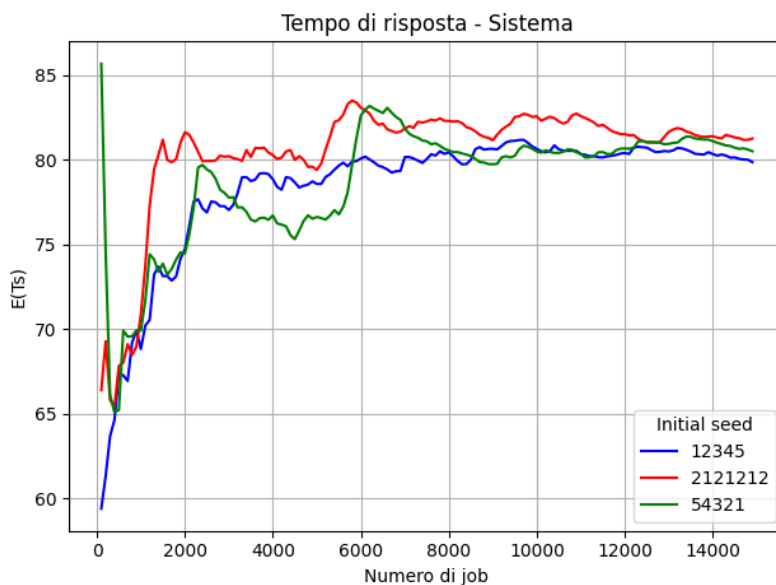
Dopo una fase iniziale in cui il comportamento del sistema è ancora fortemente dipendente dallo stato iniziale di quest'ultimo (il sistema è inizialmente vuoto), e dipendente anche dalla scelta del seed iniziale per la generazione dei processi stocastici coinvolti (arrivi e tempi di servizio), si osserva una lenta convergenza verso un valore di stazionarietà, che da questa osservazione empirica (la media teorica non è nota, avendo il sistema caratteristiche che non permettono di risolverlo nella sua interezza analiticamente) risulta essere prossimo al valore $E(Ts) \approx 135$ s, evidenziato in figura con il colore nero. Inoltre tale convergenza inizia ad essere evidente e maggiormente stabile quando si è prossimi al processamento di 10000 job. Per essere certi del conseguimento di questa stazionarietà, si esegue un'ulteriore simulazione molto più lunga, che prosegue fino al raggiungimento di 80000 job processati. I risultati ottenuti sono mostrati nella figura sottostante (che ovviamente per i primi 15000 job coincide con quella mostrata sopra):



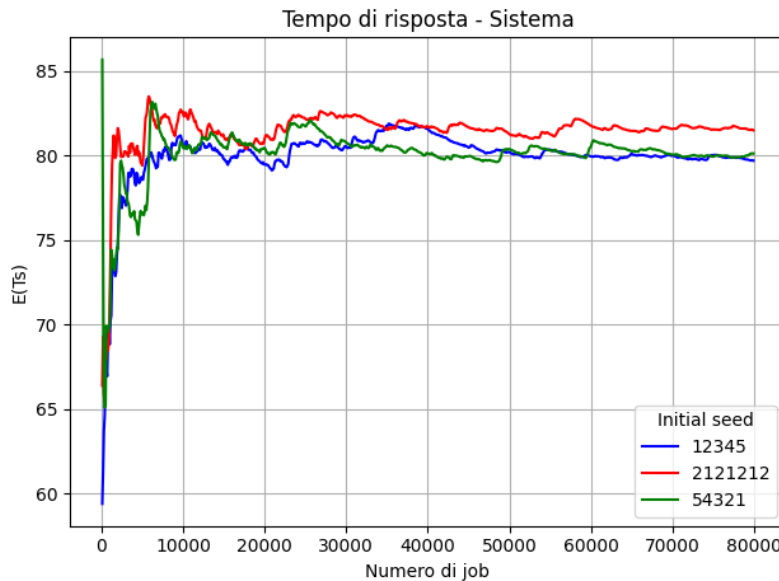
Questa simulazione molto più lunga conferma che la stazionarietà è effettivamente conseguita, con il valore indicato in precedenza.

Caso 2 ($\lambda = 0.303$ utenti/s):

Si esegue la simulazione con le stesse modalità indicate in precedenza, ma utilizzando un diverso tasso di ingresso. I risultati ottenuti sono mostrati di seguito:



Anche in questo caso si osserva la convergenza del sistema alla stazionarietà, più lenta nel caso di alcuni seed (come quelli evidenziati in rosso e verde), più rapida per altri (come per la scelta del seed in blu). La totale indipendenza dal seed iniziale e dallo stato iniziale del sistema si consegue per un numero di job processato compreso tra 8000 e 10000. Anche in questo caso vediamo come si comporta il sistema su una simulazione più lunga:



Si conferma quindi il comportamento già evidenziato con la simulazione precedente. Infine il valore raggiunto alla stazionarietà sembra collocarsi su $E(T_s) \approx 80$ s.

I risultati di questa analisi sono presi in considerazione per il dimensionamento dei batch da utilizzare nella simulazione a orizzonte infinito (oltre che l'analisi del valore di autocorrelazione come già indicato), e soprattutto assicura l'applicabilità di tale simulazione in quanto la stazionarietà è effettivamente conseguita.

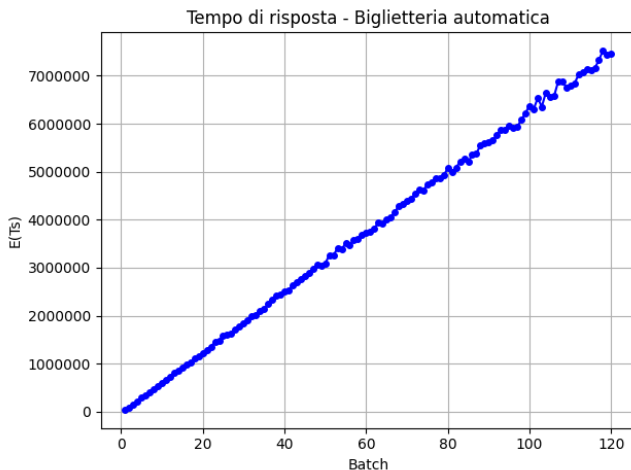
NOTA: i valori che sono stati ottenuti con la simulazione a orizzonte infinito per la prima e seconda fascia oraria (ma lo stesso ragionamento si può estendere anche alle altre fasce) appaiono in linea con quanto osservato con l'analisi del transitorio (135.720395 +/- 2.116704 s per la prima fascia oraria, 79.333859 +/- 0.517075 nella seconda fascia). Il metodo dei batch means, così configurato ($k = 120$, $b = 10000$), andando a mediare le statistiche medie ottenute nei diversi batch, mostra la convergenza del sistema su valori coerenti con quelli ottenuti con questa analisi. In particolare poiché ogni batch presenta come stato iniziale lo stato finale del batch precedente (e quindi sempre un diverso stato iniziale, che per il primo batch sarà il sistema vuoto, per i successivi una diversa quantità di job nei vari centri, in base a come sta evolvendo il sistema), è mostrata l'indipendenza che si ottiene alla stazionarietà rispetto allo stato iniziale del sistema.

9.3 Simulazione a orizzonte infinito

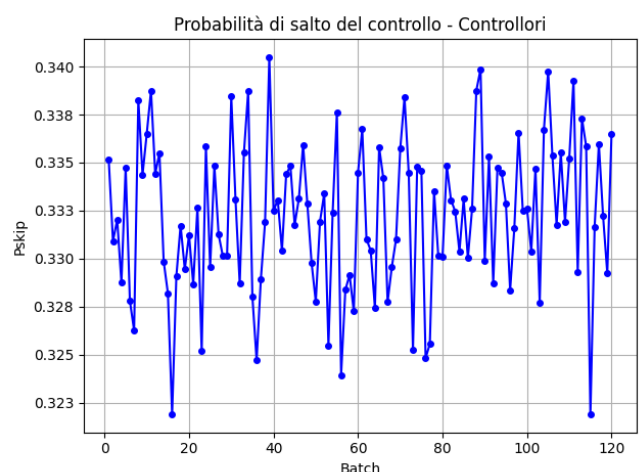
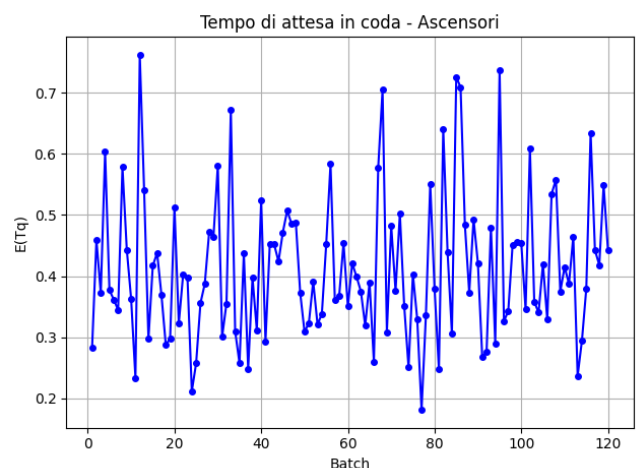
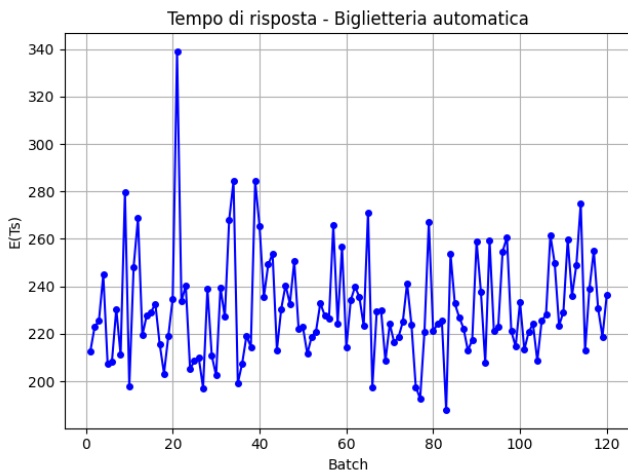
Fascia 1 (5.30-7.30):

Questa fascia oraria è caratterizzata dall'affluenza minore durante la giornata ($\lambda = 0.045$ utenti/s), e per tale motivo si presuppone l'utilizzo di una configurazione con un ridotto numero di server. Considerando l'analisi del collo di bottiglia precedentemente effettuata, si cerca di favorire un numero maggiore di server nel centro 1, caratterizzato dalla domanda di servizio maggiore. Per tale motivo si valuta dapprima la configurazione (2,1,1,1,1). Tale configurazione, come mostrato

dalla figura sottostante, comporta una utilizzazione $\rho > 1$ nel centro 1, con conseguenti tempi che crescono infinitamente.



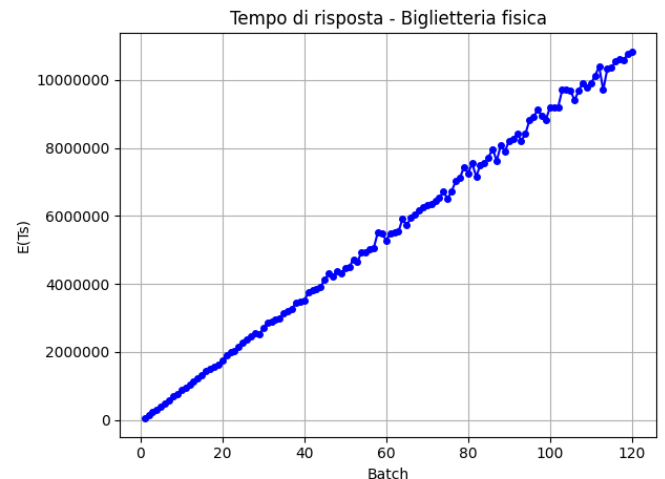
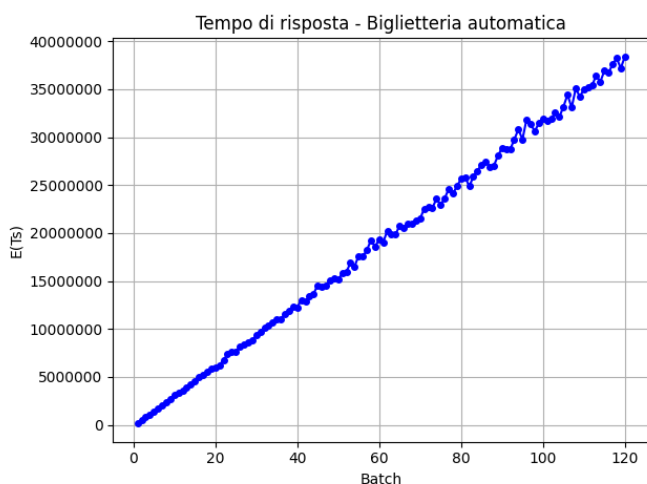
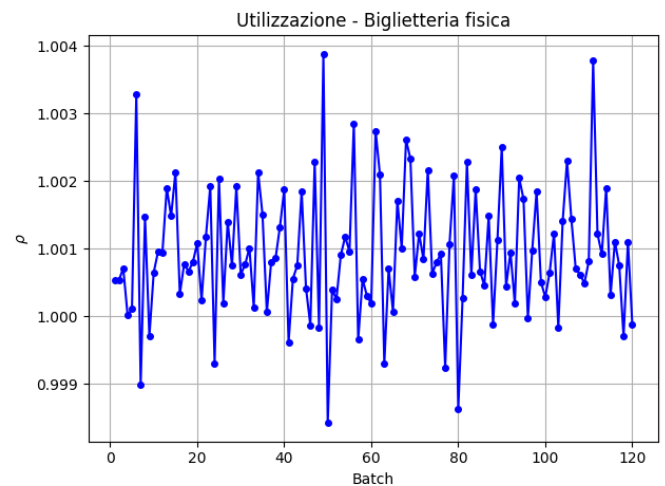
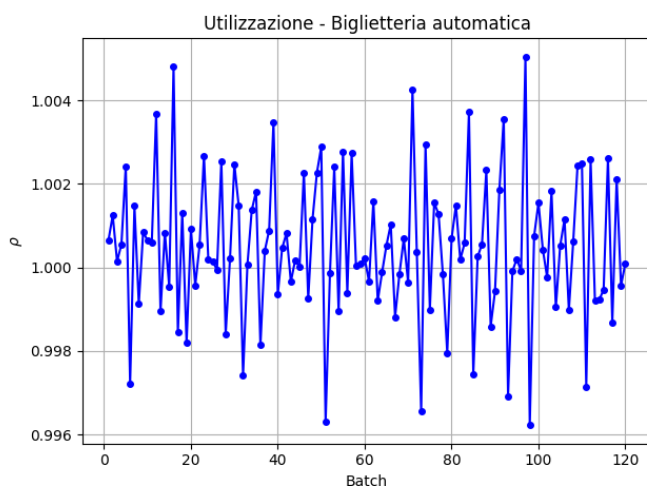
Si prova quindi ad aumentare il numero di server che costituiscono il centro 1, portandolo a 3. Con la configurazione (3,1,1,1,1) si ottiene una utilizzazione del centro 1 che risulta 0.825925 ± 0.002701 , quindi < 1 . Con tale utilizzazione si ottiene un tempo di risposta pari a 230.791342 ± 4.102363 s, quindi circa 3.83 minuti, inferiori ai 5 min imposti dal QoS. Inoltre la probabilità di saltare i controlli presso il centro controllori risulta pari a 0.332199 ± 0.000714 s, quindi la percentuale di utenti che viene sottoposta a controlli è circa pari al 67 %, maggiore del 30% imposto dai QoS. Infine anche il tempo di attesa in coda presso gli ascensori, con un solo ascensore attivo, data la poca affluenza permette di rispettare il QoS definito, con un tempo in coda pari a 0.412860 ± 0.021548 s. Questi risultati sono mostrati nelle figure sottostanti, in cui si evidenzia anche il conseguimento da parte del sistema della stazionarietà.



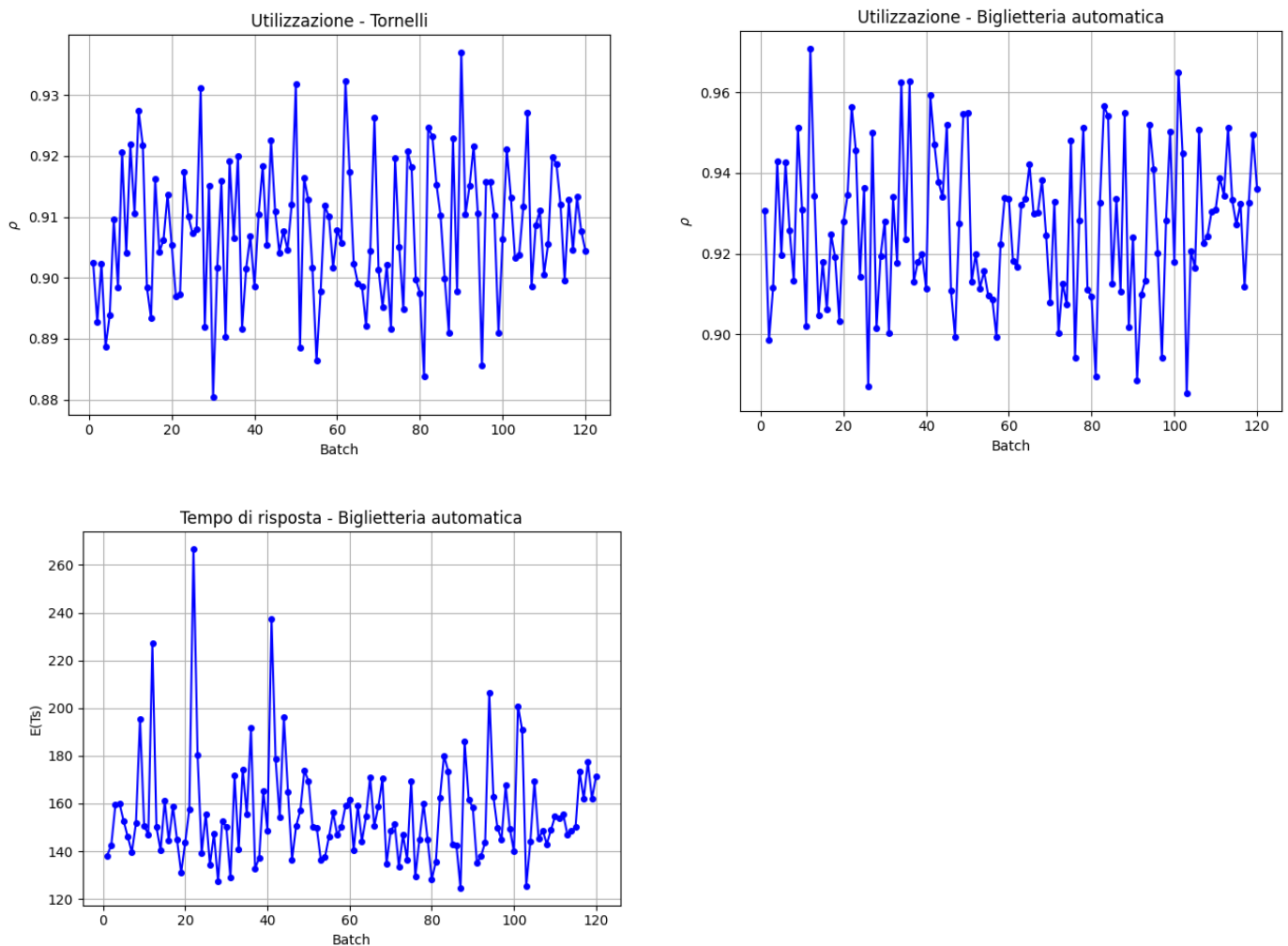
Con questa configurazione (che risulta quindi essere ottimale, minimizzando il numero di serventi utilizzati) il tempo medio per attraversare il sistema risulta essere dalla simulazione pari a $135.720395 \pm 2.116704 \text{ s} \approx 2.26 \text{ min}$, quindi di molto inferiore ai 6 min imposti dal QoS. Si sottolinea come questo tempo medio di risposta consideri tutte le tipologie di utenti, quindi sia i possessori di abbonamento (i cui tempi per raggiungere la banchina saranno minori rispetto a chi deve acquistare il biglietto), sia coloro che necessitano dell'acquisto di un biglietto, e prende inoltre in considerazione le visite medie presso i diversi centri secondo le probabilità di routing definite e ottenute a runtime in funzione della configurazione desiderata (come accade per la probabilità di skip, oppure le probabilità di routing verso le diverse code del centro tornelli, qui non visibili in quanto la configurazione ottimale prevede un solo tornello attivo). Proprio al fine di caratterizzare il comportamento di una certa tipologia di utenti, ossia coloro che acquistano il biglietto, si è deciso di considerare anche l'ulteriore QoS sul tempo di risposta del singolo centro biglietteria.

Fascia 2 (7.30-10.30):

Questa fascia oraria rappresenta un periodo di elevata affluenza presso la stazione ($\lambda = 0.303 \text{ utenti/s}$, il secondo tra i tassi più alti), dato che coincide con orari di ingresso a scuola e negli uffici del centro rispettivamente per studenti e lavoratori. Si ipotizza quindi che una configurazione (3,1,1,1,1) come quella adottata per la fascia 1 non sia adeguata in questo caso, e ciò è infatti osservabile mediante la simulazione, che dimostra come il centro 1 e 2 presentino utilizzazioni molto elevate (>1), che portano i tempi di risposta a crescere man mano che prosegue la simulazione stessa.



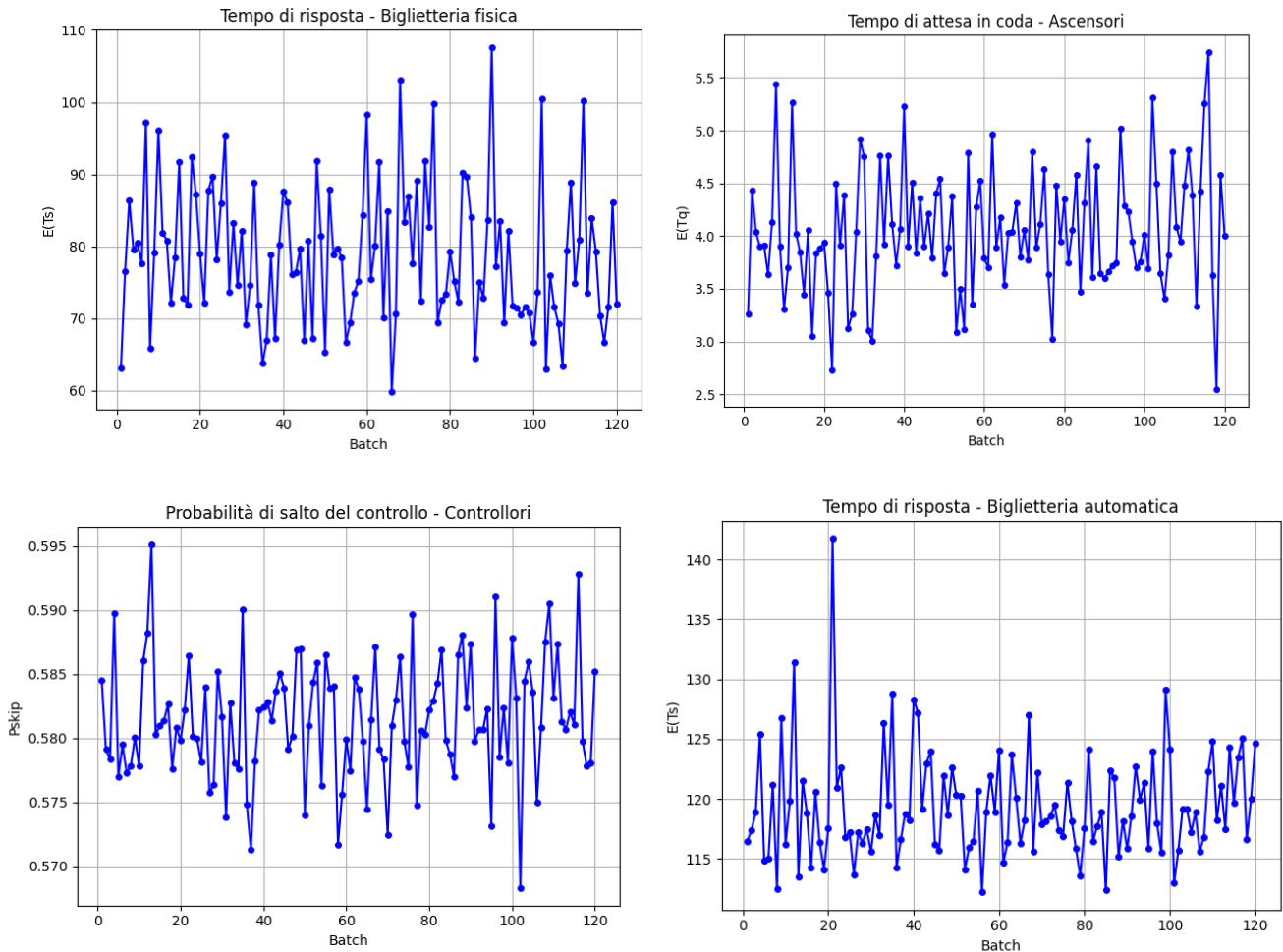
Si simula quindi una configurazione che aumenti il numero di server nei centri 1 e 2. Con la configurazione (18,3,2,1,1) si riesce ad ottenere un tempo di risposta presso la biglietteria automatica di 155.921444 ± 3.936240 s, che rispetta quindi i QoS, così come il tempo di attesa in coda presso gli ascensori che risulta 4.044928 ± 0.109703 s. La probabilità di salto dei controllori è però troppo elevata, pari a 0.778038 ± 0.000713 s, quindi maggiore di 0.7 imposto dai QoS. Oltre a dover abbassare questa probabilità, vi è in realtà un problema inerente ai centri della biglietteria automatica e dei tornelli, poichè nonostante i QoS siano rispettati, le utilizzazioni di questi centri risultano molto elevate ($\rightarrow 1$), perciò piccole perturbazioni del sistema porterebbero i tempi di questi centri ad esplodere, rendendo poco robusta la configurazione adottata. I grafici correlati a queste considerazioni sono mostrati qui sotto.



Per tale motivo si realizza un incremento di server al centro controllori, ma anche presso i centri della biglietteria e dei tornelli, ottenendo quindi la configurazione (20,3,3,2,1). In questo momento le utilizzazioni raggiungono valori accettabili, 0.835353 ± 0.003386 per la biglietteria automatica e 0.605343 ± 0.001463 per i tornelli. Tutti i QoS sono rispettati, infatti:

- $E(T_s) = 119.383099 \pm 0.799065$ s, per la biglietteria automatica
- $E(T_q) = 4.038864 \pm 0.105574$ s, per gli ascensori
- $p_{\text{skip}} = 0.581497 \pm 0.000845$, per il centro dei controllori

Si riportano i grafici della simulazione qui analizzata.



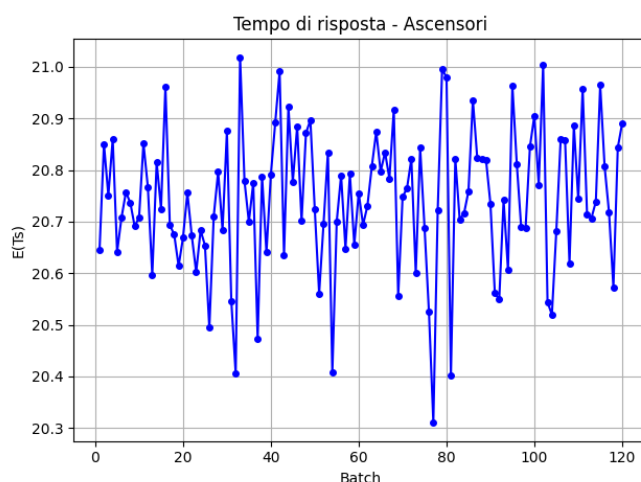
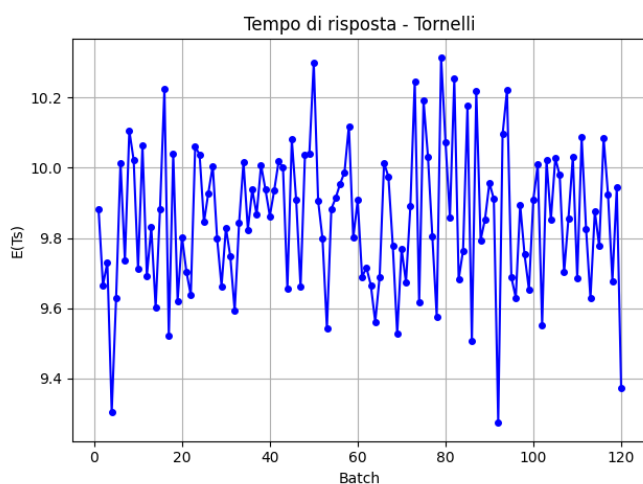
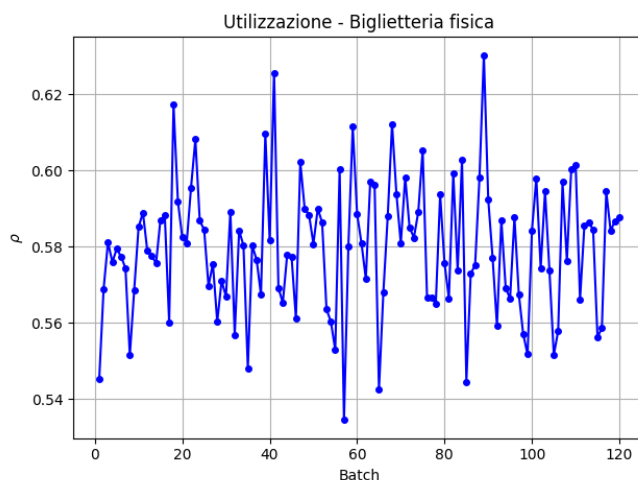
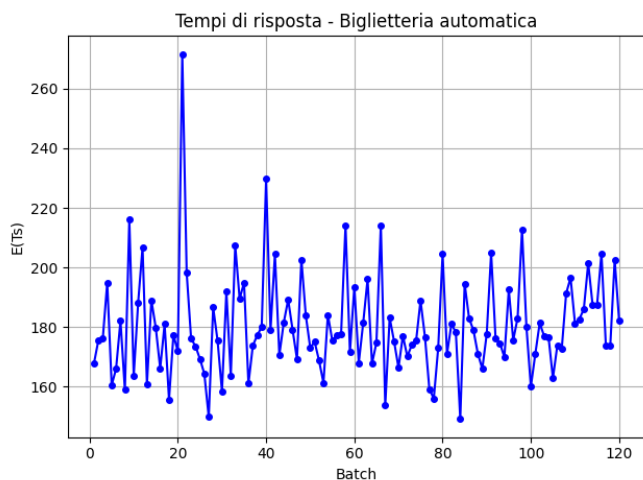
In conclusione, dalla simulazione risulta che il tempo medio di risposta per l'intero sistema è di $79.333859 \pm 0.517075 \text{ s} \approx 1.3 \text{ min}$.

Fascia 3 (10.30-14.30):

Questa fascia oraria presenta un'affluenza inferiore ($\lambda = 0.076 \text{ utenti/s}$), simile alla fascia 1, e per tale motivo le considerazioni fatte in quella fascia sono il punto di partenza per la configurazione di questa. Si ottiene che la configurazione (5,1,1,1,1) permette di rispettare tutti i Qos fissati, con utilizzazioni contenute, che si attestano al più su 0.8, come nel caso della biglietteria automatica, e inferiori per gli altri centri. Rispetto alla fascia 1 risulta necessario un incremento di server presso il centro 1, dato l'aumento del flusso in ingresso. Non risultano invece necessari aumenti nel numero di server negli altri centri (rispetto alla fascia 1) poiché i tempi di servizio ridotti permettono comunque di avere tempi di attesa, e quindi risposta, contenuti nonostante l'aumento del flusso rispetto alla fascia 1. Inerentemente ai tempi su cui sono stati fissati i Qos, si ottiene:

- $E(T_s) = 180.355857 \pm 3.085338 \text{ s}$, per la biglietteria automatica
- $E(T_q) = 0.740256 \pm 0.024935 \text{ s}$, per gli ascensori
- $p_{skip} = 0.459887 \pm 0.000737$, per il centro dei controllori.

Le performance di alcuni centri del sistema sono descritte mediante i grafici qui riportati:



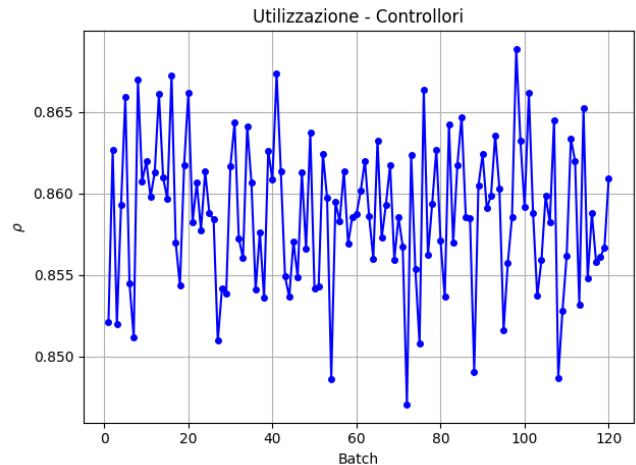
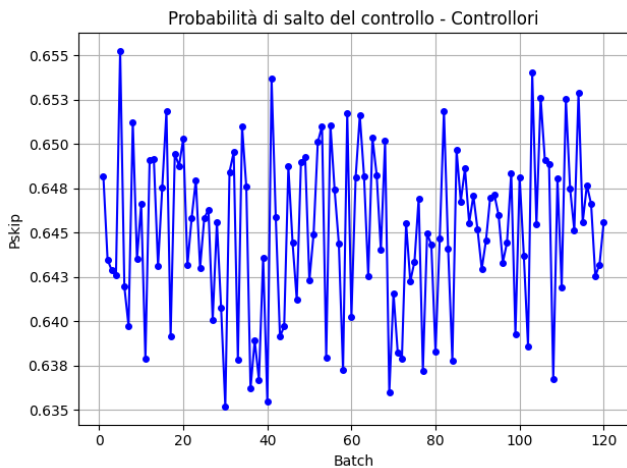
Il tempo medio di risposta per l'intero sistema risulta pari a 114.743192 ± 1.613079 s $\approx 1,9$ min.

Fascia 4 (14.30-15.30):

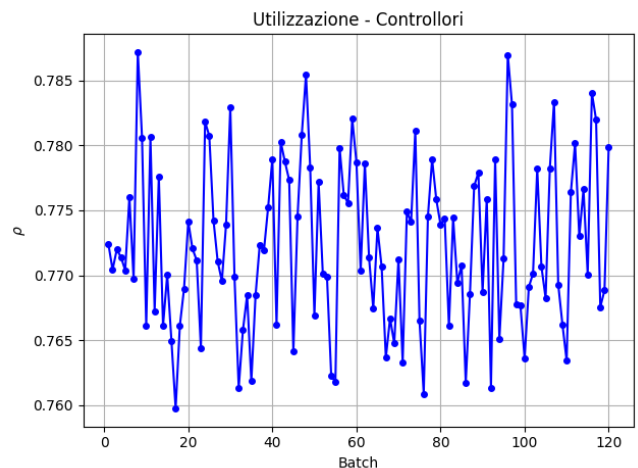
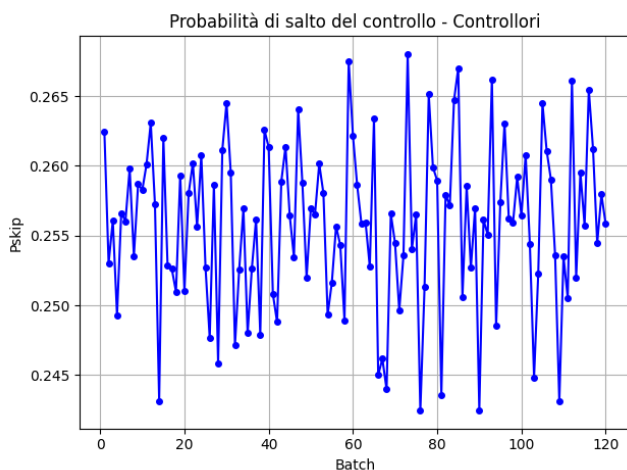
Questa fascia rappresenta il picco di affluenza del sistema ($\lambda = 0.606$ utenti/s), poichè un elevato numero di passeggeri si reca presso la stazione in un lasso di tempo relativamente breve, e deve essere quindi gestita adeguatamente per evitare sovraccarichi e code eccessivamente lunghe. Le considerazioni fatte per la fascia 2 rappresentano il punto di partenza per individuare la configurazione ottimale di questa fascia, essendo la fascia 2 quella con affluenza maggiore incontrata fino ad ora. Ovviamente la configurazione (20,3,3,2,1) della fascia 2 risulta inadeguata, poichè i centri 1 e 2 presentano code che tendono a crescere all'infinito, portando i tempi di risposta a divergere. Si aumenta quindi il numero di server in questi primi due centri, ottenendo la configurazione (42,6,5,3,1), dove si tenuto in conto, oltre dei QoA, anche dell'utilizzazione, cercando di mantenerla su un valore inferiore a 0.8, per evitare che perturbazioni del sistema portassero a divergere un qualche centro, rendendo la configurazione più robusta alle variazioni

del traffico. Anche nei centri controllori e tornelli è stato necessario aumentare il numero di server, dato l'aumento del flusso in ingresso. Il sistema presenta le seguenti performance:

- $E(T_s) = 111.068266 \pm 0.284599$ s, per la biglietteria automatica
- $E(T_s) = 57.801163 \pm 0.838264$ s, per la biglietteria fisica
- $E(T_s) = 8.228384 \pm 0.046134$ s, per i tornelli
- $E(T_s) = 71.327320 \pm 0.219346$ s, sull'intero sistema
- $E(T_q) = 14.067093 \pm 0.448360$ s, per gli ascensori
- $p_{\text{skip}} = 0.645082 \pm 0.000844$, per il centro dei controllori.



Tutti i QoS risultano rispettati. Sebbene p_{skip} risulta inferiore a 0.7, come desiderato, l'utilizzazione del centro controllori risulta piuttosto elevata, di valore pari a 0.858760 ± 0.000806 . Si ritiene quindi opportuno un aumento del numero di controllori, che sebbene non risulti necessario per rispettare il QoS su p_{skip} , è però utile a rendere il centro più robusto rispetto a variazioni che potrebbe subire (ad esempio nei tempi di servizio richiesti). Si porta quindi la configurazione a (42,6,5,7,1), che permette di avere un'utilizzazione del centro controllori pari a 0.772356 ± 0.001155 e $p_{\text{skip}} = 0.255803 \pm 0.001066$, di molto inferiore al vincolo fissato (questi nuovi valori sono mostrati nei grafici sottostanti).



Fascia 5 (15.30-18.30):

Questa fascia presenta un tasso d'arrivo ($\lambda = 0.05$ utenti/s) simile a quella della fascia 1. Per tale motivo si simula una configurazione analoga pari a (3,1,1,1,1). Come intuibile, tale configurazione non fa divergere i centri, ma non permette di rispettare i Qos del sistema, infatti $E(T_s) = 422.519246 \pm 18.524979 \text{ s} \approx 7 \text{ min}$ nel centro biglietteria automatica, superiore ai 5 min fissati come obiettivo. Si sceglie quindi una configurazione che aumenti il numero di casse automatiche, portandole a 4. Con la configurazione (4,1,1,1,1) si ottengono le seguenti performance:

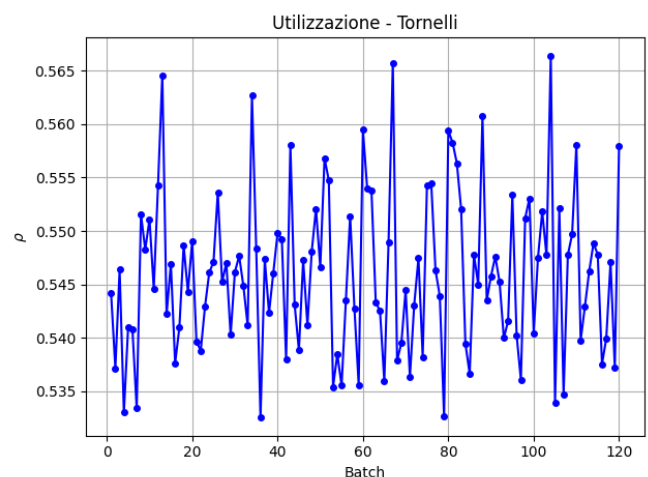
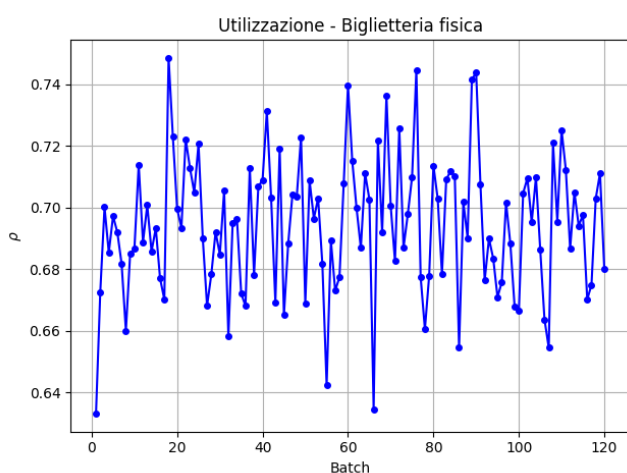
- $E(T_s) = 140.074140 \pm 0.881487 \text{ s}$, per la biglietteria automatica
- $E(T_s) = 69.647644 \pm 0.600114 \text{ s}$, per la biglietteria fisica
- $E(T_s) = 7.996500 \pm 0.020702 \text{ s}$, per i tornelli
- $E(T_s) = 90.582456 \pm 0.485420 \text{ s}$, sull'intero sistema
- $E(T_q) = 0.473915 \pm 0.023677 \text{ s}$, per gli ascensori
- $p_{\text{skip}} = 0.360262 \pm 0.000802$, per il centro dei controllori.

Tutti i Qos risultano rispettati.

Fascia 6 (18.30-21.30):

La seguente fascia oraria presenta un'affluenza ($\lambda = 0.273$ utenti/s) simile alla fascia 2. Per tale motivo si inizia la simulazione provando la medesima configurazione (20,3,3,2,1), che però dovrebbe intuitivamente risultare sovradimensionata, essendo l'affluenza, seppur di poco, inferiore. Infatti con tale configurazione il sistema presenta tempi decisamente contenuti, e utilizzazioni ridotte:

- $E(T_s) = 113.000641 \pm 0.353388 \text{ s}$, per la biglietteria automatica
- $E(T_s) = 65.647659 \pm 0.904982 \text{ s}$, per la biglietteria fisica
- $E(T_s) = 7.523524 \pm 0.027906 \text{ s}$, per i tornelli
- $E(T_s) = 74.230399 \pm 0.254817 \text{ s}$, sull'intero sistema
- $E(T_q) = 3.519824 \pm 0.084990 \text{ s}$, per gli ascensori
- $p_{\text{skip}} = 0.550768 \pm 0.000937$, per il centro dei controllori.



Rispetto alla fascia 2, con la medesima configurazione, essendo inferiore il tasso d'ingresso, le performance sono tutte migliorate. Analizziamo se è possibile minimizzare il numero di serventi, continuando a rispettare i QoS. Si prova a ridurre di uno il numero di serventi in ogni centro, ottenendo la configurazione (19,2,2,1,1).

Con tale configurazione però la simulazione mostra come il centro delle casse fisiche produca tempi di risposta che crescono all'infinito ($\rho > 1$), che comprova la divergenza del tempo di risposta dell'intero sistema, e il vincolo sulla probabilità di salto del controllo nel centro controllori non è più rispettato essendo pari a 0.758782 +/- 0.000653. L'utilizzazione del centro tornelli è aumentata (data la riduzione del numero di server), ma si attesta su un valore prossimo a 0.8, quindi ritenuto accettabile (0.812723 +/- 0.001685). Consideriamo quindi nuovamente un aumento di server nel centro casse fisiche e in quello dei controllori, ottenendo la configurazione ottimale (19,3,2,2,1). Tutti i QoS risultano rispettati:

- $E(T_s) = 115.700820 \pm 0.489871$ s, per la biglietteria automatica
- $E(T_s) = 65.647465 \pm 0.939146$ s, per la biglietteria fisica
- $E(T_s) = 16.722729 \pm 0.211698$ s, per i tornelli
- $E(T_s) = 75.659122 \pm 0.314449$ s, sull'intero sistema
- $E(T_q) = 3.410764 \pm 0.093318$ s, per gli ascensori
- $\rho_{\text{skip}} = 0.543799 \pm 0.000894$, per il centro dei controllori.

Fascia 7 (21.30-23.30):

Questa fascia presenta il medesimo tasso di ingresso della fascia 3 ($\lambda = 0.076$ utenti/s), perciò le medesime considerazioni valgono anche in questo caso, con conseguente configurazione ottimale pari a (5,1,1,1,1).

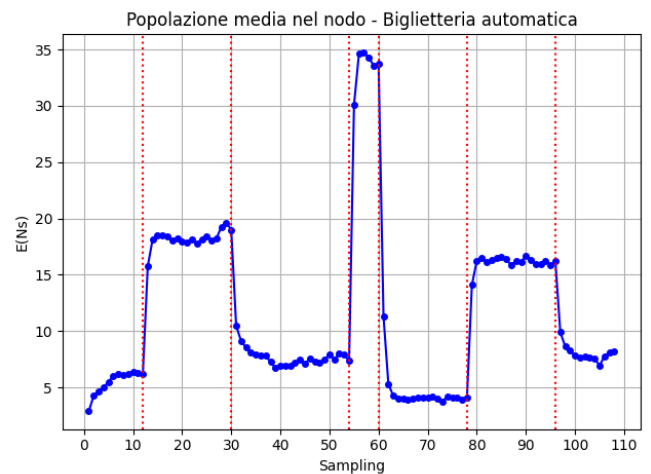
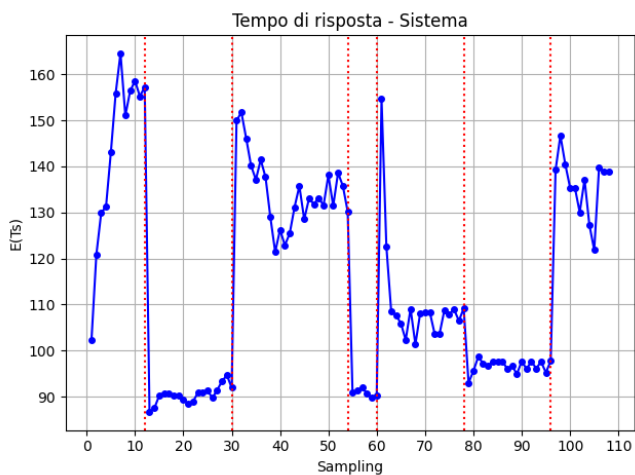
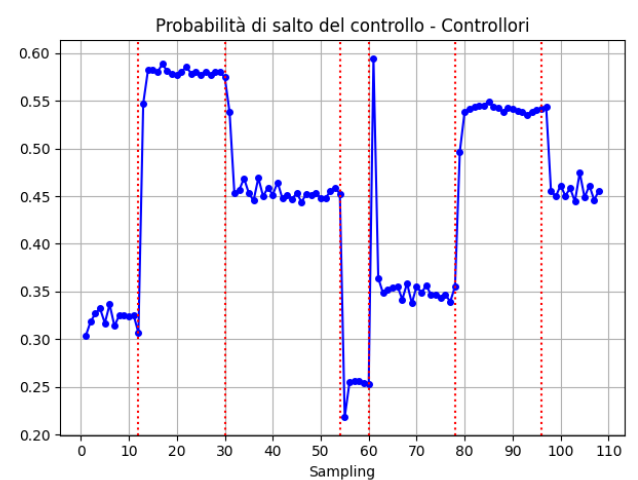
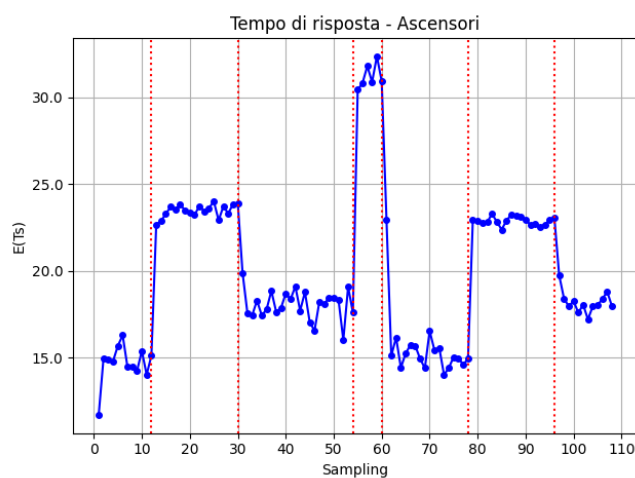
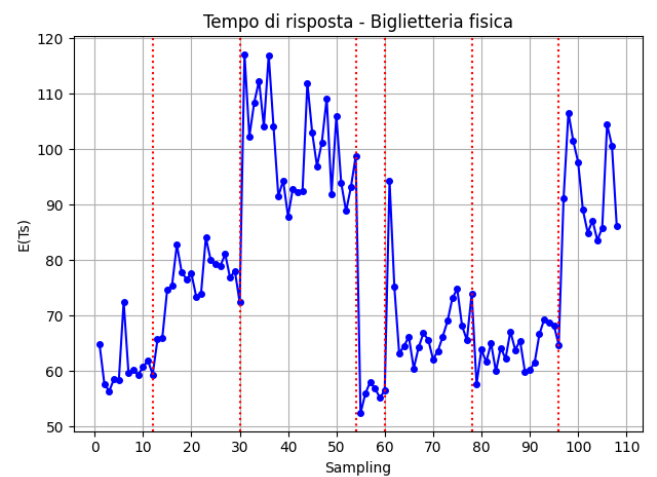
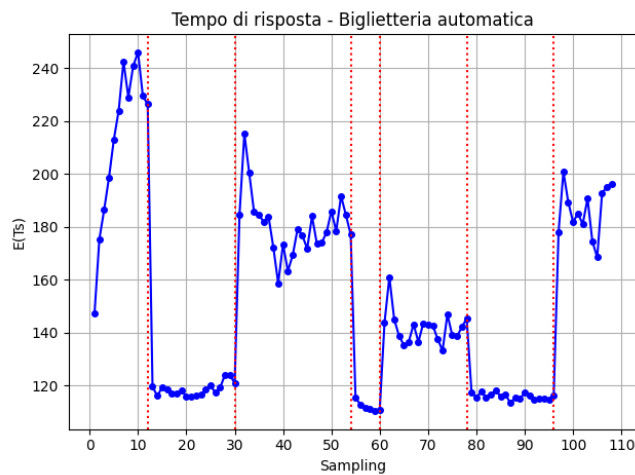
Conclusioni:

In conclusione, le configurazioni ottimali per ogni fascia oraria, che minimizzano i server utilizzati rispettando i QoS fissati risultano:

- fascia 1 \rightarrow (3,1,1,1,1)
- fascia 2 \rightarrow (20,3,3,2,1)
- fascia 3 \rightarrow (5,1,1,1,1)
- fascia 4 \rightarrow (42,6,5,7,1)
- fascia 5 \rightarrow (4,1,1,1,1)
- fascia 6 \rightarrow (19,3,2,2,1)
- fascia 7 \rightarrow (5,1,1,1,1)

9.4 Simulazione a orizzonte finito

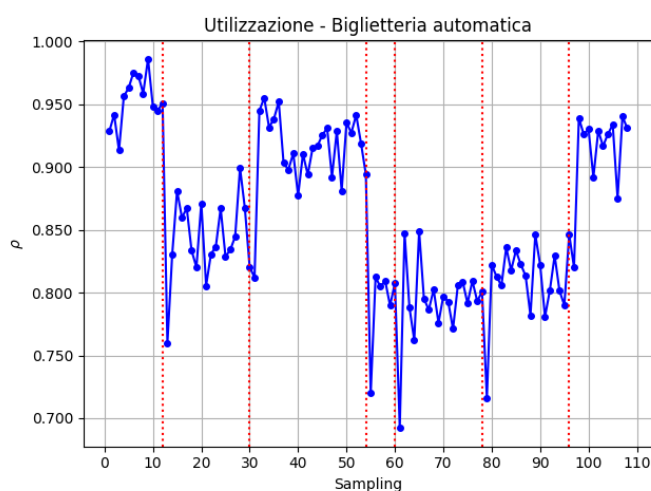
Alla luce dello studio del sistema realizzato con la simulazione a orizzonte infinito, utilizziamo le configurazioni ottimali individuate in quel caso per analizzare come si comporta il sistema simulandolo sull'effettivo tempo di operatività di 18 ore. Si esegue quindi la simulazione all'orizzonte finito utilizzando le configurazioni ottenute nella sezione precedente. I risultati sono mostrati nei grafici sottostanti, che mostrano l'andamento di varie statistiche dei centri in ogni slot di campionamento, ottenute mediando sulle 128 repliche eseguite della simulazione. Le linee rosse verticali evidenziano i cambiamenti di fascia oraria.



Di conseguenza risultano i seguenti intervalli di confidenza per i tempi di risposta dei vari centri:

Centro	$E(Ts)$
1. Biglietteria automatica	153.365615 +/- 7.099477
2. Biglietteria fisica	77.384506 +/- 3.247784
3. Tornelli	10.454418 +/- 0.924967
4. Controllori	12.114785 +/- 0.034938
5. Ascensori	19.709665 +/- 0.833470

I grafici evidenziano come il comportamento transiente del sistema in ogni fascia oraria sia quindi coerente con l'andamento alla stazionarietà studiato nella sezione precedente, ed inoltre la configurazione simulata permetta di rispettare tutti i QoS prefissati. La simulazione all'orizzonte finito ci permette però di fare delle osservazioni e giungere a delle conclusioni che la simulazione all'orizzonte infinito non può mettere in luce. Il comportamento "conservativo" messo in atto durante la simulazione all'orizzonte infinito, nelle fasce orarie caratterizzate dall'affluenza maggiore, nel tentativo di ridurre il più possibile l'utilizzazione dei centri su cui verte la domanda di servizio maggiore (ossia le biglietterie, come mostrato anche dall'analisi del collo di bottiglia), ha portato ad avere tempi di risposta inferiore proprio in queste fasce orarie, dato l'elevato numero di serventi messi a disposizione. Ciò è mostrato anche dal grafico sull'andamento dell'utilizzazione mostrato qui sotto.

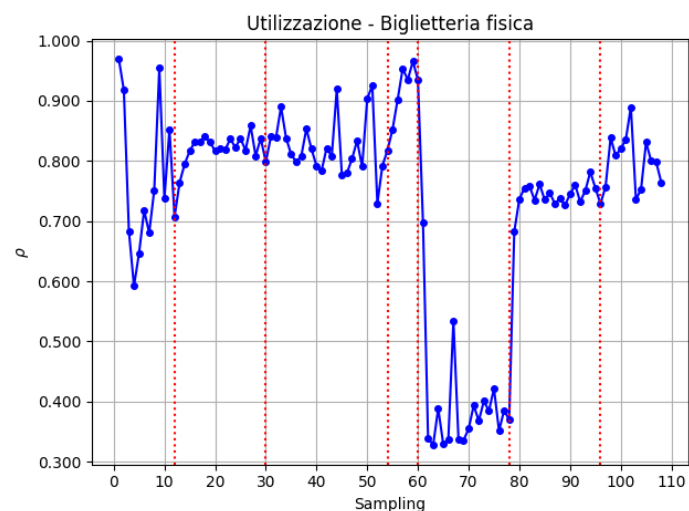
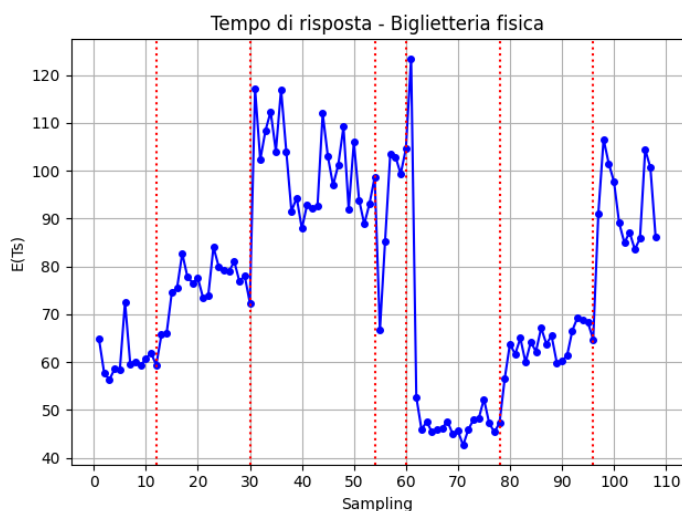
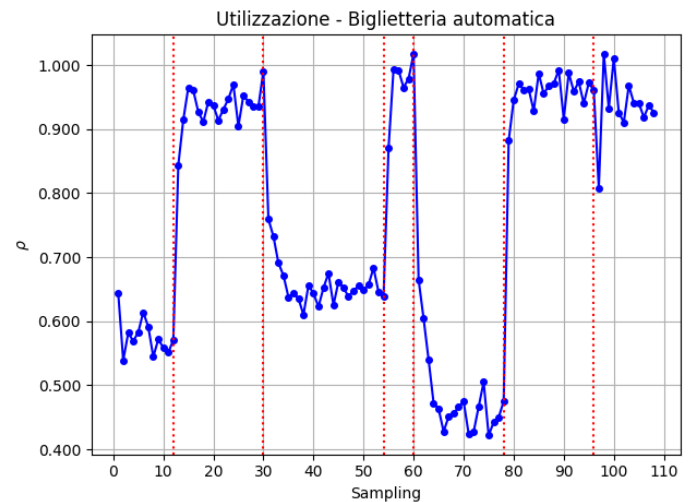
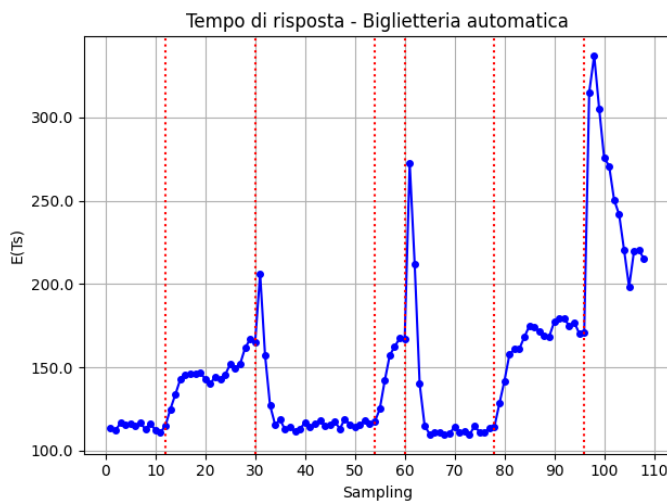


Risulta infatti che le fasce orarie ad affluenza maggiore sono anche quelle in cui si è adottata una configurazione che permettesse di avere un'utilizzazione minore, quindi tempi di risposta minori, dato che aumenta la probabilità per gli utenti in arrivo presso il centro di trovare un servente libero. Queste fasce orarie hanno però una durata limitata, si tratta infatti di picchi di uno o due ore, seguiti da periodi di affluenza molto ridotta. Si vuole quindi studiare se, con una configurazione che riduca nelle ore di punta il numero di serventi, ma aumenti leggermente questi ultimi nelle fasce con meno affluenza, sia possibile comunque rispettare i QoS fissati. L'idea è che il carico che un centro può accumulare nella fascia precedente, può essere smaltito nella fascia successiva, comunque ottenendo dei tempi consoni ai QoS.

Si considera quindi la seguente configurazione delle fasce orarie:

- fascia 1 $\rightarrow (5,1,1,1,1)$
- fascia 2 $\rightarrow (18,3,3,2,1)$
- fascia 3 $\rightarrow (7,1,1,1,1)$
- fascia 4 $\rightarrow (34,5,5,7,1)$
- fascia 5 $\rightarrow (7,2,1,1,1)$
- fascia 6 $\rightarrow (16,3,2,2,1)$
- fascia 7 $\rightarrow (5,1,1,1,1)$

Si ottengono i seguenti andamenti per tempi di risposta e utilizzazioni delle biglietterie:



Queste nuove configurazioni permettono di avere una riduzione non indifferente sul numero di server adottati nelle biglietterie (8 casse automatiche in meno da mantenere attive nella fascia 4, e una cassa fisica in meno nella fascia 4). Tale risultato si è ottenuto aumentando leggermente il numero di server di cui i centri dispongono nelle fasce successive alle fasce orarie con il picco di utenti. Questo comporta un'utilizzazione maggiore nelle fasce orarie di picco, con tempi di risposta leggermente maggiori, ma sempre rispettosi dei QoS. Il numero di utenti in coda che si accumula in queste fasce con $\rho \rightarrow 1$ riesce ad essere adeguatamente smaltito nel momento in cui avviene il cambiamento di fascia oraria, e quindi una notevole riduzione del tasso di arrivo presso la stazione. Tale configurazione è considerata vantaggiosa rispetto a quella proposta inizialmente, poichè le casse automatiche o fisiche a disposizione della stazione sono in numero prefissato. La scelta risiede nel decidere quante casse mantenere attive in ogni fascia oraria, cercando di diminuire il numero al fine di risparmiare sui costi della stazione. Poter ridurre di 8 server il centro della biglietteria automatica nella fascia 4, con solo un aumento di 2 o 3 casse nelle fasce adiacenti, risulta quindi decisamente vantaggioso. Ragionamento analogo vale per le casse fisiche. I nuovi tempi di risposta per questi centri ammettono quindi i seguenti nuovi valori medi:

- $E(T_s) = 150.291980 \pm 9.274812$ s, per la biglietteria automatica s

- $E(T_s) = 76.569082 \pm 3.975168$ s, per la biglietteria fisica s
- $E(T_s) = 114.251759 \pm 6.171507$ s, sull'intero sistema (quindi per un generico utente, possessore di abbonamento, che si reca alla biglietteria automatica o fisica).

I QoS continuano ad essere rispettati.

Si sottolinea inoltre come in tutte le simulazioni eseguite sia sempre stato rispettato il vincolo sulla capacità della banchina imposto inizialmente. Ne consegue che il dimensionamento dei vari centri proposto risulta coerente con la frequenza di transito dei treni, in una giornata di normale operatività della stazione (senza quindi eccessive attese per il transito dei treni dovute a disservizi o guasti).

9.5 Conclusioni

Lo studio di simulazione condotto permette di concludere che le biglietterie rappresentano l'elemento critico nella configurazione della stazione. Tale risultato è coerente con quello dato dall'esperienza personale, che evidenzia come spesso le casse dove si acquistano i biglietti sono caratterizzate da lunghe code, poiché in numero inferiore rispetto a quelle necessarie per una stazione con elevata affluenza come San Giovanni. La disponibilità dei tornelli, se funzionanti in numero sufficiente, risulta invece adeguata. Lo stesso vale per gli ascensori, dato il flusso contenuto in ingresso a questo centro. Anche il quesito posto all'inizio dello studio sulla presenza costante di controllori posti nella stazione trova qui una risposta, infatti, se presenti in numero adeguato (che non risulta troppo elevato), questi permettono di limitare notevolmente il problema dell'utilizzo del trasporto pubblico senza il possesso di un titolo di viaggio valido, senza avere un impatto negativo eccessivo sui tempi sperimentati dai passeggeri.

Fonti

- (1) https://www.comune.roma.it/web-resources/cms/documents/09_Mobilita_Annuario2022.pdf
- (2) <http://www.ruggerolenci.it/Didattica/UNI7508.htm>
- (3) <https://web.archive.org/web/20090821083704/http://www.metroroma.it/IT-IT/AZIENDA/DESCRIZIONE/Pagine/I%20numeri%20di%20metro.aspx>
- (4) <https://romamobilita.it/it/metro-c/treni-centrale-operativa>
- (5) Lawrence Leemis, Stephen Park. *Discrete-Event Simulation: A first course*. Pearson, 1st edition (January 6, 2006).