UNIVERSITÀ DEGLI STUDI DI ROMA TOR VERGATA

MACROAREA DI INGEGNERIA

CORSO DI STUDIO IN
Ingegneria Informatica

Progetto di

Performance Modeling Of Computer Systems And Networks

Relazione
Modellazione Ufficio Postale

matricola: *0350265*
Alessia Cinelli

matricola: *0350270*
Andrea Tozzi

Anno Accademico 2023/2024

Sommario

Introduzione 4
Presentazione del sistema 4
1. Obiettivi e Scopi 6
2. Modello Concettuale 7
2.1 Gestione code 8
2.2 Stato del Sistema 9
3. Modello delle Specifiche 10
3.1 Assunzioni del modello: 11
3.2 Valori numerici utilizzati: 11
4. Modello Computazionale 13
4.1 Stato del Sistema 13
4.2 Descrizione degli Eventi 14
4.3 Clock di Simulazione 15
5. Verifica 16
5.1 Confronto con medie teoriche 16
6. Validazione 22
7. Design degli esperimenti 24
7.1 Analisi del transitorio 24
7.2 Analisi dello Stazionario 28
7.3 Utilizzazione del sistema 29
7.4 Obiettivi studiati 29
Modello Migliorativo 30
1. Obiettivi e Scopi 31
2. Modello Concettuale 31
3. Modello delle specifiche 33
4. Modello Computazionale 33
5. Verifica 33
6. Validazione 34
7. Design degli esperimenti 35

7.1 Analisi del transitorio 35
7.1.1 Primo Miglioramento 35
7.1.2 Secondo Miglioramento 37
7.3 Analisi stazionaria 39
7.3.1 Primo miglioramento 39
7.3.2 Secondo miglioramento 40
7.4 Analisi degli obiettivi 41
7.5 Utilizzazioni del sistema 42
7.5.1 Primo Miglioramento 42
7.5.2 Secondo Miglioramento 42
7.6 Sovraccarico di lavoro 43
7.6.1 Primo Miglioramento 43
7.6.2 Secondo Miglioramento 44
8. Conclusioni 45

Introduzione

In questo progetto presentiamo una simulazione sulla gestione delle code relativo all'Ufficio Postale Frosinone 3.

Inizialmente, in assenza di dati, statistiche, o in generale informazioni ufficiali, abbiamo deciso di visitare di persona la sede al fine di osservare direttamente i flussi di clienti e le dinamiche interne, quali numero di sportelli e tipologia di servizi offerti.

Questo ci ha permesso di raccogliere dati utili per la nostra simulazione, come il numero di persone in attesa e i tempi di servizio. Durante la nostra visita, la direttrice dell'ufficio postale ci ha fornito veloci informazioni su un QoS (Quality of Service) riguardante il tempo medio in coda, che, a suo dire, non viene sempre soddisfatto.

Successivamente, tornando in sede per ulteriori domande, ci è stata presentata l'opportunità di ottenere dati ufficiali della filiale, previa autorizzazione della sede centrale di Roma. Abbiamo quindi redatto una richiesta formale per ottenere tali informazioni, che tuttavia non sono ancora stati ricevuti.

Pertanto, per svolgere il seguente progetto abbiamo utilizzato i dati raccolti autonomamente durante le nostre visite.

Presentazione del sistema

Il sistema valutato prevede tre tipologie di Operazioni che i clienti possono effettuare:

1. *Operazione Generic*a
2. *Spedizioni e Ritiri*
3. *Versamenti e Prelievi* (Tramite sportello o con ATM)

Per Operazioni Generiche, si intendono tutte le operazioni svolgibili in un ufficio postale, ad esempio: Apertura di un libretto, pagamento di una bolletta, creazioni di buoni postali, SPID, ecc.

Per Spedizioni e Ritiri, si intendono le sole operazioni di spedizione e ritiro Pacchi.

Per le Operazioni di Versamenti e Prelievi, si intendono le sole operazioni effettuabili se si ha un libretto postale attivo.

L'ufficio postale dispone di un unico sportello dedicato alle operazioni di Spedizioni e Ritiri poiché esso è dotato di uno slot per inserire i pacchi e una bilancia per pesarli.

Per entrambe le altre operazioni descritte: Operazioni Generiche e Versamenti e Prelievi sono disponibili in totale cinque sportelli.

I clienti che devono effettuare operazioni di Versamenti e Prelievi possono rivolgersi sia in uno dei cinque sportelli dedicati o, in alternativa, possono utilizzare l'ATM.

Per semplicità, l'operazione di Versamenti e Prelievi svolta presso lo sportello sarà inclusa in Operazione Generica.

Sono previste inoltre tre tipologie di utenti:

1. Clienti che hanno effettuato una prenotazione online.
2. Clienti con difficoltà.
3. Clienti senza difficoltà.

È possibile, tramite App o Sito Online, prenotare un appuntamento allo sportello dell'Ufficio Postale, specificando l'orario in cui si vuole prendere appuntamento e l'operazione da svolgere. La prenotazione avviene a intervalli di sette minuti, in cui ogni appuntamento è riservato esclusivamente a un solo utente. Gli appuntamenti possono essere fissati solo a partire dall'orario attuale con una disponibilità che inizia dalla mezz'ora successiva; pertanto, non è possibile prenotare un appuntamento per l'orario corrente o entro la prossima mezz'ora.

Si tiene conto l'impossibilità di prenotare un appuntamento per Versamenti e Prelievi svolti presso l'ATM.

Viene quindi assegnata priorità maggiore nella gestione delle code, agli utenti con prenotazione, poiché dispongono di un orario preciso per essere serviti. Successivamente, la priorità è data ai clienti con difficoltà, come da politica aziendale, seguiti in fine dagli altri clienti.

La filiale è aperta ogni giorno per 4 ore e dopo l'orario di chiusura continua a servire i clienti rimasti in coda, fino a completo svuotamento.

Lo scheduling utilizzato per ogni coda è: FIFO, senza possibilità di interrompere un'operazione per incominciarne un'altra.

I seguenti dati: numero di sportelli e intervalli di prenotazione in app sono dati veritieri relativi all'ufficio postale preso in considerazione.

1. Obiettivi e Scopi

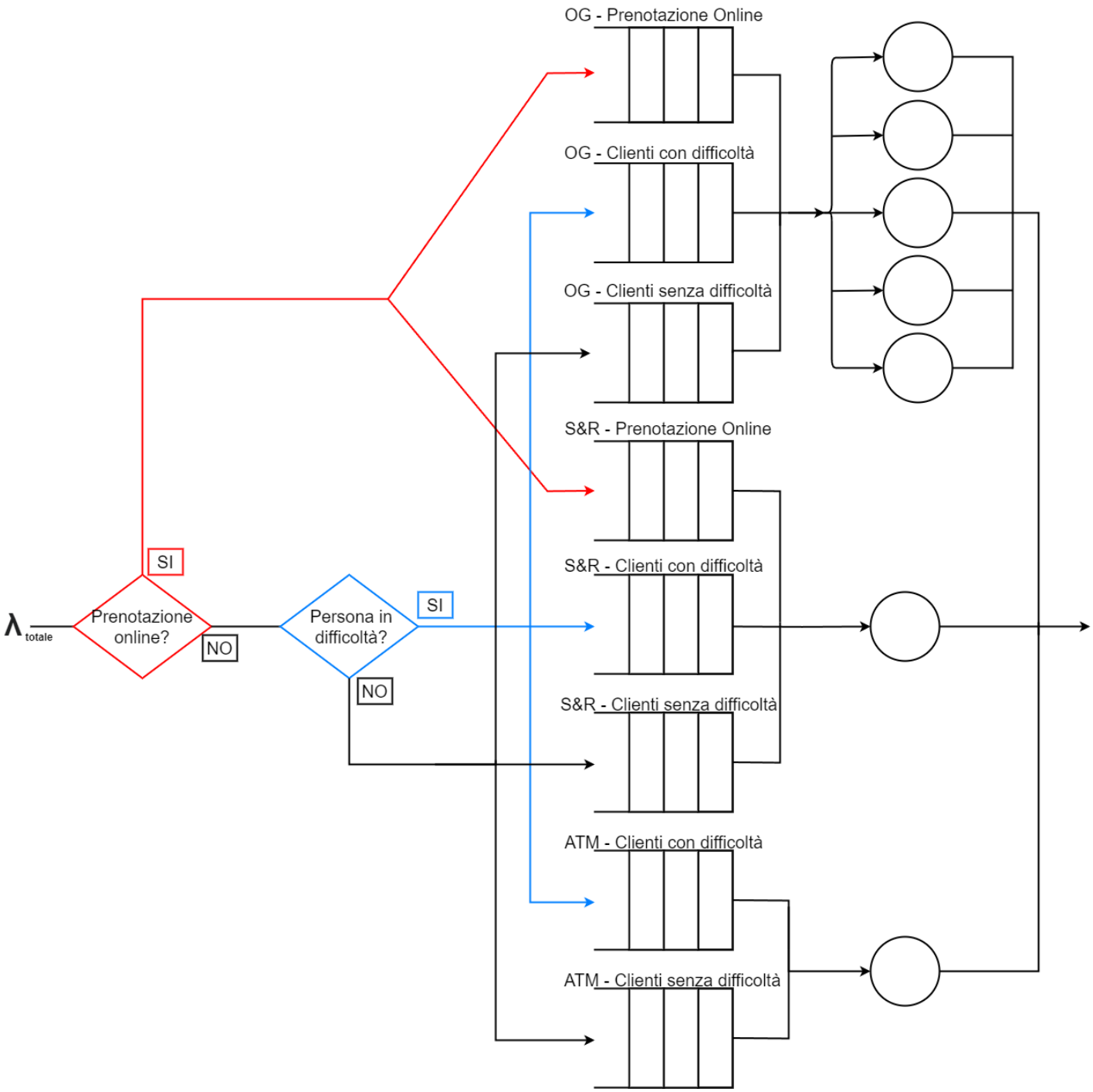
Lo scopo principale di questo studio è ridurre i tempi di attesa per i clienti presso l'ufficio postale di Frosinone 3, garantendo il rispetto, ove possibile, dei livelli di servizio (QoS - Quality of Service) stabiliti dall'azienda.

Obiettivi:

1) **Ottenere un tempo medio di coda minore di 15 minuti:** Come da QoS dell'azienda, si vuole ottenere un tempo medio di attesa in coda per ciascuna classe di utenza inferiore a 15 minuti.

2) **Limitare il sovraccarico di lavoro per i dipendenti:** Ridurre il tempo medio di sovraccarico (minuti trascorsi dopo le 4 ore di giornata lavorativa) sotto i 30 minuti, garantendo così condizioni di lavoro sostenibili.

2. Modello Concettuale



2.1 Gestione code

Sono state considerate tante code quante sono le classi di utenza:

Per i clienti che intendono effettuare una Operazione Generica attendono di essere serviti da uno dei cinque sportelli disponibili, l'ordinamento di priorità è:

1. Operazioni Generiche (OG) – Utenti con Prenotazioni
2. Operazioni Generiche (OG) – Utenti con Difficoltà
3. Operazioni Generiche (OG) – Utenti senza Difficoltà

Per i clienti che intendono effettuare Spedizioni e Ritiri attendono di essere serviti dall'unico sportello disponibile, l'ordinamento di priorità è:

1. Spedizioni e Ritiri (SR) – Utenti con Prenotazioni
2. Spedizioni e Ritiri (SR) – Utenti con Difficoltà
3. Spedizioni e Ritiri (SR) – Utenti senza Difficoltà

Per i clienti che intendono effettuare Versamenti e Prelievi presso l'ATM attendono presso lo sportello automatico (ATM), l'ordinamento di priorità è:

1. ATM – Utenti con Difficoltà
2. ATM – Utenti senza Difficoltà

Chiameremo nel corso della relazione le code associate alle classi di utenza anche in questo modo:

Classe d'utenza	Nome
Operazioni Generiche: Utenti con Prenotazioni	Coda 1
Operazioni Generiche: Utenti con Difficoltà	Coda 2
Operazioni Generiche: Utenti senza Difficoltà	Coda 3
Spedizioni e Ritiri: Utenti con Prenotazioni	Coda 4
Spedizioni e Ritiri: Utenti con Difficoltà	Coda 5
Spedizioni e Ritiri: Utenti senza Difficoltà	Coda 6
ATM: Utenti con Difficoltà	Coda 7
ATM: Utenti senza Difficoltà	Coda 8

2.2 Stato del Sistema

Ad ogni istante di tempo lo stato del sistema è univocamente determinato dalle seguenti variabili:

1. **Stato degli sportelli:**
   - **IDLE (Libero):** Lo sportello è libero e può accogliere un nuovo cliente.
   - **BUSY (Occupato):** Lo sportello è occupato nel servire un cliente.
2. **Numero di clienti in servizio, *per ciascuna classe di utenza*:**
   - Siamo interessati a conoscere quanti clienti sono attualmente in servizio presso uno degli sportelli, e in quale, per ciascuna classe di utenza.
3. **Numero di clienti in coda, *per ciascuna classe di utenza*:**
   - Siamo interessati a conoscere quanti clienti sono attualmente in coda, per ciascuna classe di utenza.

Osservazione:

Per ciascuna tipologia di Operazione tra le tre definite, si può osservare:

- Una politica di **work-conserving** (conservazione del lavoro), in cui un server (sportello) non rimane inattivo se c'è almeno un cliente in una coda a lui associata.
  Se c'è almeno uno sportello libero (IDLE), allora non devono esserci clienti nelle code da esso servite.

3. Modello delle Specifiche

Le variabili matematiche seguenti forniscono una rappresentazione univoca dello stato del sistema a livello di specifiche, per ogni istante di tempo:

- **Stato del k-esimo sportello:**
  Lo stato del k-esimo sportello in un dato momento "t" è rappresentato dalla variabile:
  $$Sportello_k(t) \in \{0,1\}$$
  dove 0 indica che lo sportello è disponibile (IDLE) e 1 che lo sportello è occupato (BUSY).

- **Numero di clienti in servizio:**
  Per ogni classe di utenza "c" relative alle Operazioni Generiche al tempo "t", il numero di clienti in servizio presso gli sportelli è rappresentato dalla variabile:

  $$InServizio_c(t) \in \{0, 1, \dots, 6\}$$

  Per le classi di utenza "c" relative alle operazioni di Spedizioni e Ritiri al tempo "t", il numero di client in servizio presso lo sportello è rappresentato dalla variabile

  $$InServizio_c(t) \in \{0, 1\}$$

  Numeri naturali limitati al numero di sportelli disponibili per quella classe di utenza.

- **Numero di clienti in coda:**

  Per ogni classe di utenza c al tempo "t", il numero di clienti in attesa nella coda è dato dalla variabile:

  $$Queue_c(t) \in \{0, 1, \dots\}$$

Relazioni:

- Dalla somma tra il numero totale di clienti presenti in servizio e il numero di clienti in coda otteniamo il numero di clienti nel sistema:

  $$InSistema_c(t) = Queue_c(t) + InServizio_c(t)$$

3.1 Assunzioni del modello:

Di seguito sono riportate le distribuzioni utilizzate:

- Gli arrivi dei clienti all'ufficio postale avvengono in modo casuale, il che può essere modellato tramite un processo di Poisson. Di conseguenza, i tempi tra un arrivo e l'altro (tempi di inter-arrivo) seguono una distribuzione esponenziale.

    - Distribuzione Esponenziale per i tempi di inter-arrivo.

- I clienti vengono serviti relativamente alla tipologia di operazione scelta, per questo motivo ciascuna di essa ha un tasso di servizio specifico. Si è scelto di modellare i tempi di servizio con una distribuzione normale troncata inferiormente perché, in molti casi, i tempi necessari per completare un'operazione tendono a concentrarsi attorno a un valore medio, con variazioni simmetriche e una minore probabilità di riscontrare tempi estremamente brevi o estremamente lunghi. Il troncamento inferiore è stato effettuato ad un valore di 10^{-6}, necessario per evitare tempi di servizio negativi.

    - Distribuzione Normale Troncata per i tempi di servizio.

3.2 Valori numerici utilizzati:

<u>Tassi in ingresso:</u>

Per i tassi di inter-arrivo, abbiamo preferito distinguere i tassi per le prenotazioni online e non, per questo motivo possiamo definire: $\boldsymbol{\lambda_{totale}} = \boldsymbol{\lambda} + \boldsymbol{\lambda_{online}}$

$\boldsymbol{\lambda}$ **:** $\frac{1}{1.5}$ job/min (Una persona circa ogni 40 secondi arriva senza prenotazione).

$\boldsymbol{\lambda_{online}}$ **:** 0.1 job/min (Una persona ogni 10 minuti).

Il tempo di inter-arrivo dei clienti che devono effettuare prenotazioni online è molto elevato, poiché è necessario accedere tramite SPID per completare la prenotazione e molte persone non sono a conoscenza di questa possibilità.

Da qui, è possibile ricavare le probabilità:

1. Probabilità Prenotazione Online: $P_{ON} = \frac{\lambda_{\text{online}}}{\lambda_{\text{totale}}} \cong 0.13$
2. Probabilità di non avere prenotazione Online: $P_{N_ON} = \frac{\boldsymbol{\lambda}}{\boldsymbol{\lambda_{totale}}} = 1 - P_{ON} \cong 0.87$

Probabilità di svolgere un determinato tipo di operazione:

In seguito, mostriamo con quali probabilità un cliente che entra in filiale senza prenotazione e con prenotazione effettuerà i vari servizi:

Clienti senza prenotazione		%
Operazione Generica	P_OG	0.45
Operazione S&R	P_SR	0.35
Operazione ATM	P_ATM	0.20

Clienti con prenotazione		%
Operazione Generica Online	P_ON_OG	0.60
Operazione S&R Online	P_ON_SR	0.40

Probabilità per clienti senza prenotazione:

- La probabilità che una operazione sia di tipo generica, data la sua specifica che include operazioni di vario genere, raggruppa un numero maggiore di utenti rispetto le altre operazioni, per questo motivo ha una probabilità maggiore di essere effettuata.

Probabilità per clienti con prenotazione:

- La probabilità di effettuare prenotazioni online per Spedizioni e Ritiri è minore rispetto la probabilità di effettuarla per le Operazioni Generiche poiché, essendo un'operazione con un tempo di servizio minore, ci si aspetta un numero minore di persone in coda e quindi meno tempo di attesa.
- Ricordiamo che non esiste la possibilità che un utente possa prenotare l'operazione di Versamenti e Prelievi presso ATM.

La percentuale dei clienti senza prenotazione che presentano una difficoltà è del 15%, non avevamo informazioni su tale probabilità. ***P_DIFF = 0.15***

Tassi di servizio:

Per i tassi di servizio è stata utilizzata una distribuzione normale troncata inferiormente, suddividendo tassi differenti per i tre tipi di operazioni, mantenendoli tali a prescindere se l'operazione è stata prenotata online o meno.

Tipo di operazione		**Tasso di servizio**	**Deviazione standard**
Operazione Generica	MU_OG	1/15 j/min	5/3
Spedizione e Ritiri	MU_SR	1/4.5 j/min	2/3
Versamenti e Prelievi	MU_ATM	1/2.5 j/min	1/3

4. Modello Computazionale

È stata utilizzata la "Next Event Simulation" come approccio per la realizzazione del modello computazionale.

Linguaggio di programmazione: Python 3.12

4.1 Stato del Sistema

Indichiamo con $\boldsymbol{SERER_NUM}$ il numero di server nel nostro sistema.

Indichiamo con $\boldsymbol{QUEUES_NUM}$ il numero di code che corrisponde al numero di classi di utenza.

Le variabili di sistema utilizzate per rappresentare univocamente lo stato del sistema ad ogni istante sono:

- $\boldsymbol{server_state[SERVER_NUM]}$: Array binario per tracciare lo stato degli sportelli.

  $server_state[i]$ = 0 implica che lo sportello con indice i è IDLE
  $server_state[i]$ = 1 implica che lo sportello con indice i è BUSY

- $\boldsymbol{num_client_in_service[QUEUES_NUM]}$: Array di valori interi per tracciare il numero di clienti in servizio per ciascuna classe di utenza.

- $\boldsymbol{num_client_in_queue[QUEUES_NUM]}$: Array di valori interi per tracciare il numero di clienti in ciascuna coda.

All'inizio della simulazione lo stato del sistema è impostato in modo da descrivere un sistema vuoto, con i serventi tutti liberi e in assenza di clienti nelle code.

Le classi di utenza citate in precedenza sono state mappate all'interno del codice con degli indici numerici, ciascuno corrispondente allo stream utilizzato per la generazione dei tempi di inter-arrivo:

Classe di utenza	Indice
Operazione Generica - Prenotazione Online	0
Operazione Generica - Persone in difficolta	1
Operazione Generica	2
Spedizione e Ritiri - Prenotazione Online	3
Spedizione e Ritiri - Persone in difficoltà	4
Spedizioni e Ritiri	5
Versamenti e Prelievi (ATM) - Persone in difficoltà	6
Versamenti e Prelievi (ATM)	7

4.2 Descrizione degli Eventi

Lo **stato del sistema può cambiare** solo al verificarsi di un **evento**. Nel nostro sistema riconosciamo come tali gli eventi di arrivo di un cliente nella filiale o all'ATM e gli eventi di completamento. Nello specifico:

- Evento: *Arrivo di un cliente della classe di utenza "c"*:

  Nel caso in cui almeno un server che può servirlo sia nello stato IDLE, comporta:
  - $\boldsymbol{num_client_in_service[c] \mathrel{+}= 1}$
  - $\boldsymbol{server_state[idle_{id}] = 1}$ (Impostato a BUSY il server che servirà il cliente appena arrivato)

  Nel caso in cui tutti i server che possono servirlo siano BUSY, comporta:
  - $\boldsymbol{num_client_in_queue[c]} \mathrel{+}= 1$ (Il cliente viene accodato)

- Evento: *Completamento di un servizio da parte di uno sportello*

  Nel caso in cui non ci siano altri clienti da servire (code vuote), comporta:
  - $\boldsymbol{num_client_in_service[c]} \mathrel{-}= 1$ ("c" classe del cliente servito)
  - $\boldsymbol{server_state[compl_{id}] = 0}$ (Impostato come IDLE il server che ha servito il cliente della classe c)

  Nel caso in cui ci siano altri clienti da servire (in coda), viene immediatamente servito il prossimo cliente secondo le regole di scheduling, comporta:
  - $\boldsymbol{num_client_in_queue[c]} \mathrel{-}= 1$

La lista degli eventi futuri è mantenuta in una classe Python, ***EventList*** che contiene a sua volta oggetti ***Event***:

```python
class EventList:
    def __init__(self):
        self.arrivals = []  # Lista degli eventi di arrivo, uno per ogni classe di utenza
        self.completed = [Event('C') for _ in range(SERVER_NUM)]  # Lista degli eventi di completamento
        self.sampling = 0  # Evento di campionamento
```

```python
class Event:
    def __init__(self, event_type=None):
        self.event_time = None          # Tempo in cui l'evento si verificherà
        self.event_type = event_type    # Arrivo o Completamento (A o C)
        self.op_index = None            # Tipo di operazione
```

4.3 Clock di Simulazione

L'orologio di sistema è contenuto all'interno di una classe Python.

```
class Times:
    def __init__(self):
        self.current = 0
        self.next = 0
        self.last = [0 for _ in range(QUEUES_NUM)]
```

Dove:

- ***Current:*** Descrive il tempo di simulazione corrente, garantendo che gli eventi vengano processati nel corretto ordine.
- ***Next:*** L'istante di tempo relativo al prossimo evento nel sistema.
- ***Last [QUEUES_NUM]:*** L'ultimo istante di arrivo per ogni classe di utenza.

Struttura del codice per gestione degli eventi e clock di simulazione:

Inizialmente viene impostato lo stato del sistema come vuoto, andando a modificare tutte le variabili di stato. Si generano i primi ***QUEUES_NUM*** istanti di arrivo, uno per ciascuna classe di utenza e viene avviato il ciclo:

while (time.current < T_FINE *OPPURE* ci sono clienti in coda)**{**

1. Ricerca del prossimo evento: *Trova l'evento più imminente nella lista degli eventi*. (Inizialmente, ci saranno solamente eventi di arrivo).
2. Processamento del prossimo evento (Next Event) e aggiornamento stato:
    a. ***Evento di arrivo****:* Viene selezionato (con *polita equity*[1]) il server e gli viene assegnato l'evento(cliente), aggiornando lo stato del sistema.
    Nel caso tutti server siano BUSY il cliente vene aggiunto nella coda. In fine viene generato un nuovo evento di arrivo per la stessa classe d'utenza.
    b. ***Evento di completamento****:* Si aggiorna lo stato del sistema.
    Se le code che possono essere servite dallo sportello in cui è avvenuto il completamento non sono vuote, verrà servito il prossimo cliente in coda secondo le regole di scheduling.
    Se viene servito un cliente in coda, viene generato l'evento di completamento per il nuovo cliente accolto.

}

[1] *Equity*: Per servire un nuovo arrivo, se ci sono più server IDLE, viene scelto *il server libero da più tempo*.

5. Verifica

La verifica riguarda il processo di controllo e di revisione di quanto elaborato durante le fasi di sviluppo, per assicurarsi che questo soddisfi i requisiti specificati. In altre parole, si tratta di verificare se il prodotto è stato costruito correttamente secondo le specifiche e i criteri di progetto. La verifica risponde alla domanda: **"Stiamo costruendo il prodotto correttamente?"**

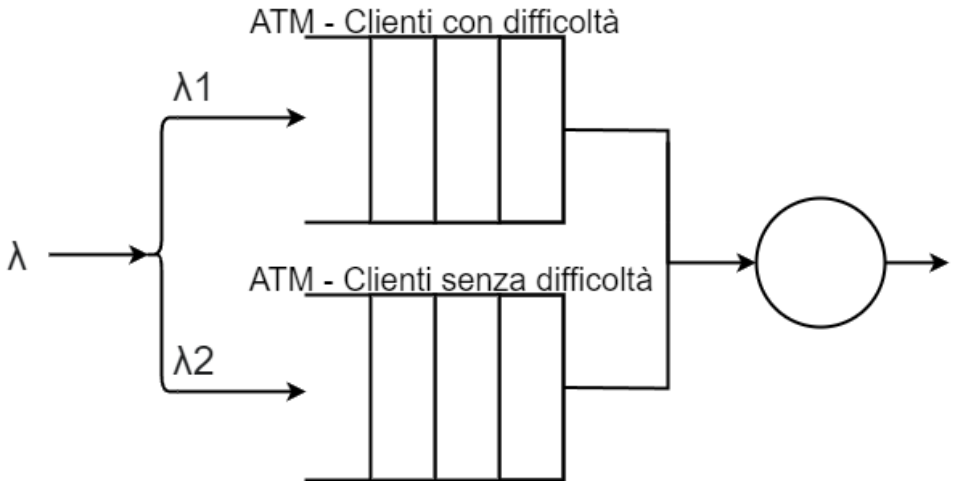
È stata effettuata una simulazione ad orizzonte infinito per garantire che il sistema operasse in uno stato stazionario, tramite il metodo **Batch Means.** Inoltre, sono stati sostituiti solo per questa fase le distribuzioni dei tempi di servizio con delle esponenziali.

La scelta della dimensione del batch, cioè del valore del parametro b, è stata guidata dal criterio di Banks et al. [2001][2], secondo cui il valore di b deve aumentare fintanto che il valore di lag-autocorrelation, tra le medie dei batch, non scenda sotto 0.2.

Ciò è stato verificato tramite l'analisi dei dati di output inseriti nella classe acs.py, ottenendo (b, k) = (512, 1024).

5.1 Confronto con medie teoriche

Sistema ATM:



È stato verificato il corretto funzionamento dell'ATM con le due code di priorità da gestire, modellato come un centro M/M/1 con abstract priority per questa fase di verifica, che presenta i seguenti tempi medi di coda:

$$E(Tq_k) = \frac{\lambda/_2\, E(S^2)}{(1-\sum_{i=0}^{k}\rho_i)(1-\sum_{i=0}^{k-1}\rho_i)}$$

[2] Banks, Carson, Nelson, and Nicol (2001, page 438)

Dove:

$\lambda_1 = \lambda_{totale} * (1 - P_{ON}) * (P_{diff}) * P_{ATM} = 0{,}034999\ job/min$

$\lambda_2 = \lambda_{totale} * (1 - P_{ON}) * (1 - P_{diff}) * P_{ATM} = 0{,}198333\, job/min$

$\lambda = \lambda_1 + \lambda_2 = 0{,}233333\ \ job/min$

$\mu = {}^{1}/_{2{,}5} = 0{,}4\, job/min$ $\qquad$ $E(S^2) = \frac{2}{\mu^2} = 12{,}499999$

$\rho_1 = \lambda_1 * E(S) = \ \ 0{,}087499$ $\qquad$ $\rho_2 = \lambda_2 * E(S) = \ \ 0{,}495832$

Tramite i quali troviamo i seguenti valori teorici, che ricadono negli intervalli di confidenza calcolati con i valori ottenuti dai 1024 batch.

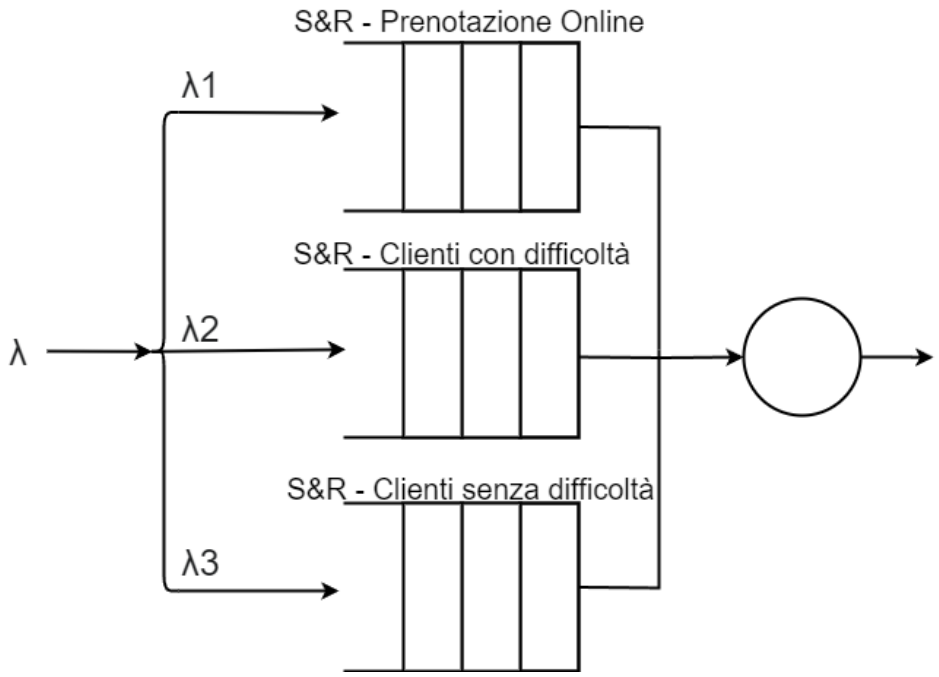
$E(Tq_1) = 1{,}5981873\, minuti$

```
based upon 1024 data points and with 95 confidence
the expected value is in the interval       1.59 +/-   0.04
```

$E(Tq_2) = 3{,}835616\, minuti$

```
based upon 1024 data points and with 95 confidence
the expected value is in the interval       3.76 +/-   0.11
```

Sistema Spedizioni e Ritiri:



Il secondo centro valutato è Spedizioni e Ritiri, con tre code di priorità, anch'esso modellato per questa fase di verifica come un M/M/1 con abstract priority.

$\lambda_1 = \lambda_{totale} * (P_{ON}) * (1 - P_{diff}) * P_{SR} = 0{,}04\ job/min$

$\lambda_2 = \lambda_{totale} * (1 - P_{ON}) * (P_{diff}) * P_{SR} = 0{,}019999\, job/min$

$\lambda_3 = \lambda_{totale} * (1 - P_{ON}) * (1 - P_{diff}) * P_{SR} = 0{,}113333\, job/min$

$\lambda = \lambda_1 + \lambda_2 + \lambda_3 = 0{,}173333\ job/min$

$\mu = {}^{1}/_{3} = 0{,}333333\, job/min$ $\qquad E(S^2) = \frac{2}{\mu^2} = 18{,}000036$

$\rho_1 = \lambda_1 * E(S) = 0{,}12$ $\qquad \rho_2 = \lambda_2 * E(S) = 0{,}0599997$

$\rho_3 = \lambda_3 * E(S) = 0{,}339999$

Tramite i quali troviamo i seguenti valori teorici, che ricadono negli intervalli di confidenza calcolati con i valori ottenuti dai 1024 batch.

$E(Tq_1) = 1{,}772727\ minuti$

```
based upon 1024 data points and with 95 confidence
the expected value is in the interval       1.79 +/-   0.05
```

$E(Tq_2) = 2{,}161862\ minuti$

```
based upon 1024 data points and with 95 confidence
the expected value is in the interval       2.17 +/-   0.08
```

$E(Tq_3) = 3{,}963414\ minuti$

```
based upon 1024 data points and with 95 confidence
the expected value is in the interval       3.90 +/-   0.13
```

Verifica Statistiche di output:

Per chiarezza, introduciamo la classe Area definita per mantenere il valore delle aree relative ad una singola classe di utenza, essenziali per produrre le statistiche di output.

```
class Area:
    def __init__(self):
        self.customers = 0  # clienti nel sistema [Unità di misura: (Num clienti) * tempo]
        self.queue = 0      # clienti in coda queues [Unità di misura: (Num Clienti) * tempo]
        self.service = 0    # Clienti in servizio [Unità di misura: (Num Clienti) * tempo]
```

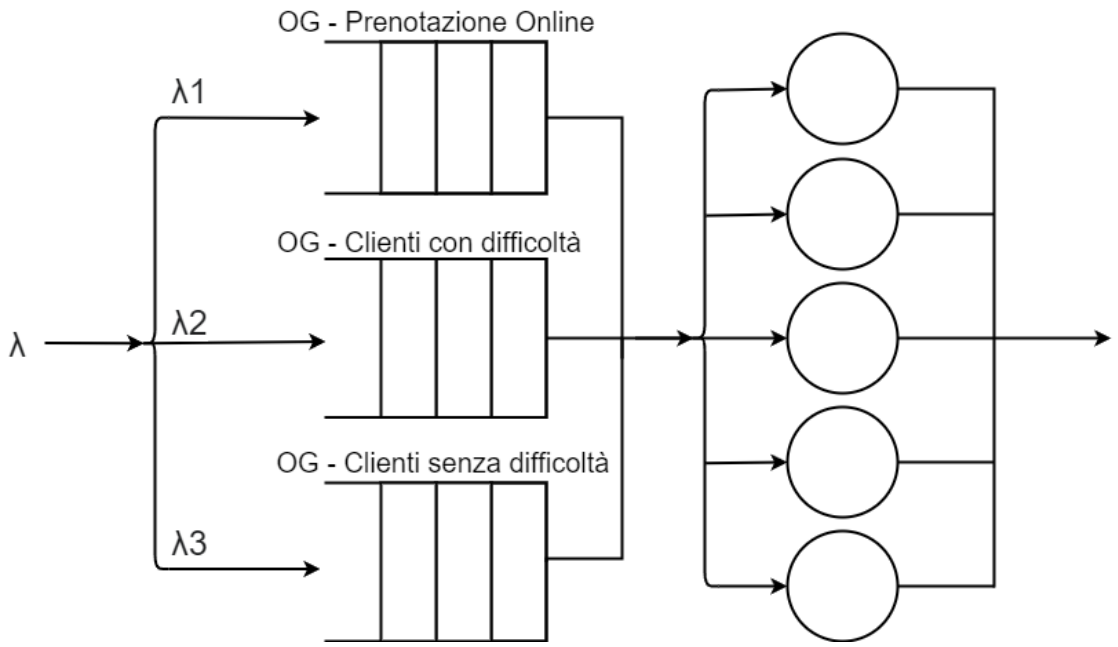
La funzione ***update_area(area_list)*** è utilizzata per aggiornare la lista degli oggetti di tipo Area() a seguito di ogni evento:

```
def update_area(area_list):
    for i in range(QUEUES_NUM):
        if num_client_in_system[i] > 0:
            area_list[i].customers += (times.next - times.current) * num_client_in_system[i]
            area_list[i].service += (times.next - times.current) * num_client_in_service[i]
            area_list[i].queue += (times.next - times.current) * (len(queues[i]))
```

Possiamo notare come per tutte e tre le statistiche, sono stati usati valori indipendenti l'uno dall'altro, ad esempio: il calcolo dei clienti in coda non è stato effettuato tramite la sottrazione tra le persone nel sistema e quelle in servizio, così da poter essere utilizzati come ulteriore verifica del sistema

Multi-Server – Operazioni Generiche:

Per quanto riguarda il multi-server, è stato modellato come un M/M/5. Per semplicità di calcolo, verifichiamo il funzionamento del sistema osservando i risultati ottenuti e paragonandoli a quelli ottenuti analiticamente dal teorema di Little.



In figura notiamo le statistiche di output ottenute tramite la classe Area() mostrata precedentemente, aggiornandole al seguito del processamento di ciascun evento.

queue	avg wait t	avg delay t	avg service t	avg # in node	avg # in queue	avg # in service
1	17.48882967	2.92454821	14.56428146	0.51235028	0.08567715	0.42667313
2	17.55592272	2.51365164	15.04227108	0.83576322	0.11966432	0.71609890
3	36.84031140	22.41511116	14.42520024	8.90396558	5.41752690	3.48643867
4	4.86612915	0.46538188	4.40074727	0.08909839	0.00852110	0.08057729
5	5.02660170	0.72095263	4.30564907	0.09203663	0.01320058	0.07883605
6	8.21018426	3.71836718	4.49181707	0.87190094	0.39488125	0.47701969
7	3.09811666	0.86869051	2.22942615	0.11345248	0.03181129	0.08164119
8	3.59748242	1.40050713	2.19697530	0.57965235	0.22565982	0.35399253

```
Simulation complete. Clients served: [8, 13, 66, 5, 5, 29, 10, 44]
last completion: 273.0761402290368
Tempo Totale 04:33:08
```

- **Statistiche mediate sui job:**

$\bar{w}$: tempo medio di attesa $\bar{d}$: tempo medio in coda $\bar{s}$: tempo medio di servizio

Si può notare che la relazione $\bar{w} = \bar{d} + \bar{s}$ sia rispettata per ciascuna coda.

- **Statistiche mediate sul tempo**

$\bar{l}$: popol. media nel centro $\bar{q}$: popol. media in coda $\bar{x}$: popol. media in servizio

È facile notare la seguente relazione $\bar{l} = \bar{q} + \bar{x}$, verificata nuovamente per ciascuna coda.

Dal Teorema di Little per la popolazione nel sistema $c_n * \bar{l} = n * \bar{w}$ possiamo notare :

- $\bar{l} = \frac{n}{c_n} * \bar{w}$ Con c_n tempo di ultimo completamento.

Tale relazione vale anche per:

- $\bar{q} = \frac{n}{c_n} * \bar{d}$
- $\bar{x} = \frac{n}{c_n} * \bar{s}$

Nell'output proposto in foto, prendiamo l'esempio più interessante, la terza coda:

$\mathrm{n} = 66$

$c_n = 273.07614\ min$

$\bar{w} = 36.84031140\ min$

$\bar{d} = 22.415111\ min$

$\bar{s} = 14.4252\ min$

Andando a svolgere i calcoli troviamo:

$\bar{l} = \frac{66}{273.07614} * 36.84031140 = 8.90396558$

$\bar{q} = \frac{66}{273.07614} * 22.415111 = 5.417526904$

$\bar{x} = \frac{66}{273.07614} * 14.4252 = 3.486438673$

Rispecchiando le statistiche ottenute in output.

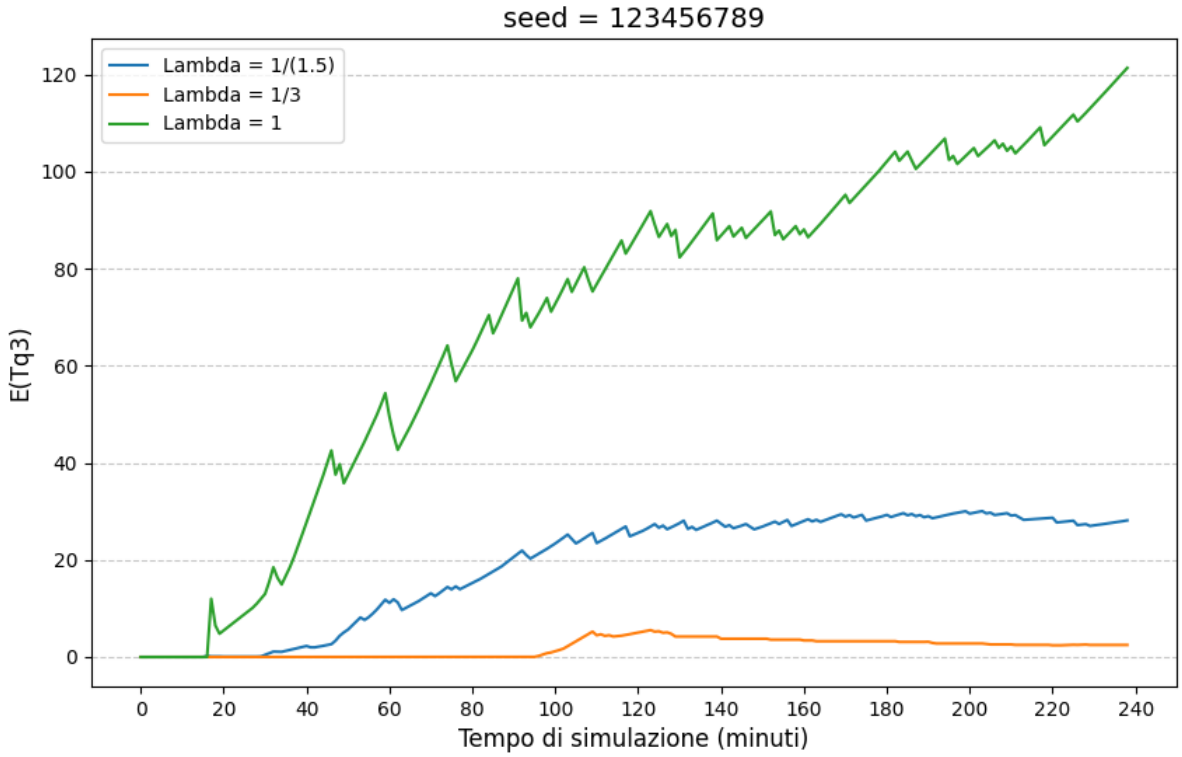
6. Validazione

Nella fase di validazione, ci si domanda se il modello computazionale realizzato sia consistente con il sistema che desideravamo rappresentare. Purtroppo, non essendo in possesso di dati relativi al sistema in esame con cui effettuare un confronto, per produrre una validazione comunque significativa, abbiamo deciso di controllare che il valore delle metriche vari al variare del tasso di arrivo. A parità di configurazione, un'utilizzazione più elevata implica un peggioramento delle prestazioni del sistema, e viceversa.

Per la precisione, è bene effettuare ogni test variando un solo parametro di input, altrimenti sarebbe difficile associare un cambiamento delle performance al cambiamento di uno specifico parametro modificato.

Parametro variato: Tasso di inter-arrivo dei clienti senza prenotazione.

Parametro base: $\lambda = \frac{1}{1.5}$ Parametro ridotto: $\lambda = \frac{1}{3}$ Parametro aumentato: $\lambda = 1$



Vediamo, come da aspettative, che i tempi di attesa medi per tutte le classi d'utenza diminuiscono e aumentano rispettivamente al diminuire e all'aumentare del tasso di inter-arrivo dei clienti nel sistema. Per brevità è mostrata la statistica della sola classe d'utenza relativa alla terza coda di priorità del multi-server.

È più significativo valutare come il parametro di output (tempo medio in coda) mediato su mille repliche, ognuna con un seed indipendente dagli altri, cambi come aspettato al variare di λ.
È mostrata unicamente la terza coda del sistema, ma precisiamo che tale relazione è valida per ciascuna classe d'utenza.

λ	E(Tq3)
Base = 1/1.5 j/min	23.081 ± 0.898 minuti
Ridotto = 1/3 j/min	1.363 ± 0,105 minuti
Aumentato = 1 j/min	83.853 ± 1.366 minuti

Inoltre, come aspettato, anche l'utilizzazione dei centri è proporzionale al variare di λ.

λ	ρ $(multi\ server)$
Base = 1/1.5 j/min	0.844
Ridotto = 1/3 j/min	0.567
Aumentato = 1 j/min	0.925

7. Design degli esperimenti

7.1 Analisi del transitorio

Effettuiamo tale analisi tramite una simulazione ad orizzonte finito, con orario di chiusura impostato a 16 ore = 960 minuti, per valutare in modo completo la fase transitoria del sistema. Per questa analisi è stato introdotto un nuovo tipo di evento: **Evento di Campionamento**, utilizzato per salvare le statistiche di sistema ad intervalli regolari senza modificare lo stato del sistema.

La simulazione ad orizzonte finito è stata effettuata sul sistema reiterandola su molteplici repliche (run di simulazione), così da ottenere stime per le statistiche transitorie indipendenti dal seed iniziale. Ogni replica viene eseguita a parità di stato iniziale del sistema e parametri di input (tassi di inter-arrivo e di servizio e probabilità di routing).

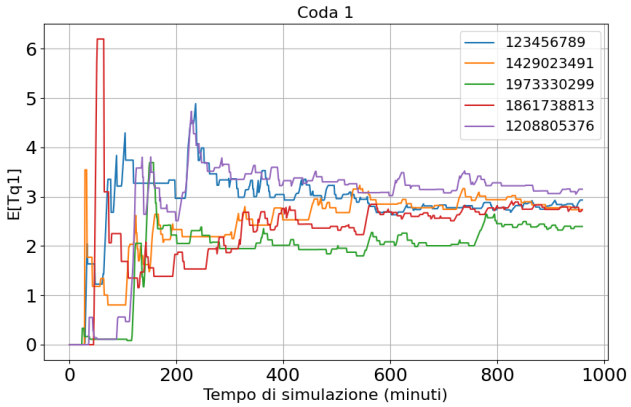
Il seed di default della libreria rngs utilizzata è pari a 123456789, inizializzato esternamente al ciclo che ripete la simulazione, in modo tale che ogni replica abbia come seed di partenza del generatore di numeri casuale il seed finale della replica precedente, così da non avere overlap nelle generazioni casuali effettuate.

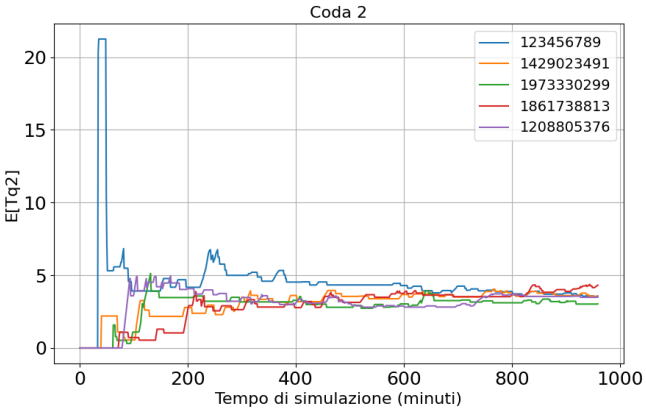
Risultati:

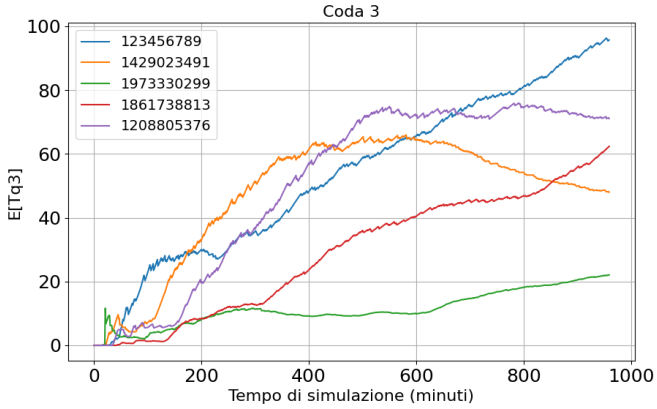
Mostriamo nella pagina seguente l'andamento della simulazione effettuata con un numero di replicazioni pari a 5, ciascuna identificata tramite il proprio seed iniziale. Per le code con priorità (Code 1, 2, 4, 5, 7), dove si ha un tempo di inter-arrivo maggiore rispetto le altre, è comprensibile un andamento a gradini, essendo il tempo medio di coda una delle statistiche job-averaged.

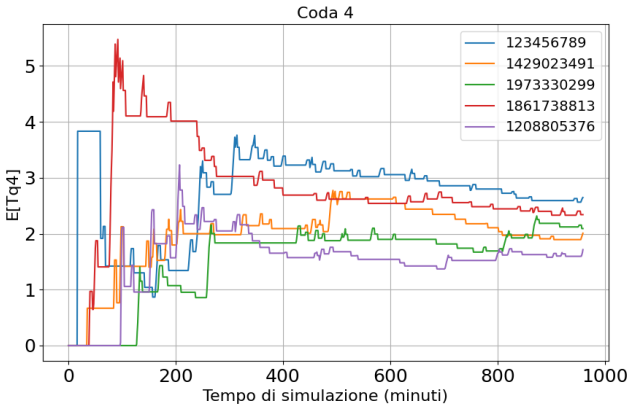
I risultati sono ancora più comprensibili sulla successiva pagina dove mostriamo l'andamento per ciascuna coda nel sistema, mediato sui risultati ottenuti dalla simulazione con un numero di repliche pari a 100, evidenziando inoltre l'intervallo di confidenza. Possiamo notare come in generale i tempi medi di coda convergano ad uno stato stabile, ad eccezione della coda 3 (Operazioni Generiche senza priorità), dove il tempo medio in coda aumenta notevolmente al passare del tempo di simulazione.

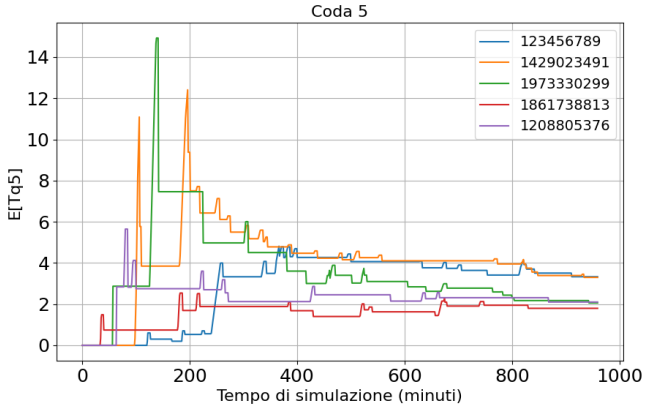
Come accennato in precedenza, il tempo di simulazione (16 ore) impostato ad un valore maggiore di una giornata lavorativa (4 ore) è risultato molto utile per valutare l'effettiva convergenza dei tempi medi in coda, poiché soffermarsi ai soli 240 minuti (linea rossa) non avrebbe mostrato il completo comportamento delle statistiche transitorie.

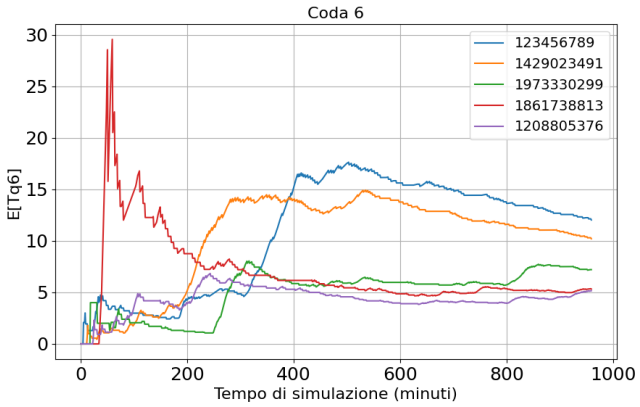

Coda 1
E[Tq1]
Tempo di simulazione (minuti)
123456789
1429023491
1973330299
1861738813
1208805376

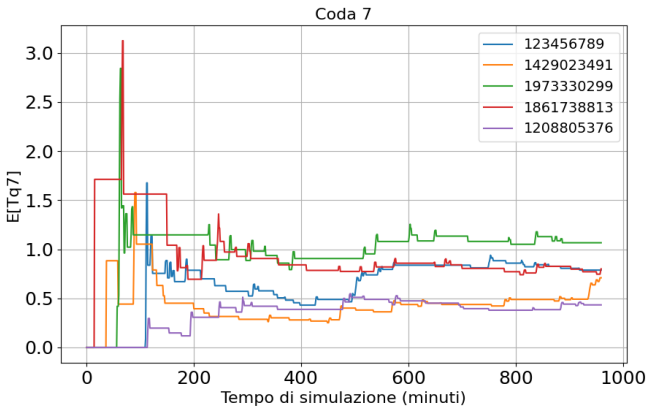

Coda 2
E[Tq2]
Tempo di simulazione (minuti)
123456789
1429023491
1973330299
1861738813
1208805376

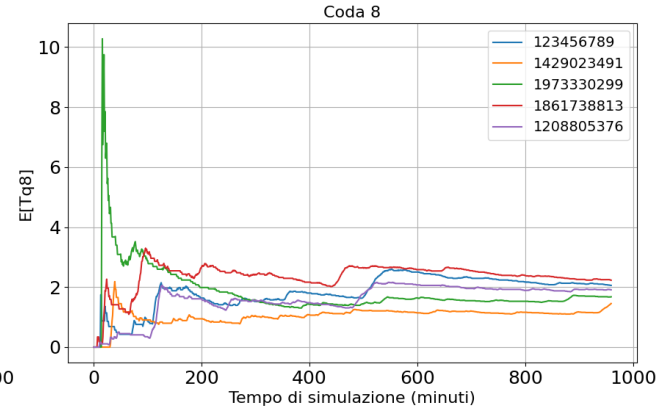

Coda 3
E[Tq3]
Tempo di simulazione (minuti)
123456789
1429023491
1973330299
1861738813
1208805376

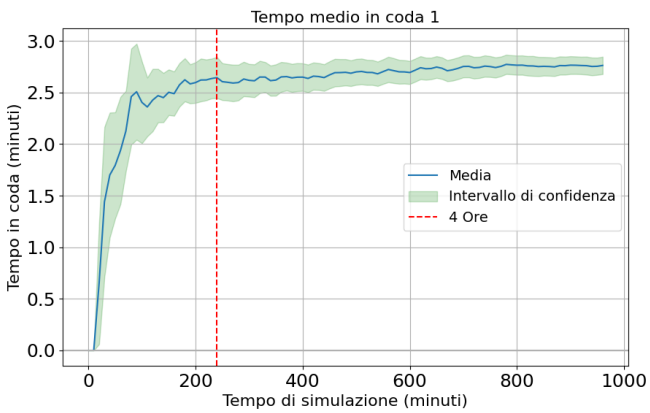

Coda 4
E[Tq4]
Tempo di simulazione (minuti)
123456789
1429023491
1973330299
1861738813
1208805376

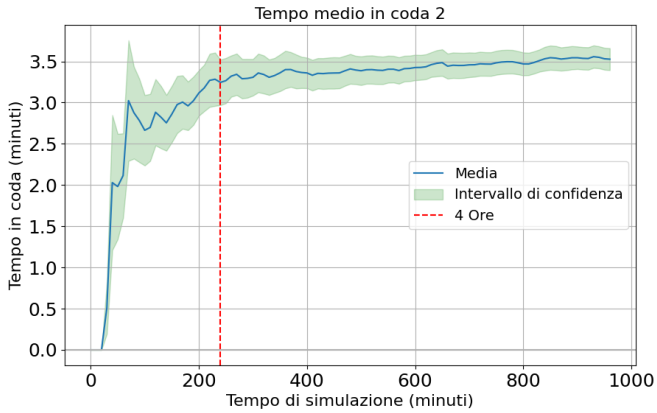

Coda 5
E[Tq5]
Tempo di simulazione (minuti)
123456789
1429023491
1973330299
1861738813
1208805376

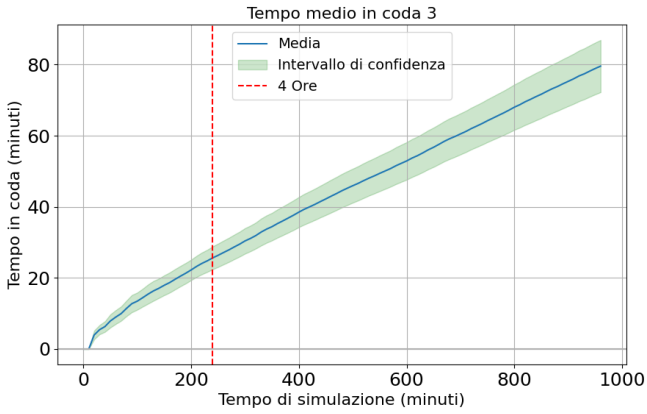

Coda 6
E[Tq6]
Tempo di simulazione (minuti)
123456789
1429023491
1973330299
1861738813
1208805376

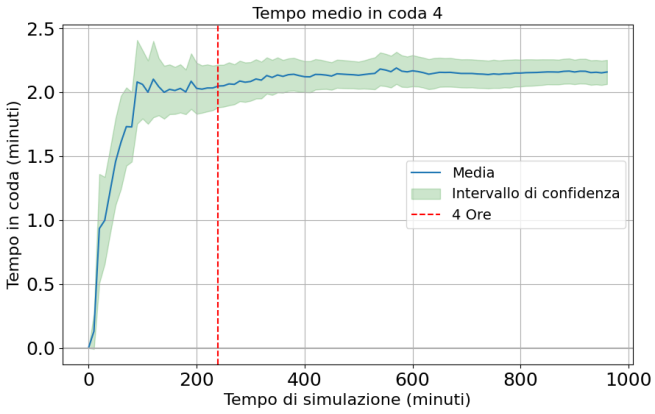

Coda 7
E[Tq7]
Tempo di simulazione (minuti)
123456789
1429023491
1973330299
1861738813
1208805376

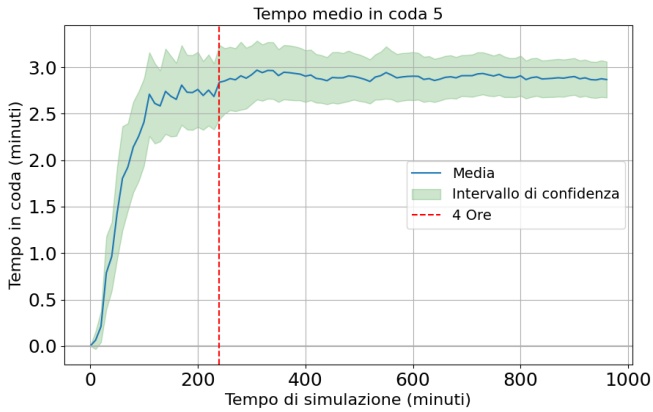

Coda 8
E[Tq8]
Tempo di simulazione (minuti)
123456789
1429023491
1973330299
1861738813
1208805376

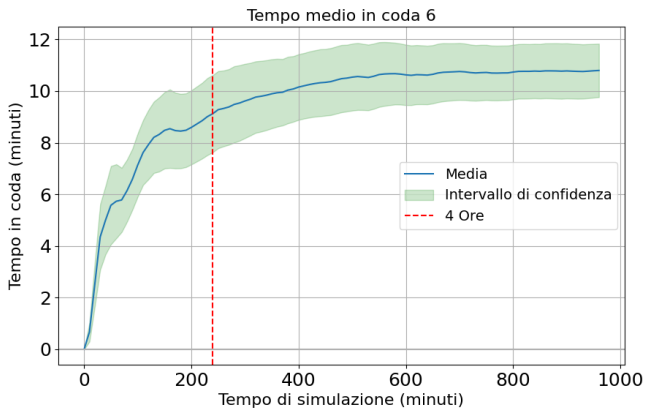

Tempo medio in coda 1
Tempo in coda (minuti)
Tempo di simulazione (minuti)
Media
Intervallo di confidenza
4 Ore

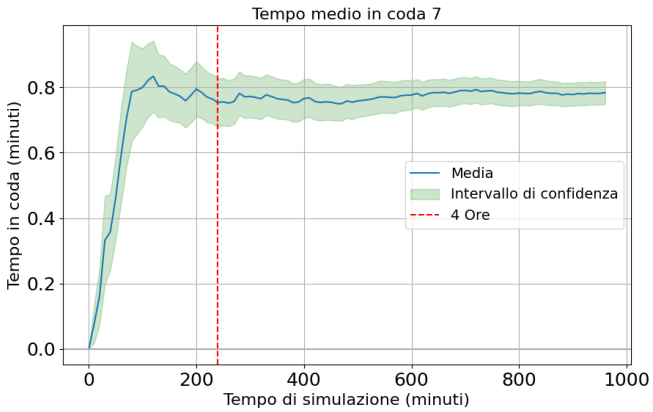

Tempo medio in coda 2
Tempo in coda (minuti)
Tempo di simulazione (minuti)
Media
Intervallo di confidenza
4 Ore

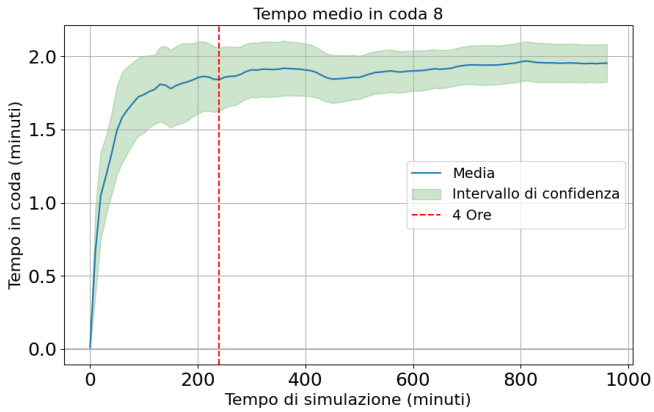

Tempo medio in coda 3
Tempo in coda (minuti)
Tempo di simulazione (minuti)
Media
Intervallo di confidenza
4 Ore


Tempo medio in coda 4
Tempo in coda (minuti)
Tempo di simulazione (minuti)
Media
Intervallo di confidenza
4 Ore

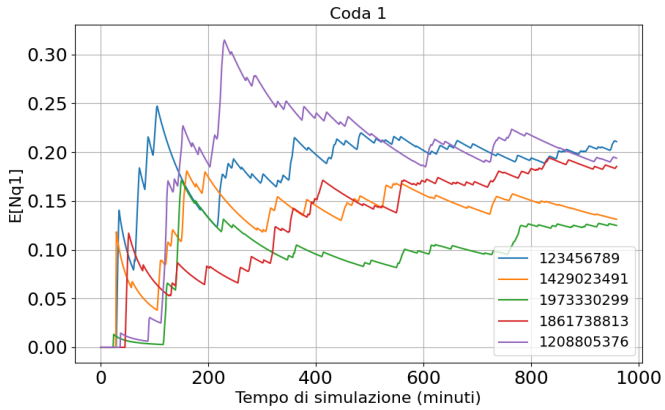

Tempo medio in coda 5
Tempo in coda (minuti)
Tempo di simulazione (minuti)
Media
Intervallo di confidenza
4 Ore

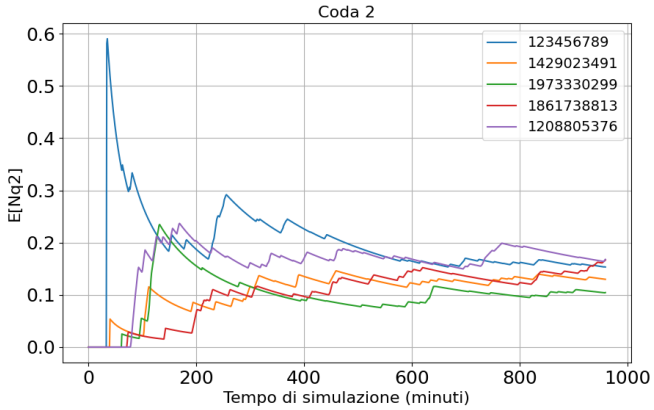

Tempo medio in coda 6
Tempo in coda (minuti)
Tempo di simulazione (minuti)
Media
Intervallo di confidenza
4 Ore

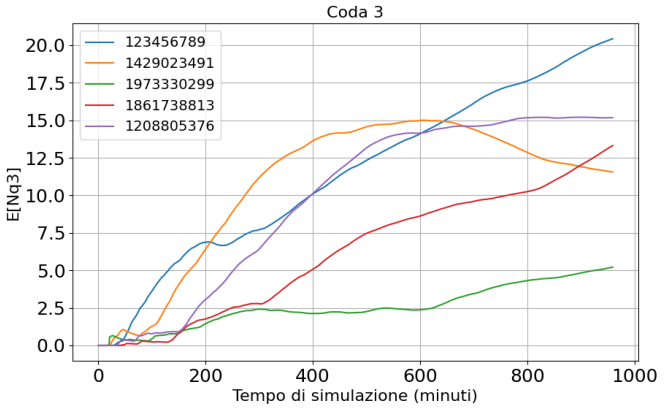

Tempo medio in coda 7
Tempo in coda (minuti)
Tempo di simulazione (minuti)
Media
Intervallo di confidenza
4 Ore

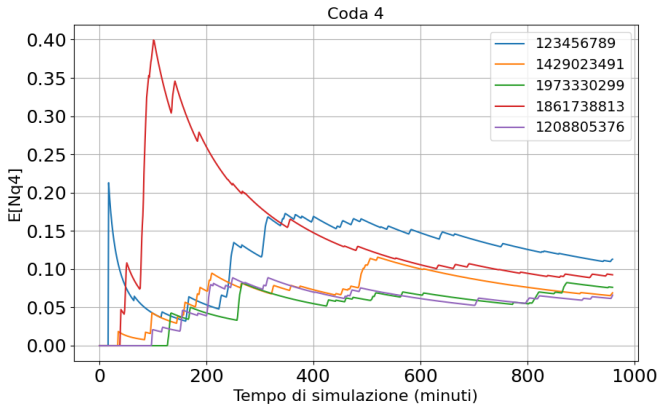

Tempo medio in coda 8
Tempo in coda (minuti)
Tempo di simulazione (minuti)
Media
Intervallo di confidenza
4 Ore

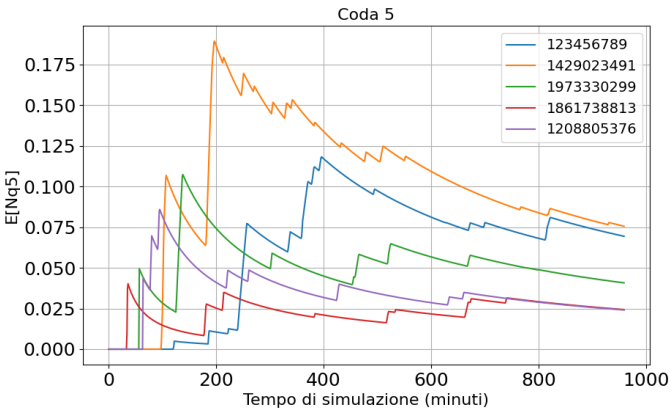
Per le statistiche time-averaged (in questo caso le popolazioni medie in coda) constatiamo subito l'assenza del comportamento a gradini. È inoltre evidente come per le code a più alta priorità il valore di tale statistica sia abbondantemente sotto il valore uno, come risultato di una bassa affluenza e precedenza nello svolgere le operazioni richieste.

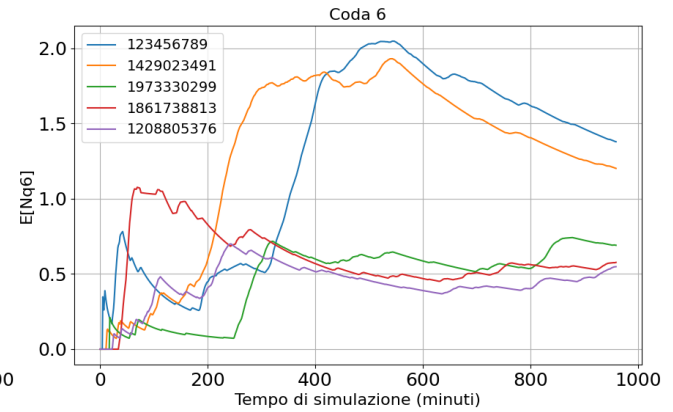


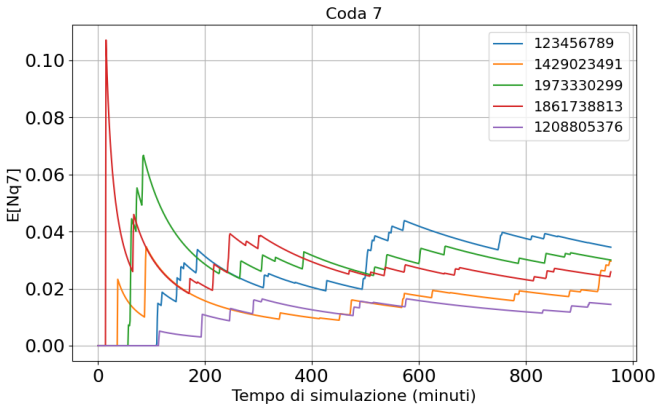


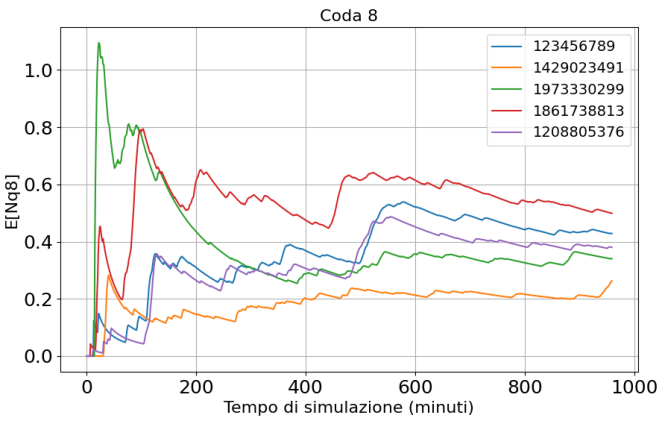












7.2 Analisi dello Stazionario

Per l'analisi dello stazionario, abbiamo effettuato una simulazione ad orizzonte infinito con l'ausilio delle batch means (b,k) = (512,1024), così che i risultati non siano influenzati dallo stato iniziale. Da essa possiamo valutare come il comportamento del nostro sistema sia coerente con le aspettative:

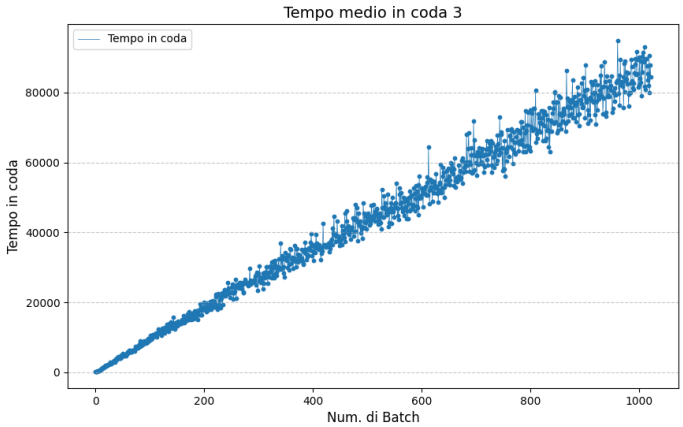
- Il tempo medio nella coda 3, non tende ad alcun valore stazionario, ma bensì all'infinito, all'aumentare del numero di job processati nel sistema. Poiché il sistema non riesce a smaltire in alcun modo i clienti al suo interno, non viene mai raggiunto uno stato stazionario.
  Otteniamo per l'appunto un'utilizzazione media di: `Mean: 0.9999789914950903`
  I job in ingresso non vengono smaltiti con la stessa velocità con cui arrivano, causando innumerevoli accodamenti.
- Per le altre code (valutiamo per comodità esclusivamente quelle a priorità minore: 6 e 8) notiamo come il tempo medio in coda rimanga costante.

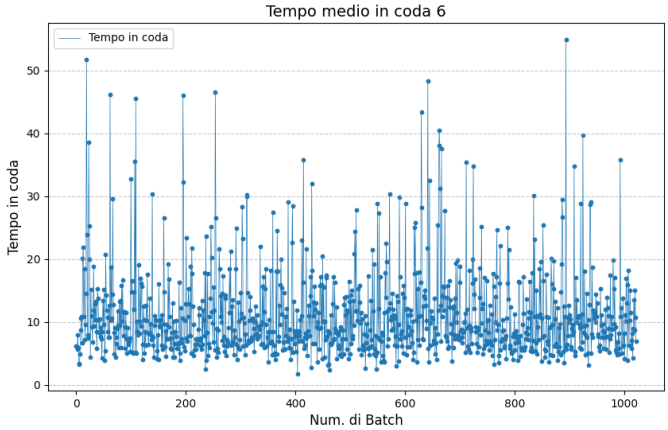
Andando ad analizzare l'utilizzazione del server Spedizioni e Ritiri otteniamo:

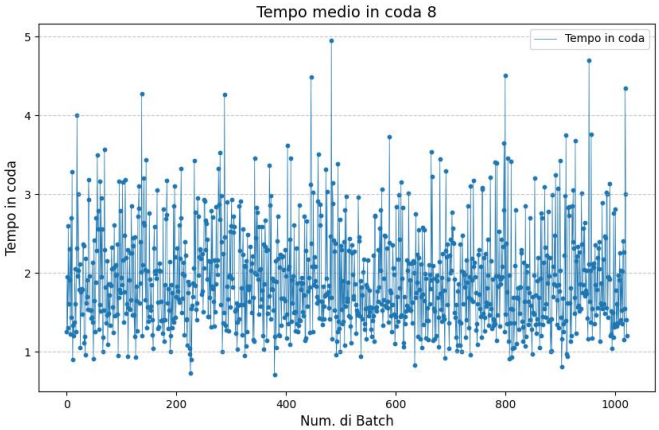
```
based upon 1024 data points and with 95 confidence
the expected value is in the interval 0.78461446 +/- 0.00434790
```

Mentre per l'utilizzazione del servente ATM otteniamo:

```
based upon 1024 data points and with 95 confidence
the expected value is in the interval 0.58257749 +/- 0.00283777
```







7.3 Utilizzazione del sistema

È interessante valutare l'utilizzazione di ciascun centro, l'analisi è stata svolta mediando ciascuna utilizzazione su mille repliche. Tale valore è stato raccolto al tempo di chiusura della filiale, 240 minuti, e non al tempo dell'ultimo completamento, questo perché altrimenti sarebbe influenzata dal fatto che a seguito della chiusura, ciascun centro può solamente svuotarsi, e/o possono esserci serventi vuoti.

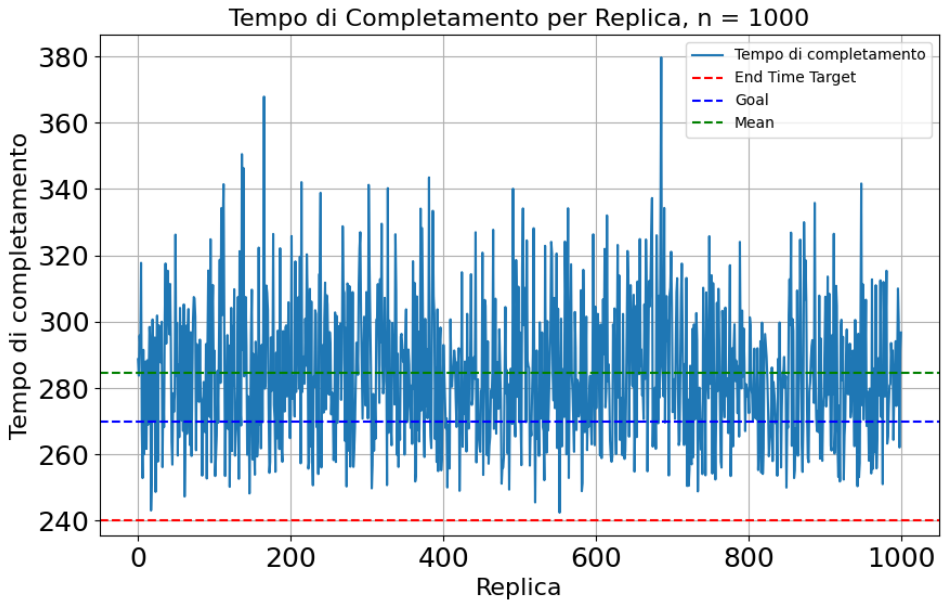
Multi-Server	0.89851858 ± 0.00296195
Spedizioni e Ritiri	0.74158964 ± 0.00682837
ATM	0.57433404 ± 0.00468570

7.4 Obiettivi studiati

1. Il primo obiettivo proposto prevede di mantenere il tempo medio in coda per ciascuna classe di utenza inferiore ai 15 minuti. Nei grafici mostrati nell'analisi transitoria, possiamo vedere come al termine delle 4 ore lavorative solamente la classe d'utenza 3, relativa alle Operazioni Generiche senza priorità, eccede, essendo in un intorno di 23 minuti.

2. Il secondo obiettivo proposto prevede la riduzione del tempo medio di sovraccarico (minuti trascorsi dopo le 4 ore di giornata lavorativa, linea rossa) sotto i 30 minuti.
   Dalla seguente figura, rappresentata grazie ad una simulazione ad orizzonte finito con mille repliche, è evidente come tale obiettivo non venga raggiunto, trovando una media di circa 45 minuti oltre il termine dell'orario lavorativo.
   In verde è rappresentato il valore medio ottenuto, mentre in blu l'obiettivo che si vuole raggiungere.

```
Basato su 1000 punti dati e con 95% di confidenza
Il valore atteso è nell'intervallo 284.43204503 +/- 1.29771595
```



Modello Migliorativo

Per raggiungere gli obiettivi stabiliti, abbiamo individuato due strategie migliorative.

La prima riguarda l'utilizzo dello sportello dedicato alle operazioni di Spedizioni e Ritiri. Abbiamo analizzato come l'utilizzazione di tale sportello sia inferiore rispetto l'utilizzazione dei multi-server, in effetti questo sportello rimane inutilizzato in assenza di clienti che richiedono tali operazioni, comportando un'inefficienza operativa. Per ovviare a questo problema, abbiamo proposto una gestione più dinamica dello sportello: quando non ci sono clienti in coda per spedizioni e ritiri, lo sportello può essere impiegato per servire altre tipologie di operazioni, sempre rispettando le priorità di servizio stabilite per le diverse classi di utenza. Questa soluzione permette di massimizzare l'uso delle risorse disponibili, riducendo i tempi di inattività dello sportello.

Il secondo miglioramento proposto prevede l'installazione di un Locker automatico all'esterno della filiale, a fronte di un investimento iniziale. Questo Locker è progettato esclusivamente per Spedizione e Ritiri di pacchi di piccole e medie dimensioni, escludendo quelli più voluminosi. L'introduzione del Locker consente ai clienti di effettuare l'operazione autonomamente, senza dover attendere in coda presso lo sportello. Di conseguenza, si prevede una riduzione del tempo di attesa per tali clienti, migliorando l'esperienza complessiva del servizio e diminuendo l'affollamento all'interno della filiale.

Tuttavia, per maggiore precisione nell'analisi, abbiamo deciso di considerare inizialmente i risultati senza l'introduzione del Locker, rendendolo un miglioramento opzionale a causa del costo iniziale richiesto. In questo modo, sarà possibile valutare separatamente l'impatto di ciascun miglioramento e determinare la loro efficacia.

1. Obiettivi e Scopi

Nel modello migliorativo abbiamo mantenuto gli stessi obiettivi del modello base, con lo scopo di migliorarli ove possibile.

2. Modello Concettuale

Per la prima ottimizzazione, il sistema e le classi d'utenza restano invariate. Il cambiamento principale risiede nella gestione delle code: lo sportello dedicato alle operazioni di "Spedizioni e Ritiri" può ora prendere in carico anche le "Operazioni Generiche" quando non ci sono clienti in attesa per "Spedizioni e Ritiri". Questa modifica permette di utilizzare al meglio lo sportello, riducendo i tempi di inattività e aumentando la flessibilità operativa.

È stato aggiunto un collegamento tra le code delle Operazioni Generiche ed il servente per le operazioni Spedizioni e Ritiri.

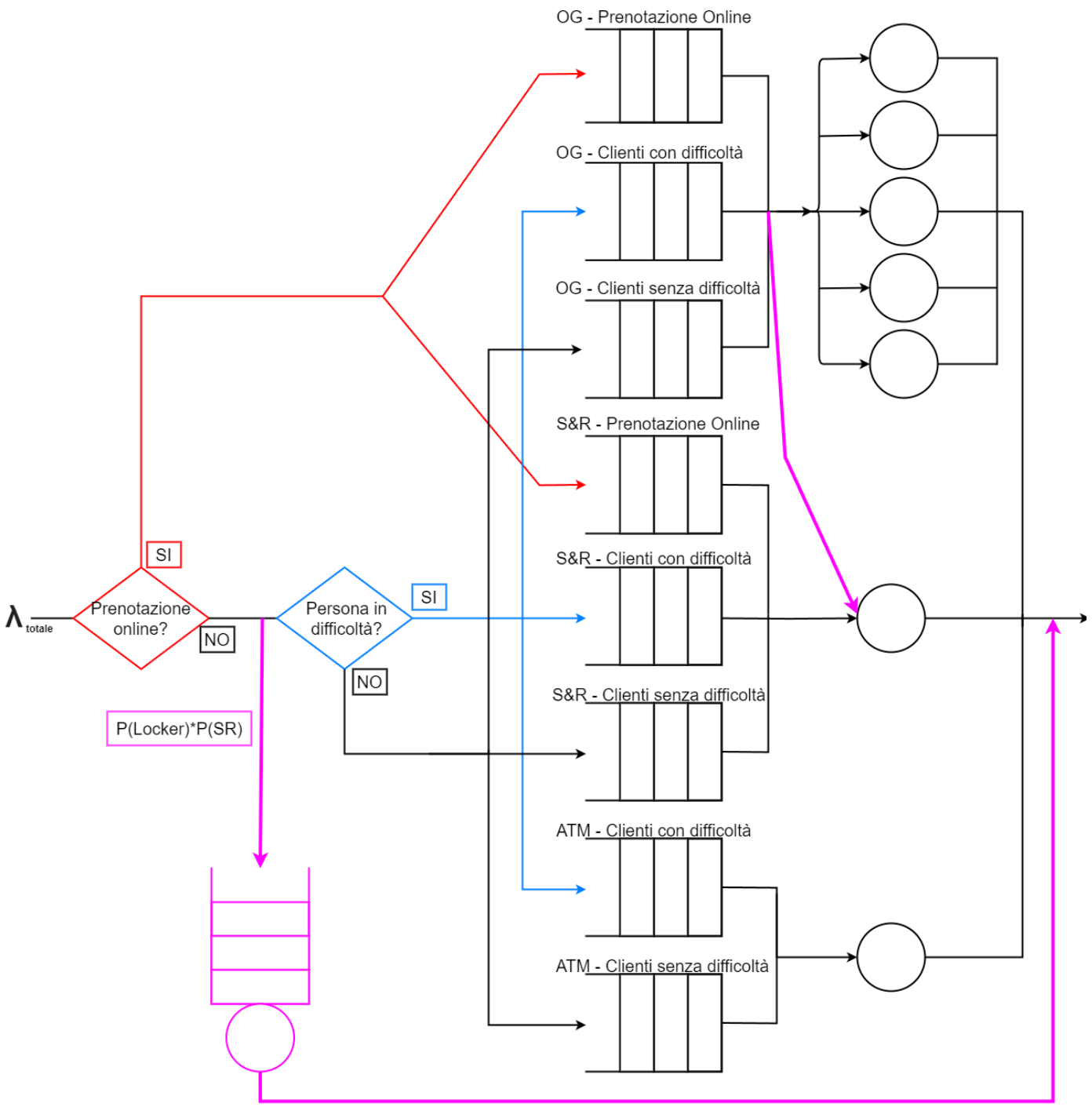
Le code di priorità per lo sportello Spedizioni e Ritiri sono così definite:

1. Spedizioni e Ritiri – Utenti con Prenotazioni
2. Spedizioni e Ritiri – Utenti con Difficoltà
3. Spedizioni e Ritiri – Utenti senza Difficoltà
4. Operazioni Generiche – Utenti con Prenotazioni
5. Operazioni Generiche – Utenti con Difficoltà
6. Operazioni Generiche – Utenti senza Difficoltà

I rimanenti sportelli lavorano con le stesse code di priorità precedentemente descritte.

Nel caso del secondo miglioramento, l'introduzione di un Locker comporta l'aggiunta di un nuovo servente e la creazione di una coda dedicata. Questo nuovo elemento introduce una nuova dinamica nel sistema, permettendo ai clienti di utilizzare il Locker per ritirare e spedire i pacchi senza dover attendere allo sportello. Data la velocità nell'utilizzo di tale Locker, non è stata aggiunta una coda di priorità per persone in difficoltà.

Nel modello sotto illustrato, i miglioramenti sono in colore rosa.


OG - Prenotazione Online
OG - Clienti con difficoltà
OG - Clienti senza difficoltà
S&R - Prenotazione Online
S&R - Clienti con difficoltà
S&R - Clienti senza difficoltà
ATM - Clienti con difficoltà
ATM - Clienti senza difficoltà
λ totale
Prenotazione online?
SI
NO
Persona in difficoltà?
SI
NO
P(Locker)*P(SR)

3. Modello delle specifiche

Nel modello migliorativo, le specifiche rimangono simili a quelle del modello base, con l'aggiunta di nuove funzionalità per riflettere i miglioramenti implementati:

a. **Prima Ottimizzazione**: In questo caso, il modello delle specifiche rimane invariato rispetto al modello base.
b. **Seconda Ottimizzazione**: L'aggiunta del Locker introduce un nuovo servente (Locker) e una coda dedicata per le operazioni di ritiro. Questo richiede l'inclusione di nuove variabili per monitorare lo stato del Locker (IDLE o BUSY) e la popolazione della nuova coda.

   - Il tasso di ingresso al Locker è: $\boldsymbol{P_{SR} * P_{Locker} * (1 - P_{ON}) * \lambda_{totale}}$ La probabilità che un cliente che intende effettuare operazioni di Spedizione e Ritiri, vada in un Locker è $P_{Locker} = 0{,}5$.
   - Il tasso medio di servizio nel Locker è di 1 job/min, dovendo solamente scansionare un codice QRCode e ritirare o inserire il pacco nello sportello indicato, che si aprirà in automatico.
   - È stata adottata una distribuzione Normale con deviazione standard: 0.5/3 e opportunamente troncata su valori inferiori a 10^{-6}.

4. Modello Computazionale

Prima Ottimizzazione: In questo caso è stato modificato unicamente l'array relativo alle code associate al servente Spedizioni e Ritiri, garantendo una corretta gestione della priorità.

Seconda Ottimizzazione: La modifica del codice ha influenzato più funzioni, poiché c'è stata l'aggiunta e la gestione di una nuova coda e un nuovo servente, mantenendo lo stesso funzionamento di base dell'algoritmo Next event simulation.

5. Verifica

Verifichiamo in questa fase la corretta implementazione del nuovo centro che corrisponde ad un: singolo servente con singola coda infinita, modellato in questa fase come un M/M/1 di cui riportiamo dalla teoria delle code le relative formule matematiche:

$$E(Tq) = \frac{\rho E(s)}{1 - \rho} = 0{,}071428\, min$$

Dove: $\lambda_{locker} = 0{,}066666\, j/min$ $E(s) = \mu_{locker} = 1\, j/min$

Calcolando un intervallo di confidenza al 95% tramite la funzione estimate.py, sui 1024 valori medi raccolti all'interno di ogni batch, riscontriamo che la media teorica ricade al suo interno:

```
based upon 1024 data points and with 95 confidence
the expected value is in the interval 0.06841337 +/- 0.00330980
```

6. Validazione

Per validare il sistema, è stata aumentata la probabilità che i clienti di Spedizioni e Ritiri si rechino al Locker, impostandola al 100%, così che tutti coloro che devono effettuare questa operazione vi si dirigano. Inoltre, si è eliminata la possibilità di prenotazioni online per Spedizioni e Ritiri. Di conseguenza, il servente dedicato a questa classe di utenza gestisce esclusivamente Operazioni Generiche.
Dall'output, si osserva che il server 6 (dedicato a Spedizioni e Ritiri), evidenziato in rosso, ha lavorato in media in modo analogo ai serventi del multi-server (Dal server 1 al 5).
Precisiamo che per semplicità di lettura, l'output proposto è relativo ad una sola replica, corrispondente ad un solo seed, in particolare 123456789.

server	utilization	avg service	share
1	0.892	14.97	0.088
2	0.839	15.02	0.083
3	0.826	14.79	0.083
4	0.895	14.14	0.094
5	0.832	14.89	0.083
6	0.820	15.73	0.077
7	0.498	2.48	0.298
8	0.131	1.01	0.193

queue	avg wait t	avg delay t	avg service t	avg # in node	avg # in queue	avg # in service
1	16.63364134	0.87358073	15.76006061	0.80531240	0.04229413	0.76301827
2	18.77696677	3.16934254	15.60762423	0.90908081	0.15344270	0.75563811
3	21.00356303	6.41507676	14.58848627	5.16262535	1.57681046	3.58581489
7	3.17318010	0.62932746	2.54385264	0.11817576	0.02343745	0.09473831
8	3.87305838	1.41236939	2.46068899	0.63465897	0.23143800	0.40322096
9	1.02972822	0.02367963	1.00604859	0.13422220	0.00308657	0.13113563

```
Simulation complete. Clients served: [13, 13, 66, 0, 0, 0, 10, 44, 35]
```

Possiamo notare inoltre come, il tempo medio in coda (colore verde) per la classe d'utenza tre sia sceso a 6 minuti, mentre dell'output seguente, dove è stata ripristinata la probabilità di andare nel Locker al 50% ma mantenuta l'assenza di prenotazioni online, il tempo medio in coda per la classe tre sia di circa 11 minuiti. Un ulteriore differenza è l'incremento dell'utilizzazione del server 6 a circa 92%, dovendo, nella seconda figura, servire sia utenti Spedizioni e Ritiri che Operazione Generica.

server	utilization	avg service	share
1	0.892	14.97	0.089
2	0.884	14.83	0.089
3	0.871	14.62	0.089
4	0.938	14.81	0.095
5	0.874	14.67	0.089
6	0.920	9.15	0.151
7	0.498	2.48	0.302
8	0.063	1.00	0.095

queue	avg wait t	avg delay t	avg service t	avg # in node	avg # in queue	avg # in service
1	16.71489349	1.04883224	15.66606124	0.80924619	0.05077888	0.75846732
2	18.36154514	3.11831832	15.24322683	0.88896830	0.15097238	0.73799593
3	26.48585167	11.80707512	14.67877655	6.51015873	2.90215071	3.60800802
5	8.09123043	2.24057339	5.85065704	0.03013341	0.00834436	0.02178905
6	12.57658165	8.04111020	4.53547146	0.70256672	0.44920127	0.25336545
7	3.17318010	0.62932746	2.54385264	0.11817576	0.02343745	0.09473831
8	3.87305838	1.41236939	2.46068899	0.63465897	0.23143800	0.40322096
9	0.99629030	0.00000000	0.99629030	0.06307664	0.00000000	0.06307664

```
Simulation complete. Clients served: [13, 13, 66, 0, 1, 15, 10, 44, 17]
```

7. Design degli esperimenti

7.1 Analisi del transitorio

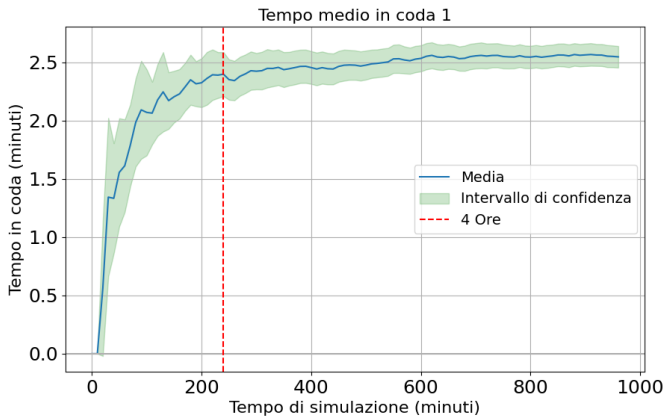
7.1.1 Primo Miglioramento

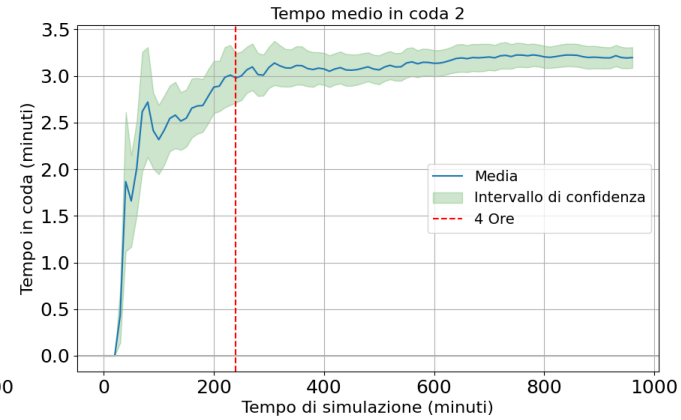
Mostriamo i grafici nella pagina seguente dell'analisi transitoria mediata su 100 repliche di una simulazione ad orizzonte finito, con un tempo di chiusura della filiale impostato a 960 minuti:

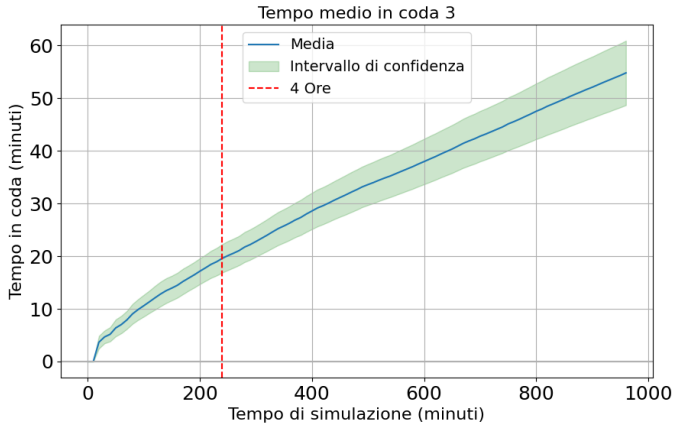
Dal primo miglioramento si evince come il tempo medio in coda per la classe di utenza tre sia effettivamente ridotto di circa 20 minuti dal modello precedente, ma a fronte di un aumento considerevole del tempo medio in coda per gli utenti senza priorità di Spedizioni e Ritiri.

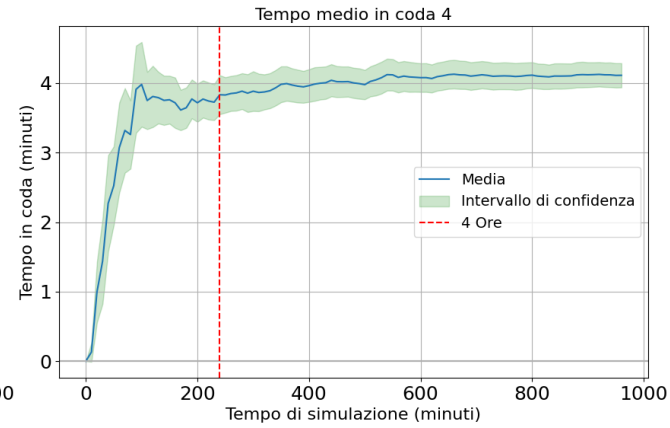
Questo accade poiché gli utenti che devono effettuare operazioni di Spedizioni e Ritiri hanno un tempo di servizio minore, ma in questo caso potrebbero trovare lo sportello occupato da un cliente Operazione Generica, che ha un tempo di servizio maggiore.

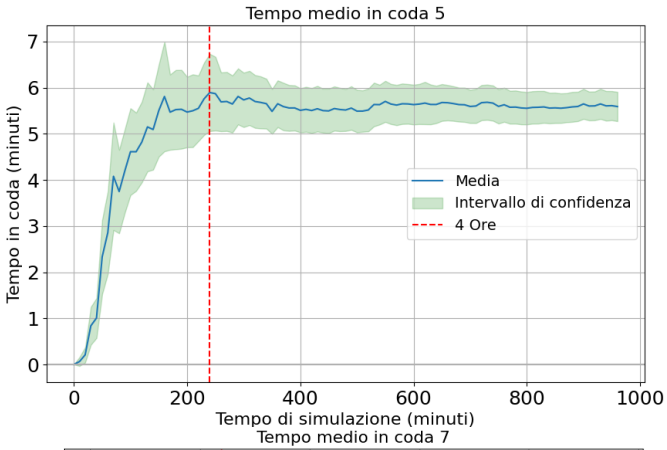
Possiamo notare ulteriormente come, in generale, i tempi medi in coda per le classi delle Operazioni Generiche siano tutti diminuiti, mentre per le operazioni di tipo Spedizioni e Ritiri sono tutti aumentati.

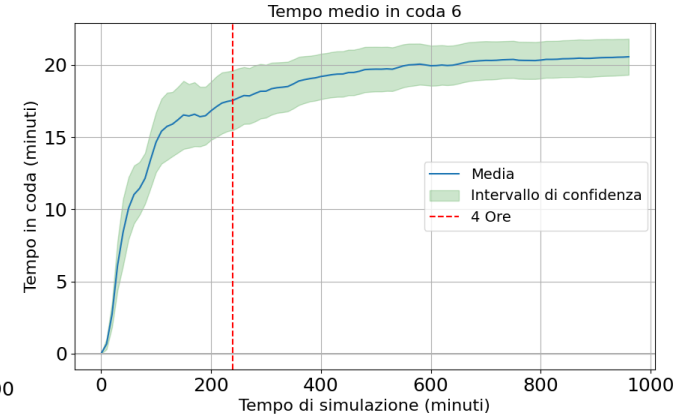

Tempo medio in coda 1
Tempo in coda (minuti)
Tempo di simulazione (minuti)
Media
Intervallo di confidenza
4 Ore

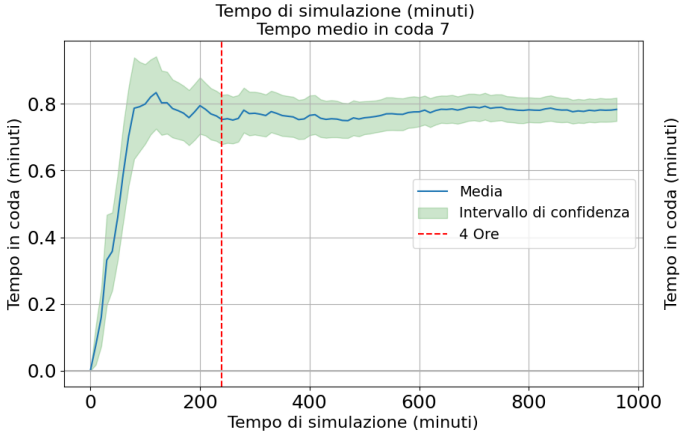

Tempo medio in coda 2
Tempo in coda (minuti)
Tempo di simulazione (minuti)
Media
Intervallo di confidenza
4 Ore

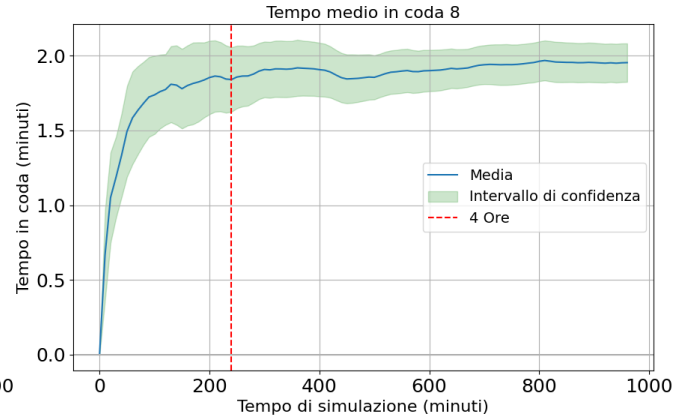

Tempo medio in coda 3
Tempo in coda (minuti)
Tempo di simulazione (minuti)
Media
Intervallo di confidenza
4 Ore


Tempo medio in coda 4
Tempo in coda (minuti)
Tempo di simulazione (minuti)
Media
Intervallo di confidenza
4 Ore


Tempo medio in coda 5
Tempo in coda (minuti)
Tempo di simulazione (minuti)
Media
Intervallo di confidenza
4 Ore


Tempo medio in coda 6
Tempo in coda (minuti)
Tempo di simulazione (minuti)
Media
Intervallo di confidenza
4 Ore

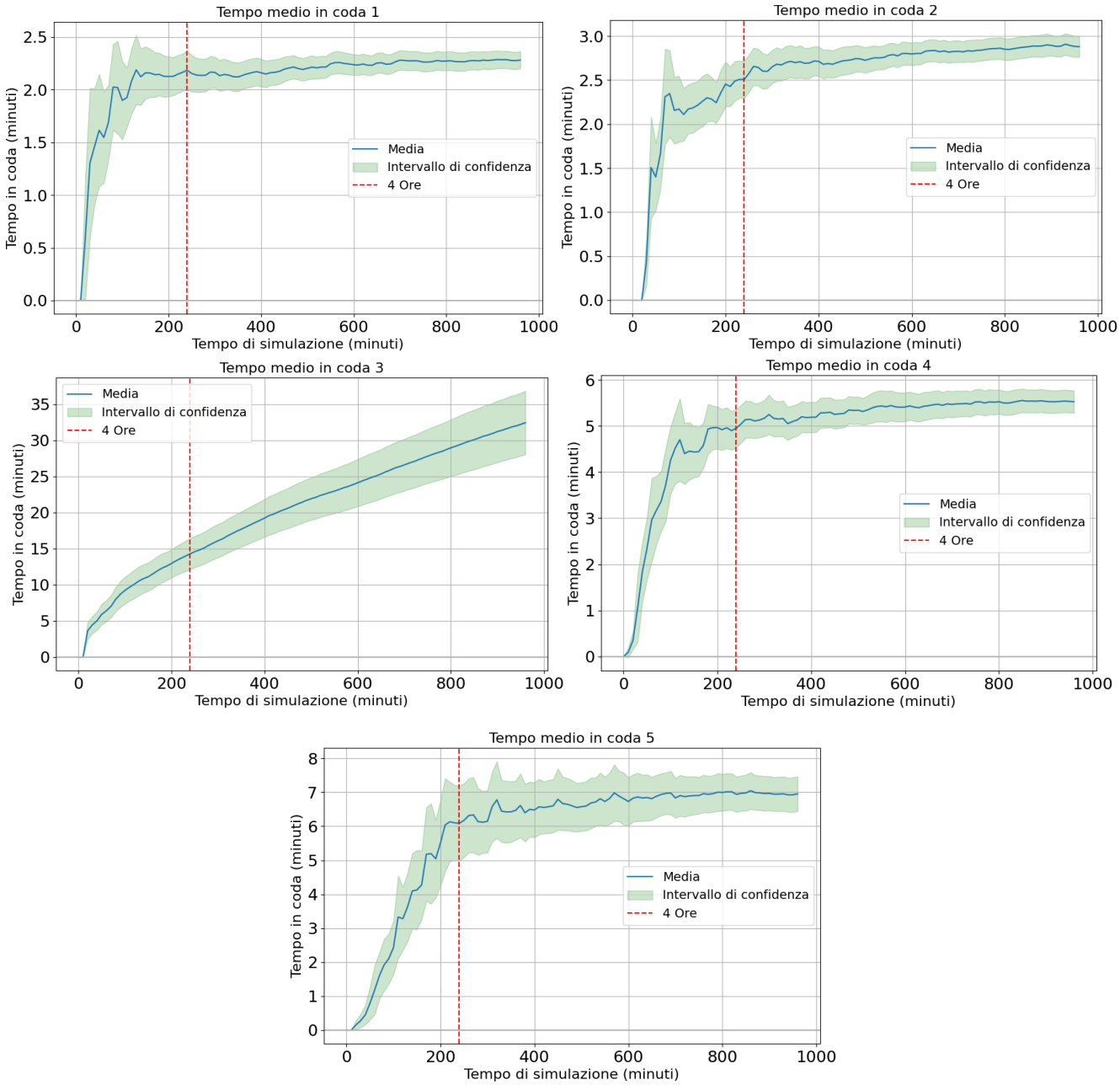

Tempo medio in coda 7
Tempo in coda (minuti)
Tempo di simulazione (minuti)
Media
Intervallo di confidenza
4 Ore

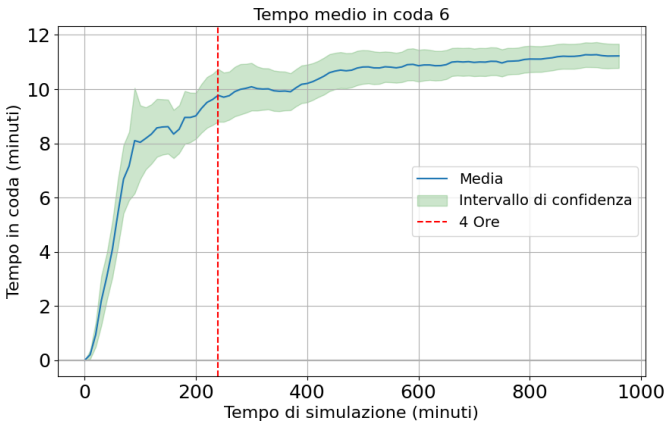

Tempo medio in coda 8
Tempo in coda (minuti)
Tempo di simulazione (minuti)
Media
Intervallo di confidenza
4 Ore

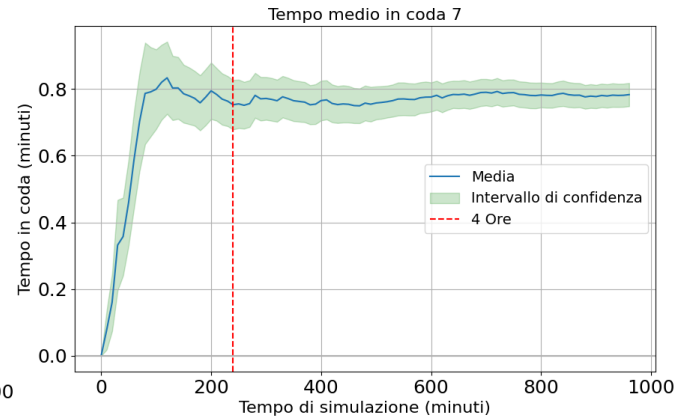
7.1.2 Secondo Miglioramento

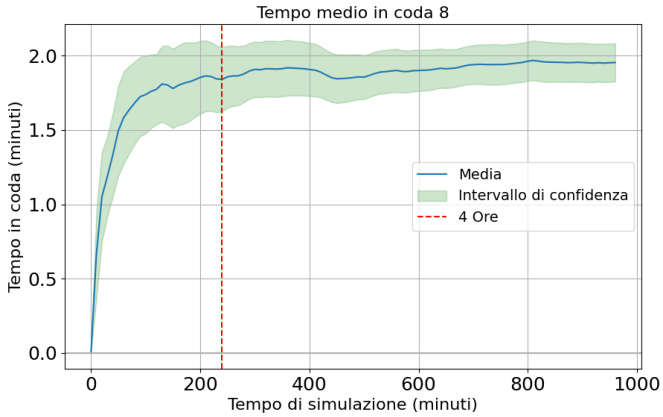
Introducendo il secondo miglioramento, metà del traffico delle operazioni Spedizione e Ritiri (senza prenotazione) è indirizzato verso il Locker.

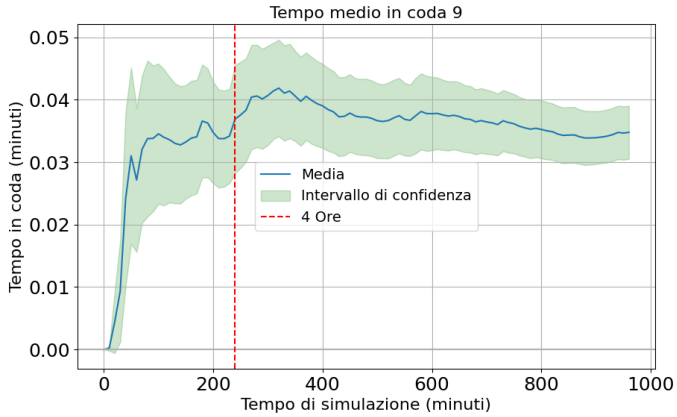
Il servente Spedizioni e Ritiri, con un traffico minore in ingresso, riesce ora a servire clienti per Operazioni Generiche con più frequenza, ne consegue una pendenza minore nella crescita dei tempi medi in coda per la classe 3, che purtroppo sembra non convergere.




Tempo medio in coda 6
Tempo in coda (minuti)
Tempo di simulazione (minuti)
Media
Intervallo di confidenza
4 Ore

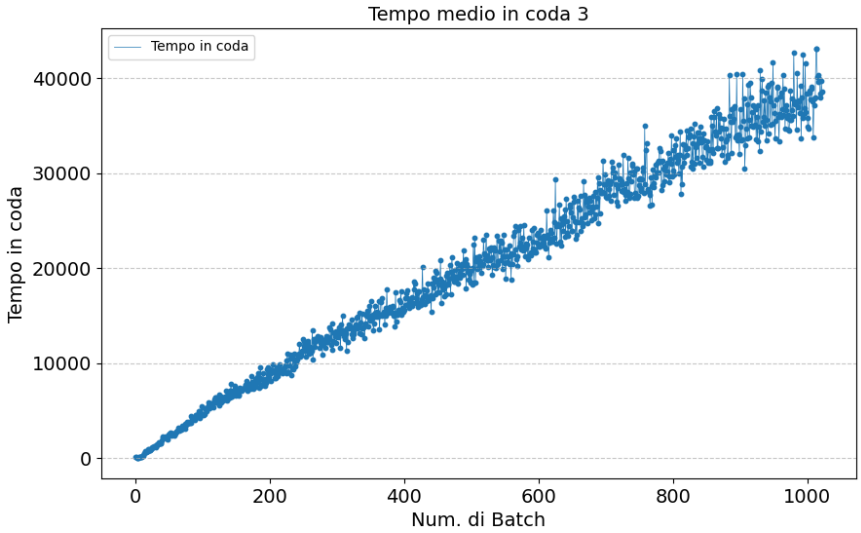

Tempo medio in coda 7
Tempo in coda (minuti)
Tempo di simulazione (minuti)
Media
Intervallo di confidenza
4 Ore

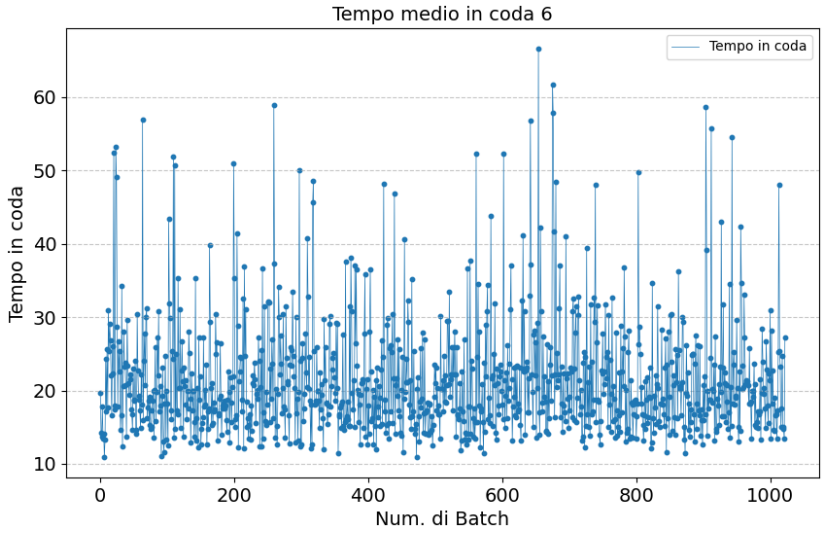

Tempo medio in coda 8
Tempo in coda (minuti)
Tempo di simulazione (minuti)
Media
Intervallo di confidenza
4 Ore

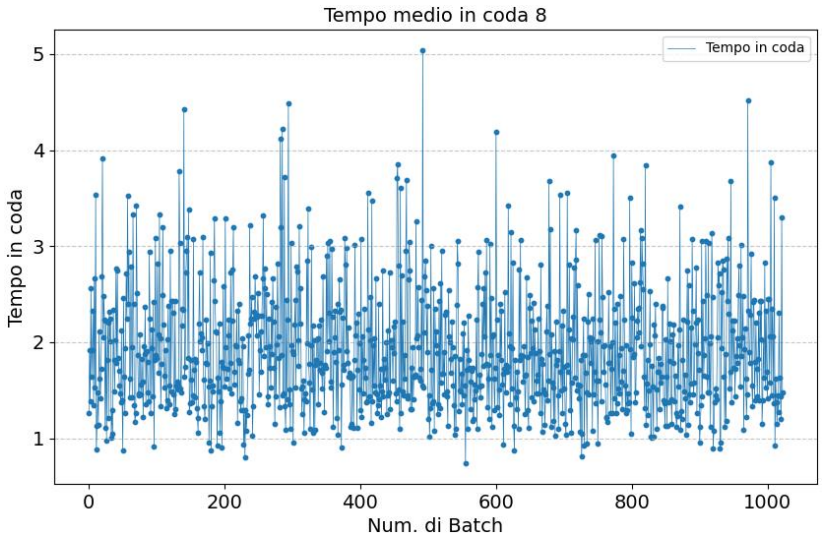

Tempo medio in coda 9
Tempo in coda (minuti)
Tempo di simulazione (minuti)
Media
Intervallo di confidenza
4 Ore

7.3 Analisi stazionaria

7.3.1 Primo miglioramento: Analizzando il comportamento del sistema per un periodo di tempo assimilabile all'infinito, nonostante il cambiamento apportato, la coda 3 non tende ancora ad uno stato stazionario. Mentre la coda 6, nonostante un incremento del tempo medio, converge ancora ad uno stato stazionario.

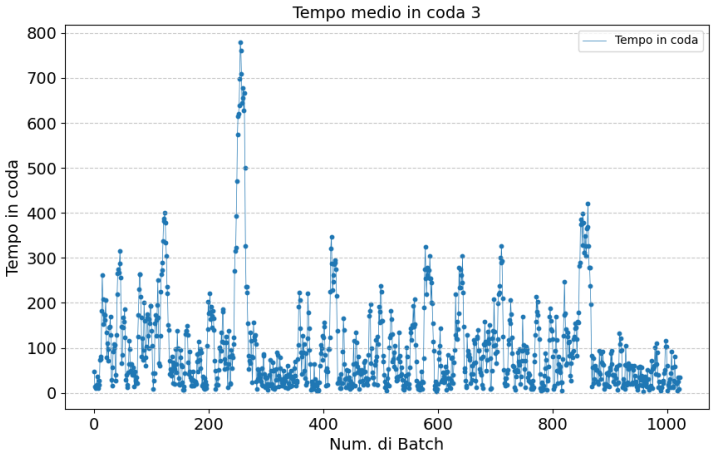


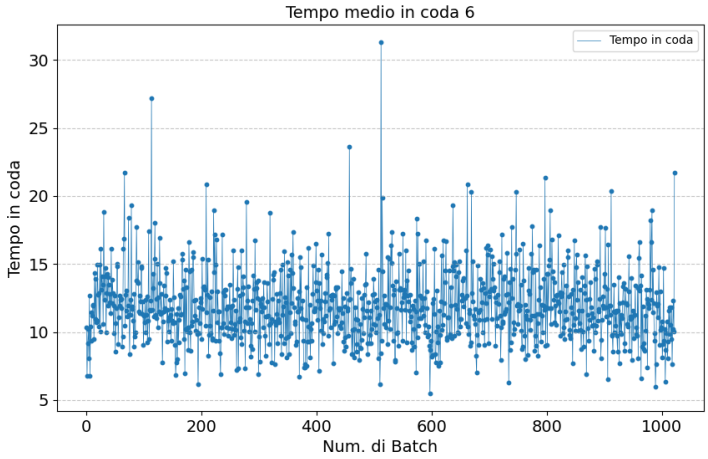


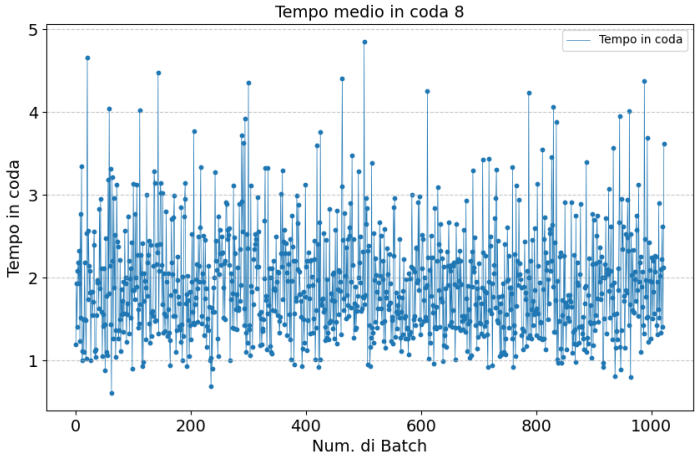


7.3.2 Secondo miglioramento: notiamo invece una convergenza allo stato stazionario per la coda 3, con un tempo medio intorno a 85 minuti calcolata con estimate.py sui valori ottenuti dalla simulazione con batch means.

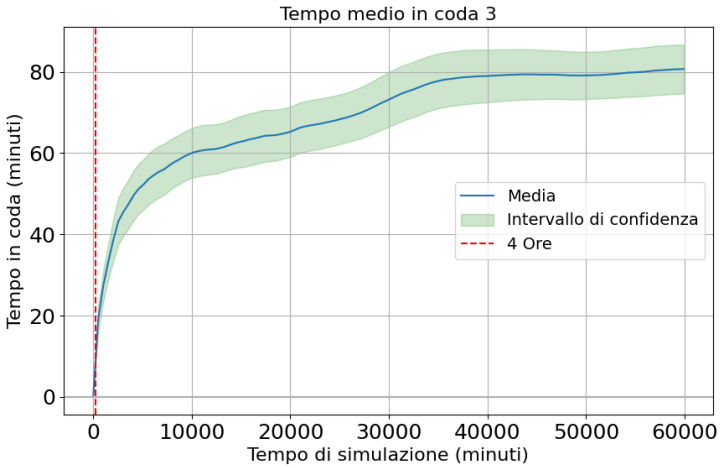
Questo vuol dire che con le 16 ore lavorative mostrate precedentemente nell'analisi transitoria non era visibile a pieno lo stato transitorio.

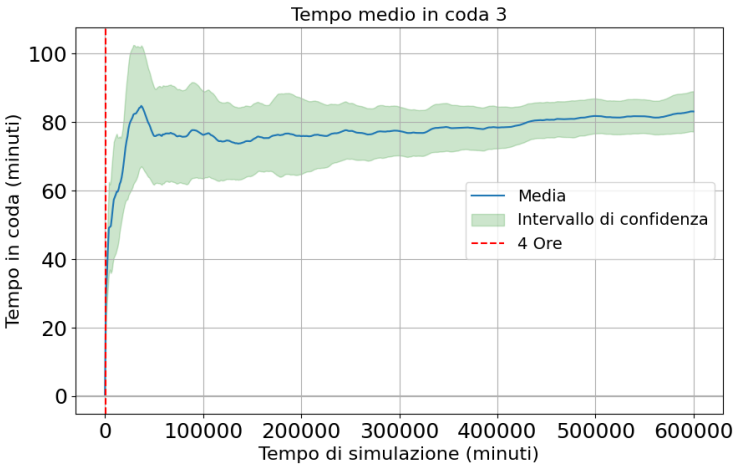






Seguono dunque i grafici dell'analisi del transitorio per la coda 3, dove questa volta viene eseguita nella la prima figura per 60 mila minuti, mentre la seconda a 600 mila minuti.





Si è effettuata inizialmente la simulazione con termine a 60 mila minuti, ma non avendo raggiunto la media stazionaria calcolata, si è proseguito nella seconda analisi, dove si nota una salita, corrispondente a quella del primo grafico, ma successivamente una ascesa.

Dopodiché il tempo in coda medio è tornato nuovamente a crescere fino ad un intorno di 85 minuti, e da ciò possiamo assumere di avere una visuale quasi completa dello stato transitorio.

7.4 Analisi degli obiettivi

Valutiamo il tempo medio in coda, mediato su 1000 repliche, con una simulazione a orizzonte finito e tempo limite di 4 ore, possiamo basarci sui grafici mostrati nell'analisi transitoria, dove la linea rossa tratteggiata rappresenta le 4 ore limite, in cui non vengono più accettati clienti nel sistema.

Con il primo miglioramento otteniamo una diminuzione del tempo di coda della classe d'utenza Operazioni Generiche senza priorità (Coda 3), andando però a sfavore della coda Spedizioni e Ritiri senza priorità (Coda 6), poiché i suoi tempi medi in coda aumentano, superando il QoS stabilito dall'azienda.

Otteniamo infatti:

E[Tq3]	17.19778805 +/- 0.74878457
E[Tq6]	17.43954114 +/- 0.51200833

Con il secondo miglioramento, tramite l'aggiunta del Locker otteniamo un tempo medio nelle code 3 e 6:

E[Tq3]	12.45096245 +/- 0.59794339
E[Tq6]	10.79945894 +/- 0.24375387

Ora tutte le classi di utenza rispettano il QoS stabilito, sul tempo medio in coda.

7.5 Utilizzazioni del sistema

Questi dati fanno riferimento ad uno studio ad orizzonte finito, con tempo di fine 240 minuti, mediato su mille repliche. Come spiegato precedentemente è stata valutata l'utilizzazione all'istante di chiusura delle porte della filiale, e non all'ultimo tempo di completamento, poiché altrimenti l'utilizzazione di alcuni serventi potrebbe essere intaccata dal fatto che loro non lavorino per quel periodo di tempo.

7.5.1 Primo Miglioramento

ρ Multi-Server	0.881680 ± 0.003502
ρ Spedizioni e Ritiri	0.938521 ± 0.002247
ρ ATM	0.574334 ± 0.004658

7.5.2 Secondo Miglioramento

ρ Multi-Server	0.865002 ± 0.003911
ρ Spedizioni e Ritiri	0.895639 ± 0.003336
ρ ATM	0. 574334 ± 0. 004658
ρ Locker	0.06626601 ± 0.001036

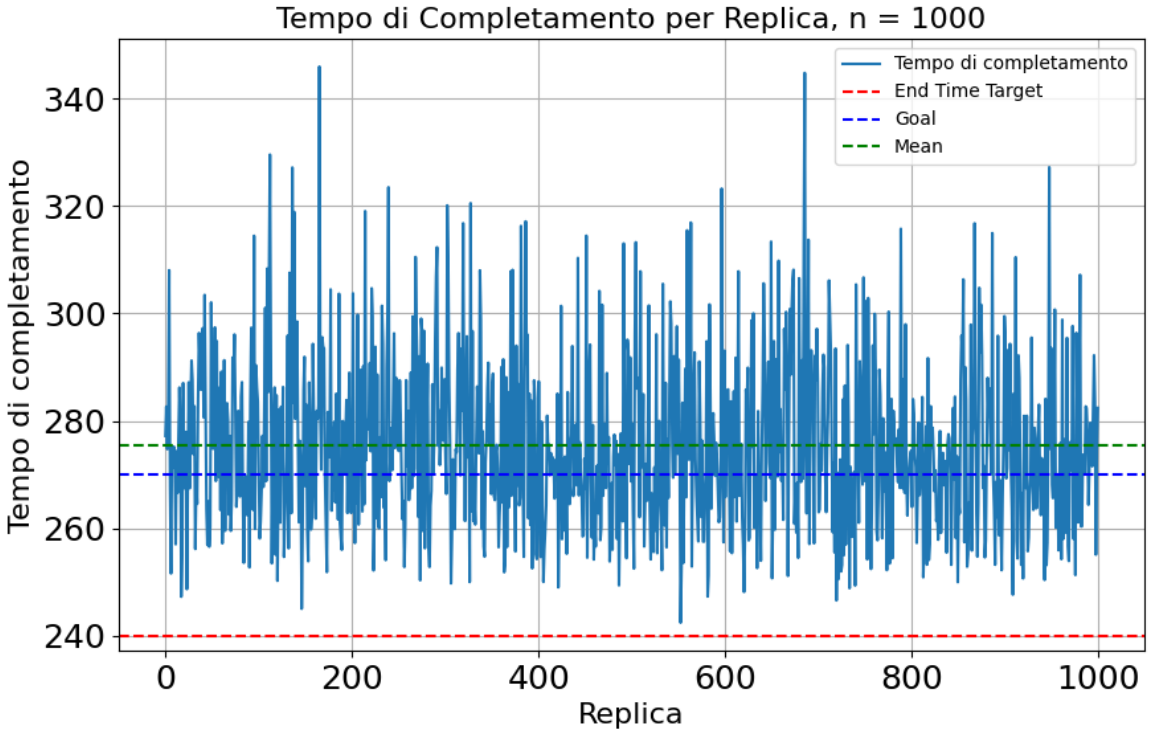
Notiamo come l'intero centro dell'ATM, coda 8, rimane invariato dal modello base ai modelli migliorativi, rispettando a pieno le aspettative, poiché questo centro non è stato influenzato dai miglioramenti. Nonostante ciò, i grafici sono stati inseriti per completezza.

7.6 Sovraccarico di lavoro

Ricordiamo il valore medio del tempo totale di lavoro del sistema, senza introdurre migliorie: 285 minuti, che si traducono in 45 minuti oltre il tempo desiderato di chiusura della filiale.

Nel capitolo che segue sono rappresentati, fronte dei nuovi miglioramenti, gli ultimi istanti di completamento delle 1000 repliche della simulazione ad orizzonte finito con un CLOSE_THE_DOOR_TIME = 240 min (4 ore).

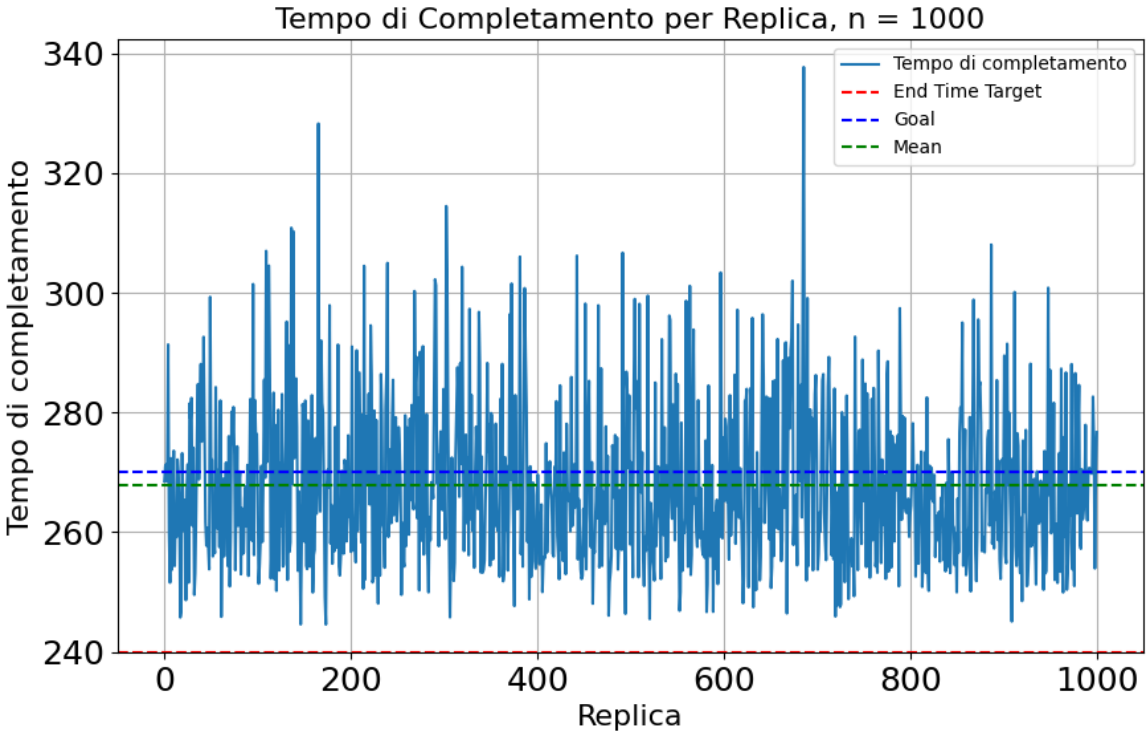
7.6.1 Primo Miglioramento




```
Basato su 1000 punti dati e con 95% di confidenza
Il valore atteso è nell'intervallo 275.54343220 +/- 0.98998217
```

Riscontriamo che in media, il carico lavorativo di una giornata eccede di 35 minuti. Rimane un valore elevato per l'obiettivo stabilito, ma comunque una riduzione di 10 minuti rispetto al modello iniziale.

7.6.2 Secondo Miglioramento




```
Basato su 1000 punti dati e con 95% di confidenza
Il valore atteso è nell'intervallo 267.92828650 +/- 0.84328156
```

Grazie al secondo miglioramento, siamo riusciti a raggiungere l'obiettivo proposto sul sovraccarico di lavoro, con una media di circa 28 minuti di supplemento.

Nonostante l'aggiunta di un servente siamo molto vicini al limite prefissato.

8. Conclusioni

In definitiva, analizzando le prestazioni senza l'aggiunta del Locker, possiamo affermare che l'utilizzo dello sportello Spedizioni e Ritiri per servire clienti che devono effettuare Operazioni Generiche gioverebbe con una diminuzione di circa 5 minuti per i clienti senza priorità. Tale diminuzione non basta per soddisfare a pieno il QoS, e purtroppo, sia ha un effetto collaterale sui tempi medi in coda per i clienti Spedizioni e Ritiri, talmente sentito da portare oltre il QoS anche i clienti senza priorità per tali operazioni. Questo effetto è dovuto ai tempi medi di servizio troppo diversi tra le due tipologie di operazione, un Operazione Generica richiede in media 15 minuti, mentre una Spedizione e Ritiri richiede in media 4.5 minuti, il che porta i clienti di Spedizioni e Ritiri in alcune situazioni, ad attendere un cliente che deve effettuare un Operazione Generica. È chiaro che un solo sportello non sia abbastanza adeguato a gestire i flussi delle due tipologie di clienti senza svantaggiare troppo le operazioni di Spedizioni e Ritiri.

L'introduzione di un Locker aiuta in modo considerevole il sistema nel problema sopra descritto, servendo metà del flusso di clienti per operazioni Spedizioni e Ritiri e quindi riducendo il carico per lo sportello dedicato a tali operazioni. Con la riduzione dell'affluenza di clienti allo sportello dedicato, questo potrà servire un numero più elevato di clienti per le Operazioni Generiche. Come ulteriore vantaggio, un minor numero di clienti Spedizioni e Ritiri, attenderà il servizio di una Operazione Generica, riducendo così il tempo medio di coda permettendo di rispettare i QoS.