GESTIONE PRONTO SOCCORSO TOR VERGATA

Eugenio Di Gaetano - 0349278, Marco Lorenzini - 0353515

AGENDA

1. Introduzione
2. Modello
3. Verifica
4. Validazione
5. Simulazione
6. Modello Migliorativo
7. Considerazioni sul personale
8. Considerazione sugli accessi
9. Risultati

1. INTRODUZIONE

Contesto

Funzione:

- Assistenza urgente e non programmabile
- Gestione pazienti in arrivo diretto o tramite ambulanza

Processo di Gestione:

- Triage Iniziale: Assegnazione priorità (Rosso, Arancione, Azzurro, Verde, Bianco)
- Prima Visita Medica: Valutazione e pianificazione trattamenti
- Ulteriori Esami: Svolgimento ed ulteriore visita
- Esito: Dimissione o ricovero per cure specialistiche

1. INTRODUZIONE

Problema

Dati di Attesa (Anno 2022) per il Pronto Soccorso del Policlinico Tor Vergata:

- Livello 2 (Arancione): 57 minuti
- Livello 3 (Azzurro): 164 minuti
- Livello 4 (Verde): 154 minuti
- Livello 5 (Bianco): 148 minuti

Criticità Identificate:

- Tempi di attesa superiori ai limiti stabiliti dalle linee guida nazionali
- Disallineamento tra domanda di servizi e capacità di risposta
- Impatto negativo su qualità delle cure e sui risultati clinici, specialmente per urgenze intermedie

1. INTRODUZIONE

Obiettivo

Rispettare le Nuove Linee Guida (2019):

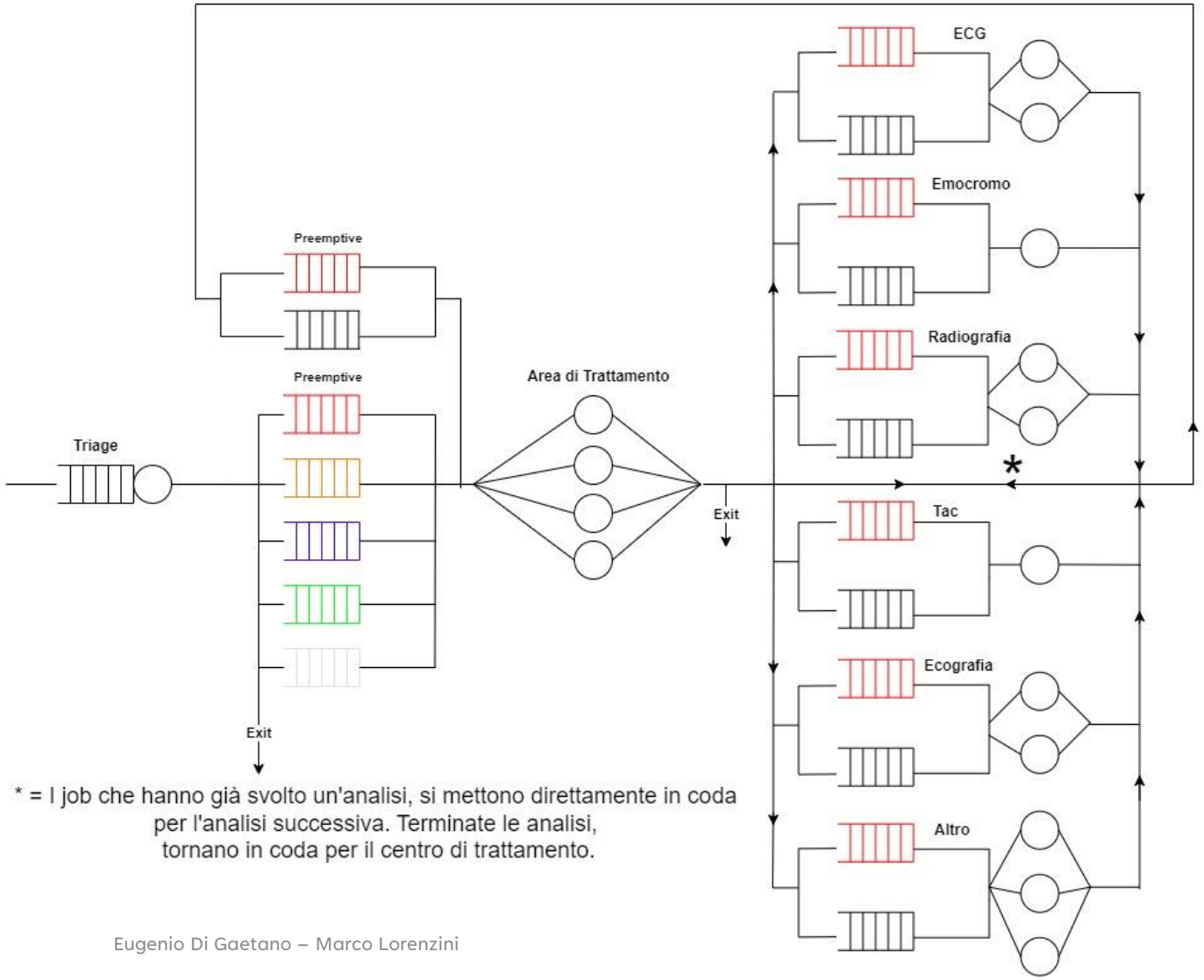
- Livello 1 (Rosso): Accesso immediato
- Livello 2 (Arancione): Accesso entro 15 minuti
- Livello 3 (Azzurro): Accesso entro 60 minuti
- Livello 4 (Verde): Accesso entro 120 minuti
- Livello 5 (Bianco): Accesso entro 240 minuti

Strategie di Miglioramento:

- Analizzare e ottimizzare i processi interni del Pronto Soccorso
- Implementare miglioramenti senza aumentare il numero di medici
- Migliorare l'efficienza operativa e la gestione dei flussi di pazienti

2. MODELLO

Modello concettuale



Astrazione:

- Job = Paziente
- Sistema = Insieme delle aree
- Server Triage = Infermieri
- Server Area di trattamento = Medici
- Server Area di analisi = Specialisti

Eventi:

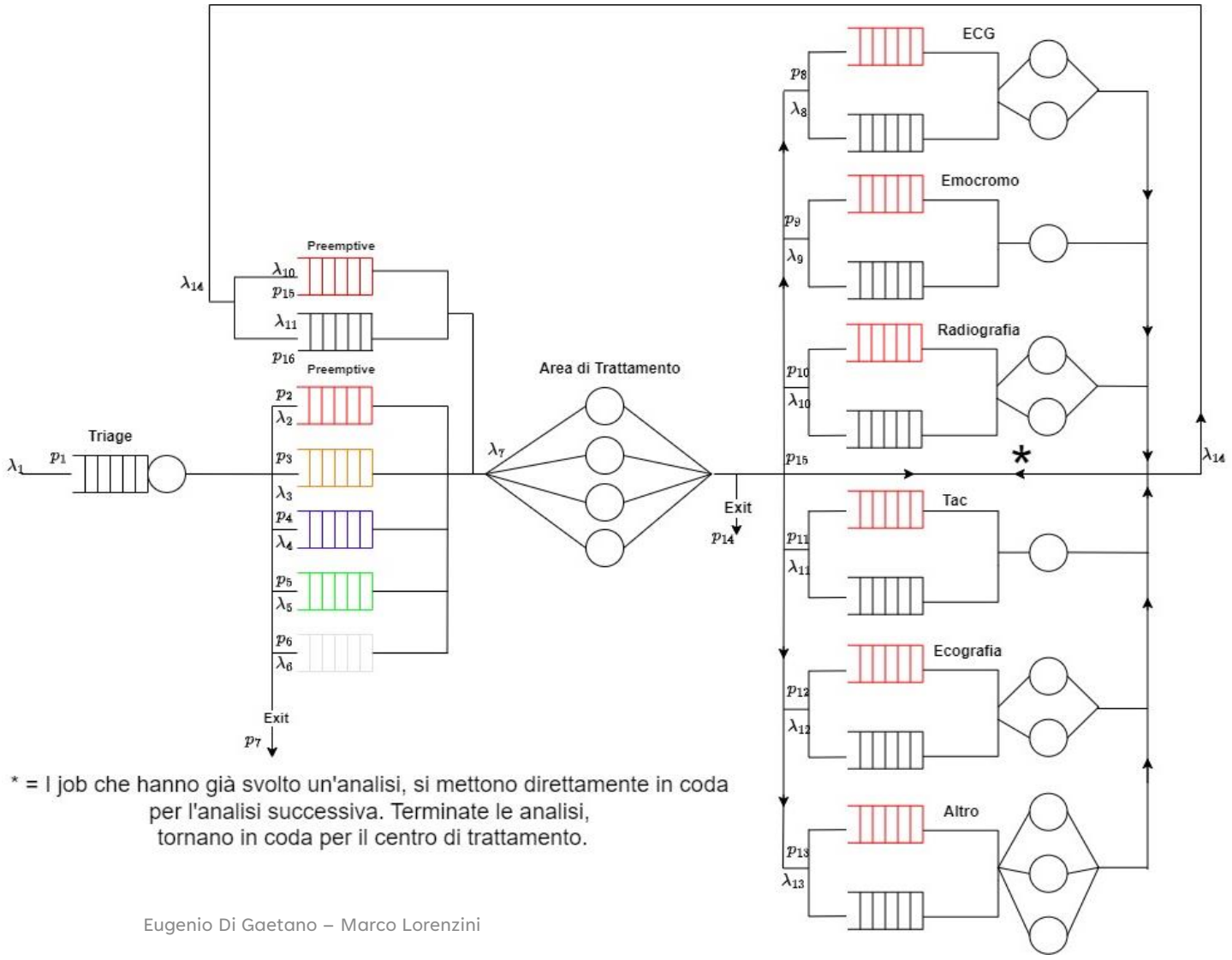
- Ingresso in un centro
- Uscita da un centro

Procedura:

1. Assegnazione codice
2. Prima visita
   - Assegnazione Esami
   - Dimissioni / Ospedalizzazione
3. Svolgimento Esami
4. Visita
   - Ulteriori esami -> 3.
   - Dimissioni / Ospedalizzazione

2. MODELLO

Modello delle specifiche



$$\lambda_2 = \lambda_1 \cdot p_2$$
$$\lambda_3 = \lambda_1 \cdot p_3$$
$$\lambda_4 = \lambda_1 \cdot p_4$$
$$\lambda_5 = \lambda_1 \cdot p_5$$
$$\lambda_6 = \lambda_1 \cdot p_6$$
$$\lambda_7 = \lambda_1 \cdot (1 - p_7) + \lambda_{14}$$
$$\lambda_8 = \lambda_7 \cdot p_8$$
$$\lambda_9 = \lambda_7 \cdot p_9$$
$$\lambda_{10} = \lambda_7 \cdot p_{10}$$
$$\lambda_{11} = \lambda_7 \cdot p_{11}$$
$$\lambda_{12} = \lambda_7 \cdot p_{12}$$
$$\lambda_{13} = \lambda_7 \cdot p_{13}$$
$$\lambda_{14} = \lambda_7 \cdot p_{15}$$
$$p_8 + p_9 + p_{10} + p_{11} + p_{12} + p_{13} = p_{15}$$
$$p_{14} + p_{15} = 1$$

2. MODELLO

Modello computazionale

Struttura:

- Directory controller: Logica della simulazione (gestione del Triage, del Trattamento e delle Aree di Analisi).
- Directory model: Implementazione del singolo job come classe.
- Directory utility: Facility e librerie esterne utilizzate.

Pseudo Random Numbers Generator:

- Algoritmo di Lehmer
- Approccio multi stream

Tempi di Servizo:

- Normale Troncata

```
def idfTruncatedNormal (m, s, lower_bound, upper_bound)
    a = cdfNormal(m, s, lower_bound)
    b = 1.0-cdfNormal(m, s, upper_bound)
    u = idfUniform(a, 1.0-b, random())
    return idfNormal(m, s, u)
```

2. MODELLO

Modello computazionale

Logica degli eventi

Selezione prossimo evento

```
def next_event(current_triage, current_queue, t_analisi):
    prox_evento = INFINITY
    for i in range(len(t_analisi)):
        if t_analisi[i].current > 0:
            prox_evento = min(prox_evento, t_analisi[i].current)
    if current_queue > 0:
        prox_evento = min(prox_evento, current_queue)
    if current_triage > 0:
        prox_evento = min(prox_evento, current_triage)
    if prox_evento >= INFINITY:
        prox_evento = -1
    return prox_evento
```

Esecuzione prossimo evento

```
def switch(prox_operazione, t_triage, t_queue, t_analisi):
    t_triage.current = prox_operazione
    t_queue.current = prox_operazione
    for i in range(len(t_analisi)):
        t_analisi[i].current = prox_operazione
    if prox_operazione == t_triage.arrival:
        processa_arrivo_triage()
    elif prox_operazione == t_triage.min_completion:
        processa_completamento_triage()
    elif prox_operazione == t_queue.min_completion:
        processa_completamento_queue()
    else:
        for i in range(len(t_analisi)):
            if prox_operazione == t_analisi[i].min_completion:
                processa_completamento_analisi(i)
                break
```

3. VERIFICA

Assunzioni

- Tempi di servizio esponenziali.
- Code uniche senza distinzione di colore.
- Il paziente non può spostarsi tra i vari centri di analisi.

Risultati analitici

CENTRI	ρ	$E[Tq]$ (min)	$E[Ts]$ (min)
Triage	0.5	5.65	11.15
Area di Trattamento	0.23	52.42	68.42
ECG	0.07	0.02	5.02
Emocromo	0.17	1.02	6.02
Radiografia	0.29	2.27	27.27
Tac	0.18	5.52	30.52
Ecografia	0.1	0.19	15.19
Altro	0.43	0.79	35.79

Risultati simulazione

CENTRI	ρ	$E[Tq]$ (min)	$E[Ts]$ (min)
Triage	0.5	5.35	10.85
Area di Trattamento	0.93	52.62	68.62
ECG	0.07	0.08	5.08
Emocromo	0.17	1.07	6.07
Radiografia	0.33	1.69	26.69
Tac	0.18	6.18	31.18
Ecografia	0.11	0.59	15.59
Altro	0.54	0.79	39.02

4. VALIDAZIONE

Considerazioni

La complessità del sistema con molteplici centri e feedback rende necessarie delle semplificazioni.

La differenza tra i dati del report e i risultati della simulazione è attribuibile a variabilità e fattori non simulabili.

Report		Risultati simulazione	
Codice rosso:	0 *minuti*	Codice rosso:	0 ± 0 *minuti*
Codice arancione:	57 *minuti*	Codice arancione:	67.94 ± 6.03 *minuti*
Codice azzurro:	164 *minuti*	Codice azzurro:	124.60 ± 7.23 *minuti*
Codice verde:	154 *minuti*	Codice verde:	251.32 ± 8.99 *minuti*
Codice bianco:	148 *minuti*	Codice bianco:	541.64 ± 79.74 *minuti*

5. SIMULAZIONE

Analisi del transitorio

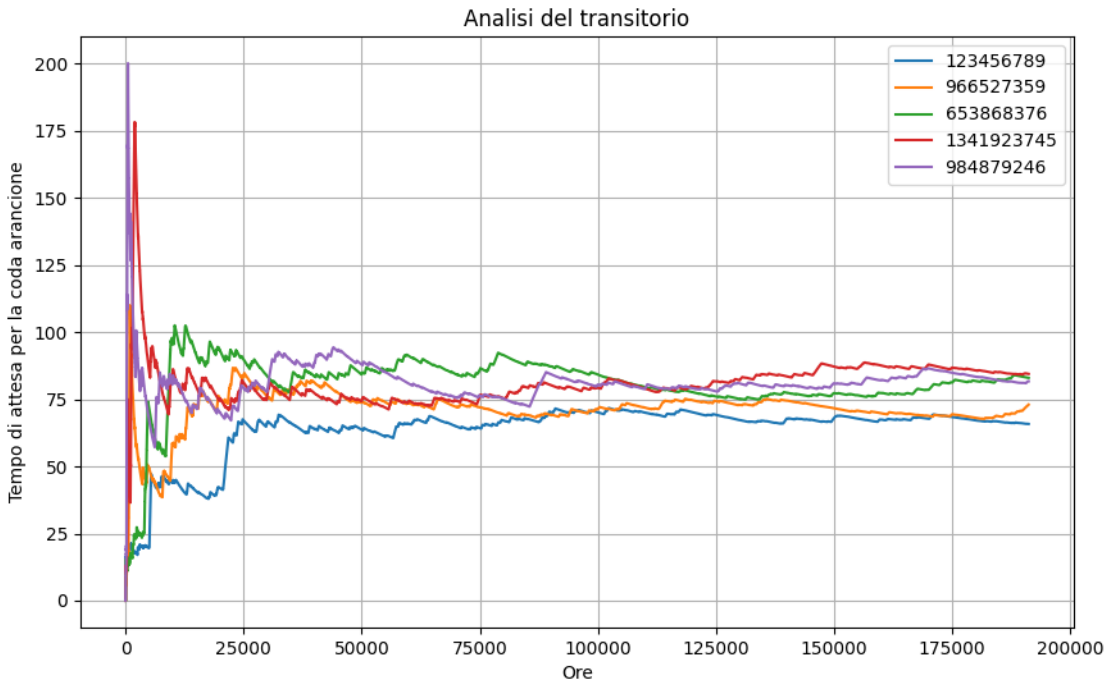
- Verifichiamo se il sistema converge allo stato stazionario

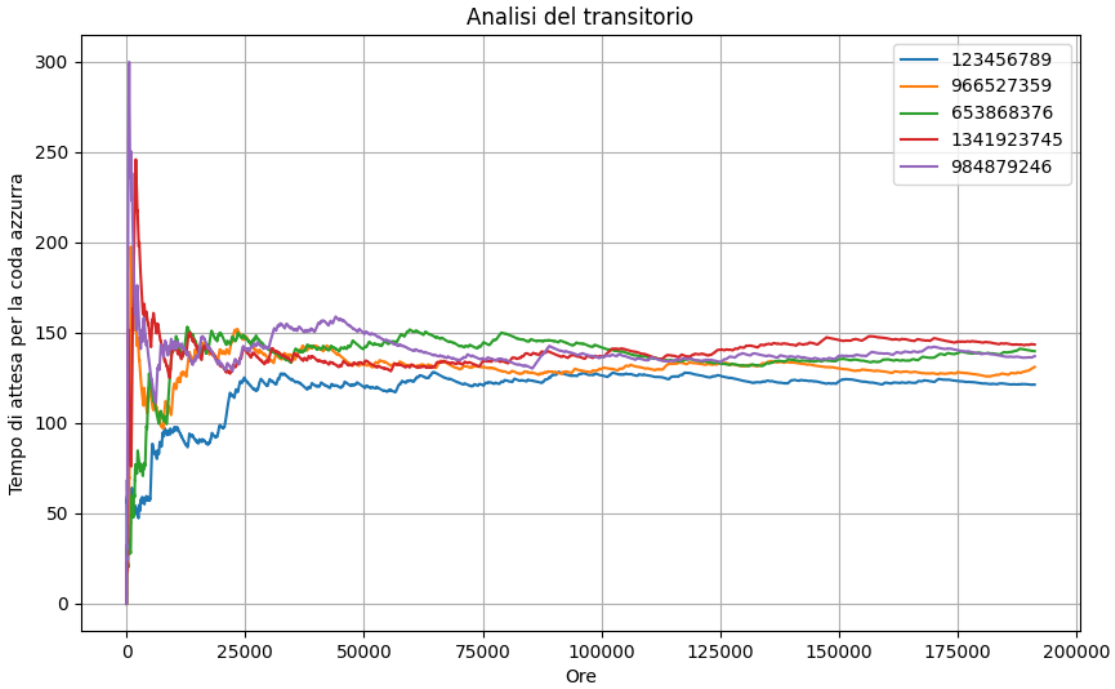
- La variabile di riferimento è il tempo di attesa del centro di trattamento

- Effettuiamo più simulazioni:
  - Stesse condizioni iniziali
  - Seed indipendenti

5. SIMULAZIONE

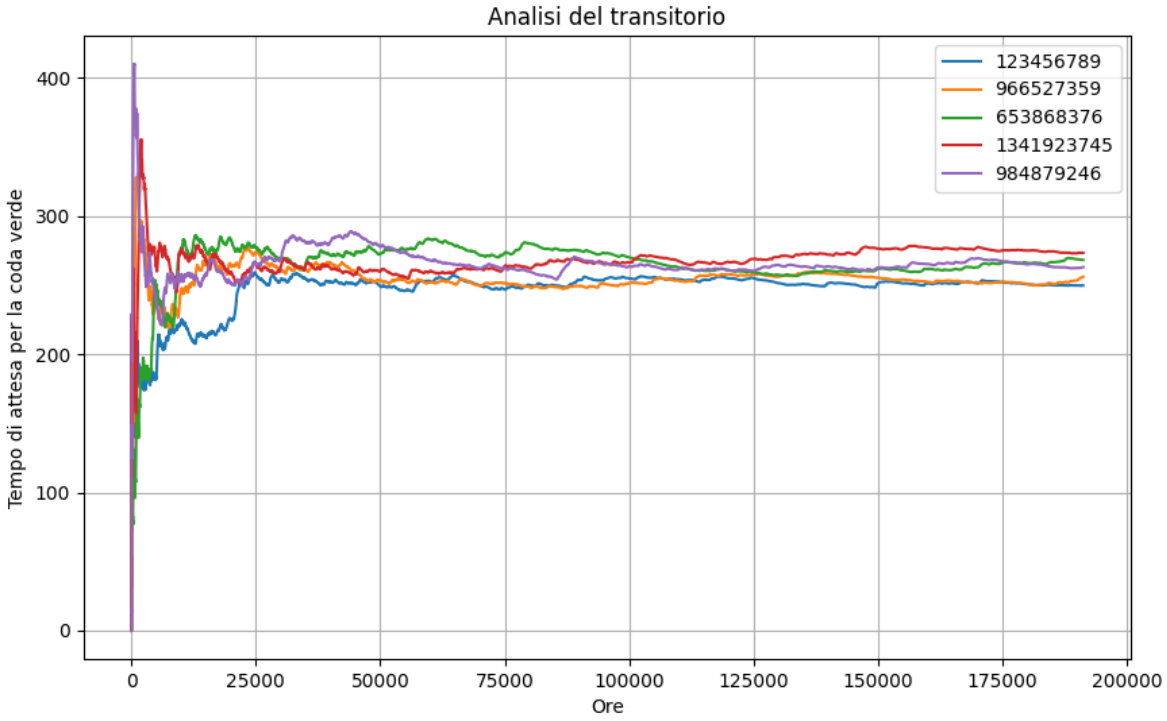
Analisi del transitorio

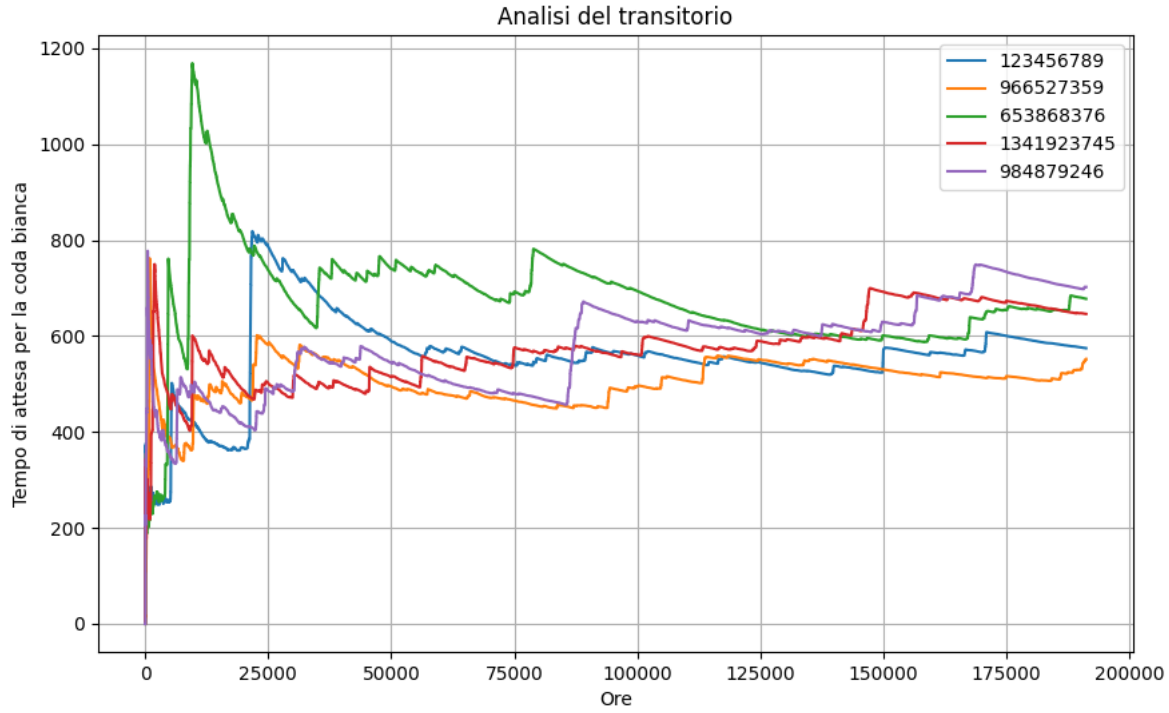




5. SIMULAZIONE

Analisi del transitorio





5. SIMULAZIONE

Simulazione ad orizzonte finito

In generale:

- Obiettivo: Studio del comportamento del sistema durante il transitorio

Nel nostro caso:

- Qos da verificare dopo aver raggiunto lo stato stazionario
- Simulazione ad orizzonte finito non adatta al raggiungimento dei nostri obiettivi

5. SIMULAZIONE

Simulazione ad orizzonte infinito

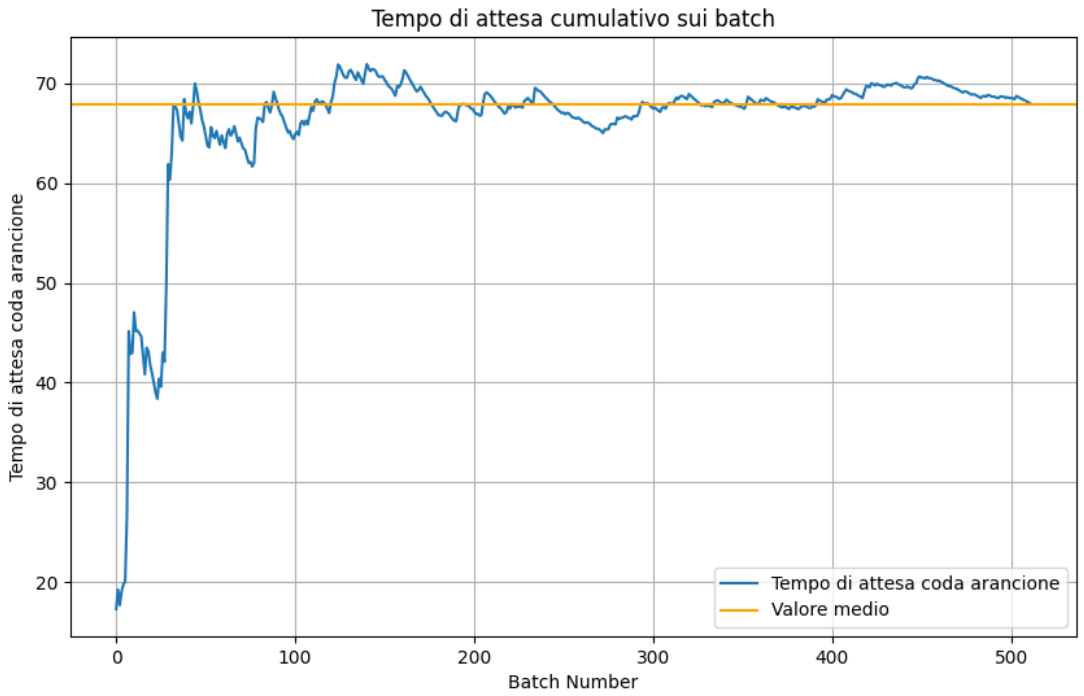
- Tecnica batch means
- Elementi per batch(k): 4096
- Numero di batch(b): 512
- Intervallo di confidenza: 95%

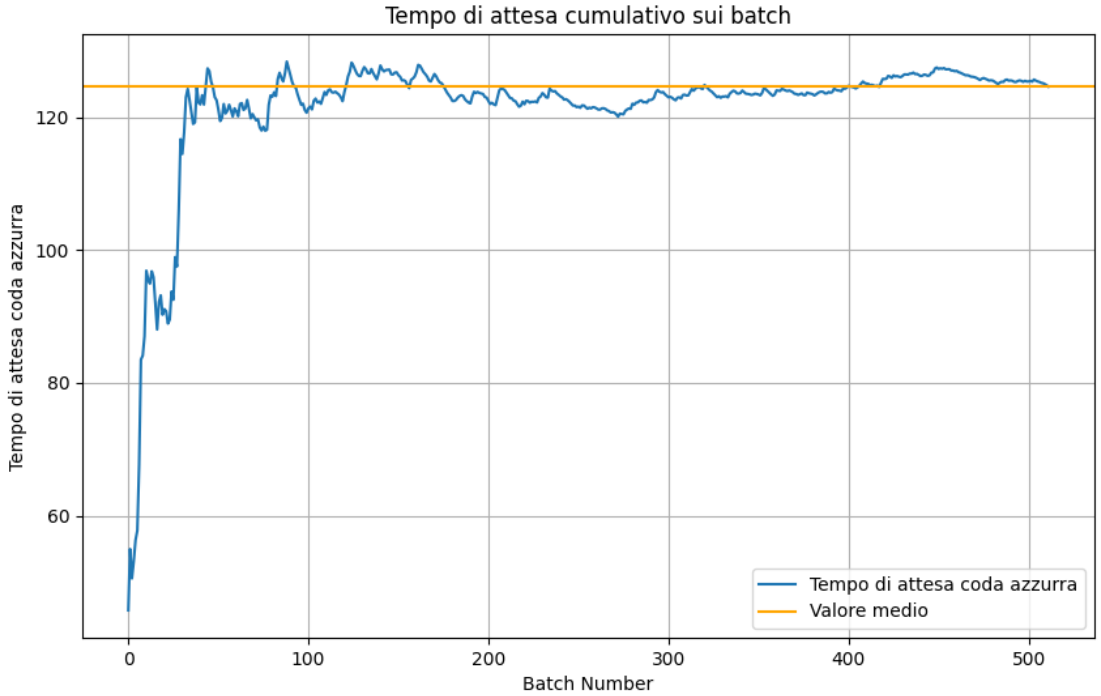
Table 1: Tempi di attesa e risposta nelle code

Colore	**Attesa**	**Risposta**
Rosso	0.0005 +/- 0.0003	16.5329 +/- 0.0187
Arancione	67.9448 +/- 6.0376	84.5434 +/- 6.0391
Azzurro	124.6084 +/- 7.2344	141.833 +/- 7.2399
Verde	251.3208 +/- 8.9988	270.7792 +/- 9.0096
Bianco	541.6481 +/- 79.7361	565.7939 +/- 80.0485

5. SIMULAZIONE

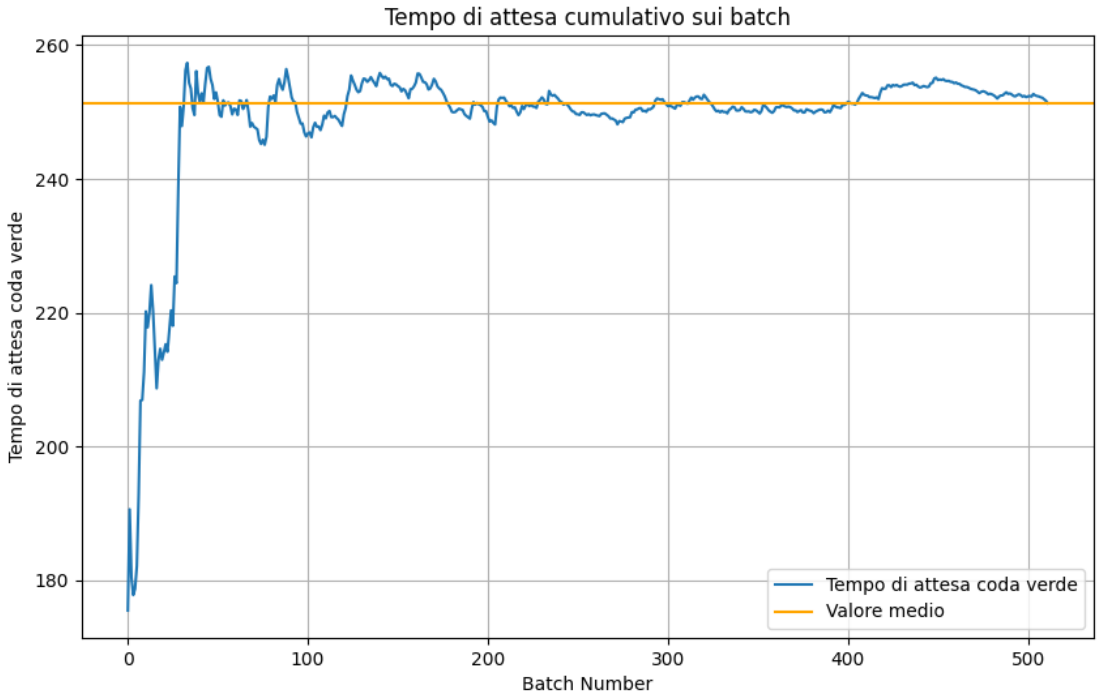
Simulazione ad orizzonte infinito

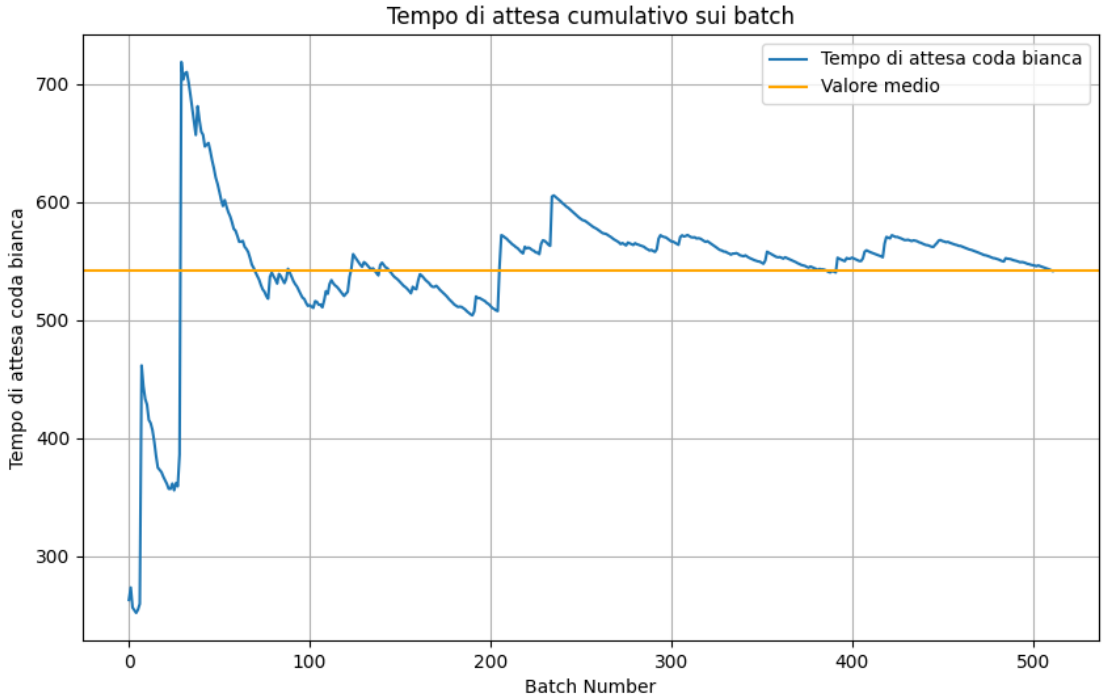




5. SIMULAZIONE

Simulazione ad orizzonte infinito





5. SIMULAZIONE

Considerazioni

Elevata percentuale di pazienti non trattati entro i tempi limite stabiliti:

- Codice arancione: 45.50%
- Codice azzurro: 40.29%
- Codice verde: 61.33%
- Codice bianco: 66.35%

6. MODELLO MIGLIORATIVO

Introduzione

- Risorse inalterate
- Nuova policy di scheduling
- Nuovo tempo limite

```
def get_job_old(list_of_queues, t):
     """Selects the next job to serve based on a priority policy"""
    if Parameters.migliorativo:
        for queue in list_of_queues:
            if queue and queue[0] and (t.current - queue[0].get_id()) > OBIETTIVO_MIGLIORATIVO[
                queue[0].get_codice() - 1]:
                return queue.pop(0)
```

6. MODELLO MIGLIORATIVO

Analisi del transitorio

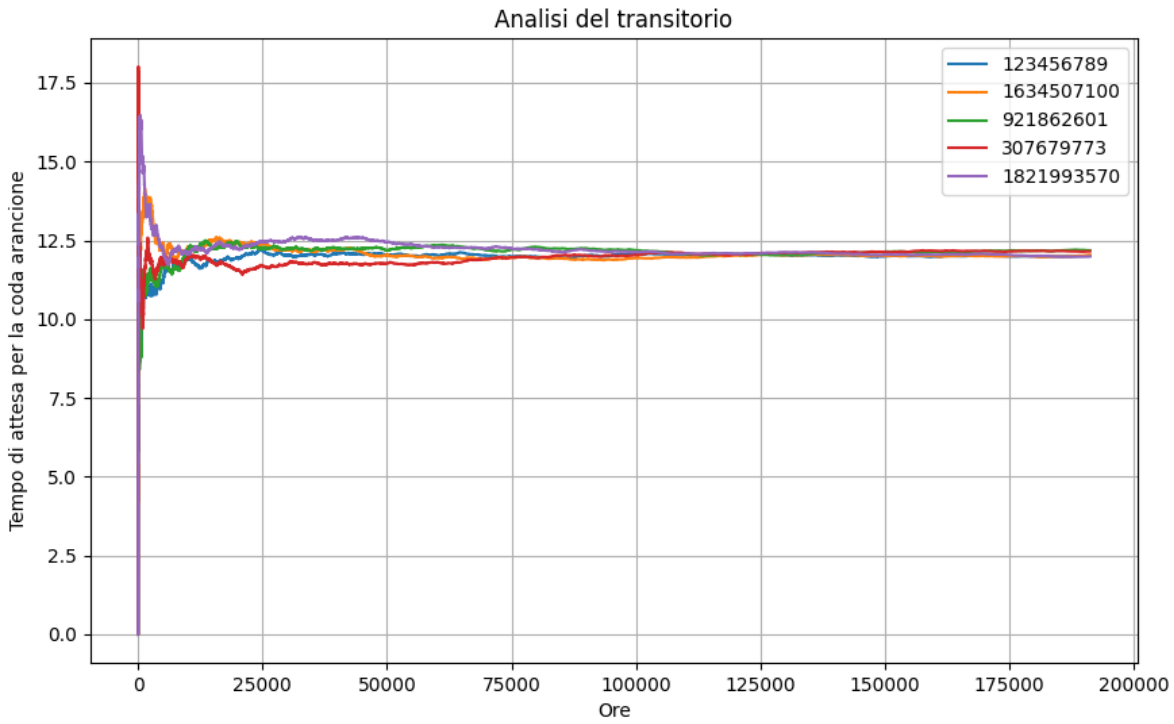
- Verifichiamo se il sistema converge allo stato stazionario

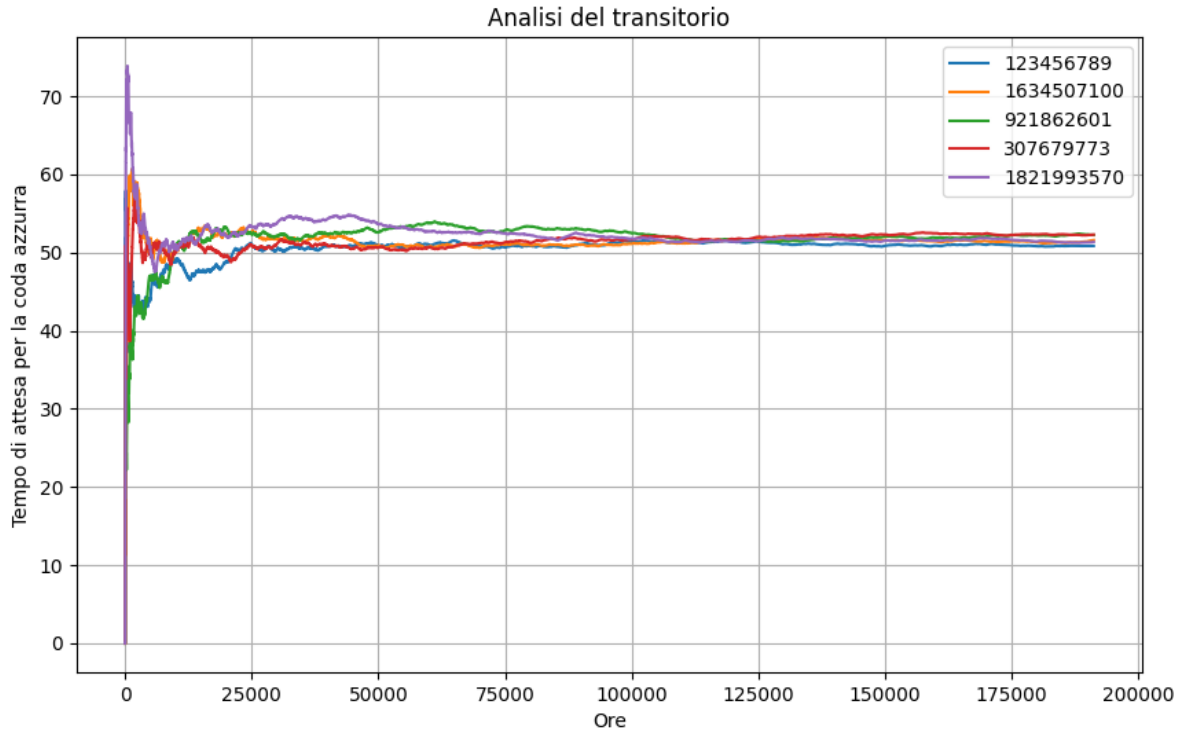
- La variabile di riferimento è il tempo di attesa del centro di trattamento

- Effettuiamo più simulazioni:
  - Stesse condizioni iniziali
  - Seed indipendenti

6. MODELLO MIGLIORATIVO

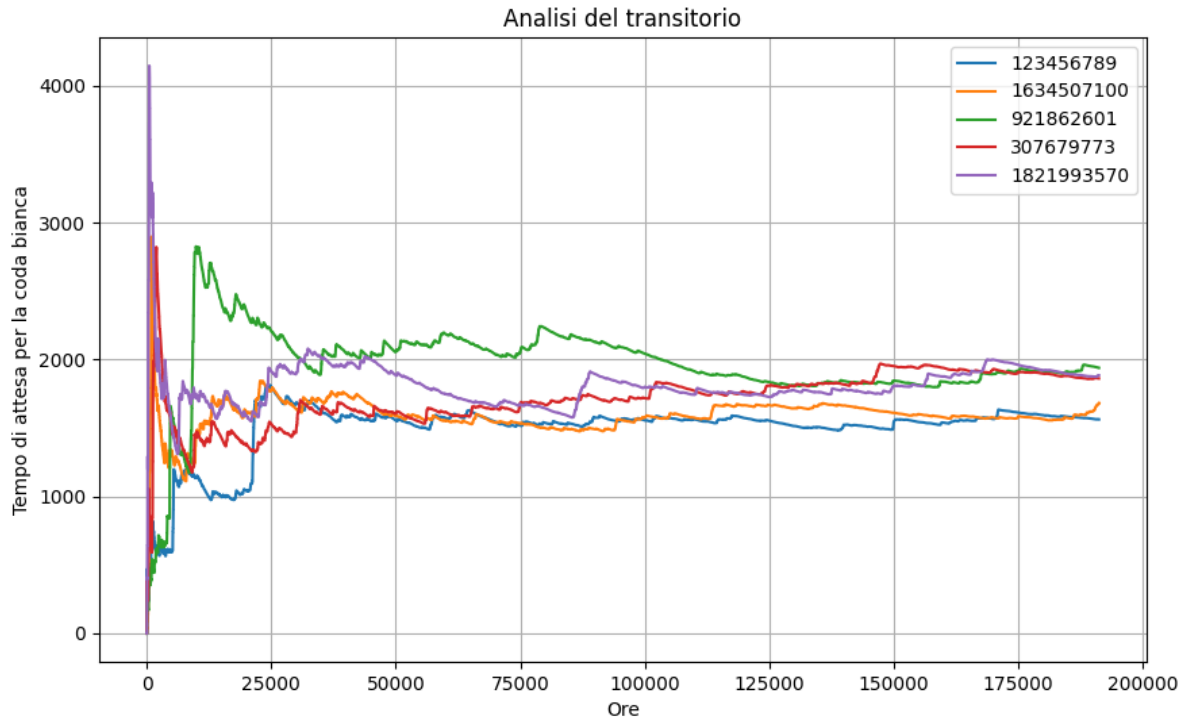
Analisi del transitorio





6. MODELLO MIGLIORATIVO

Analisi del transitorio



6. MODELLO MIGLIORATIVO

Simulazione ad orizzonte infinito

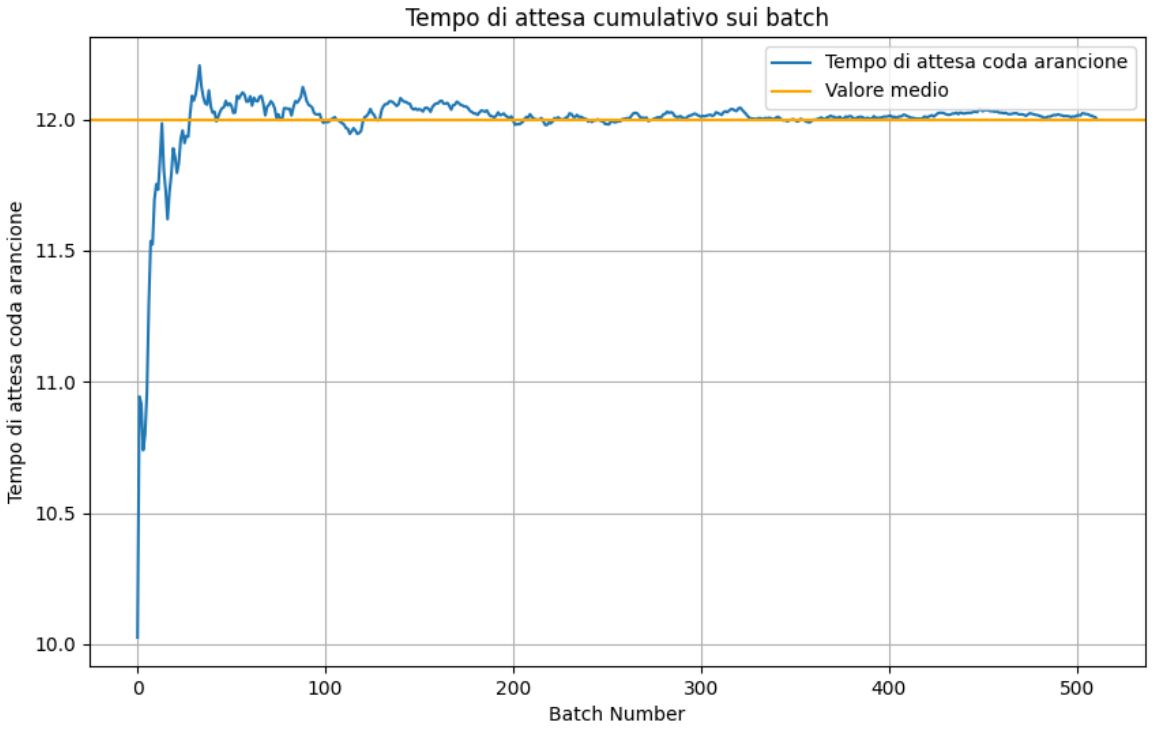
- Tecnica batch means
- Elementi per batch(k): 4096
- Numero di batch(b): 512
- Intervallo di confidenza: 95%

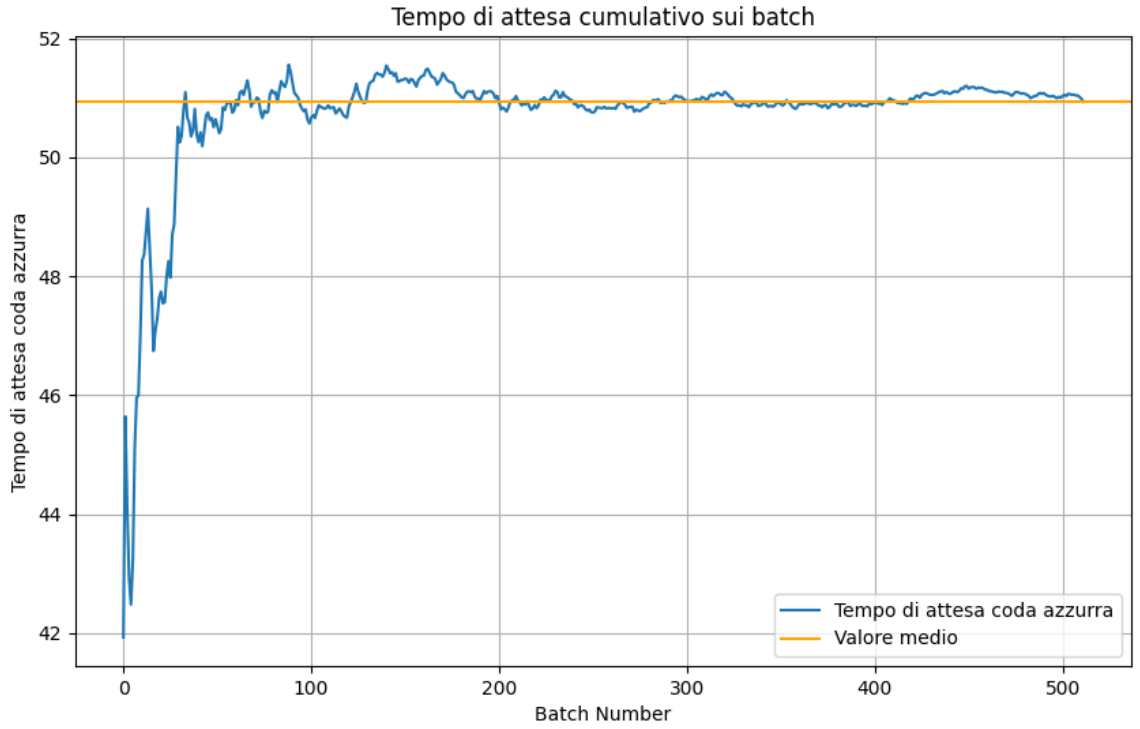
Table 4: Tempi di attesa e risposta nelle code

Colore	**Attesa**	**Risposta**
Rosso	0.0002 +/- 0.0001	16.53 +/- 0.1934
Arancione	11.9988 +/- 0.1295	28.5612 +/- 0.1311
Azzurro	50.9357 +/- 0.8525	68.0385 +/- 0.8603
Verde	204.3922 +/- 6.8145	224.5949 +/- 6.8523
Bianco	1531.8898 +/- 149.6749	1570.3316 +/- 150.1254

6. MODELLO MIGLIORATIVO

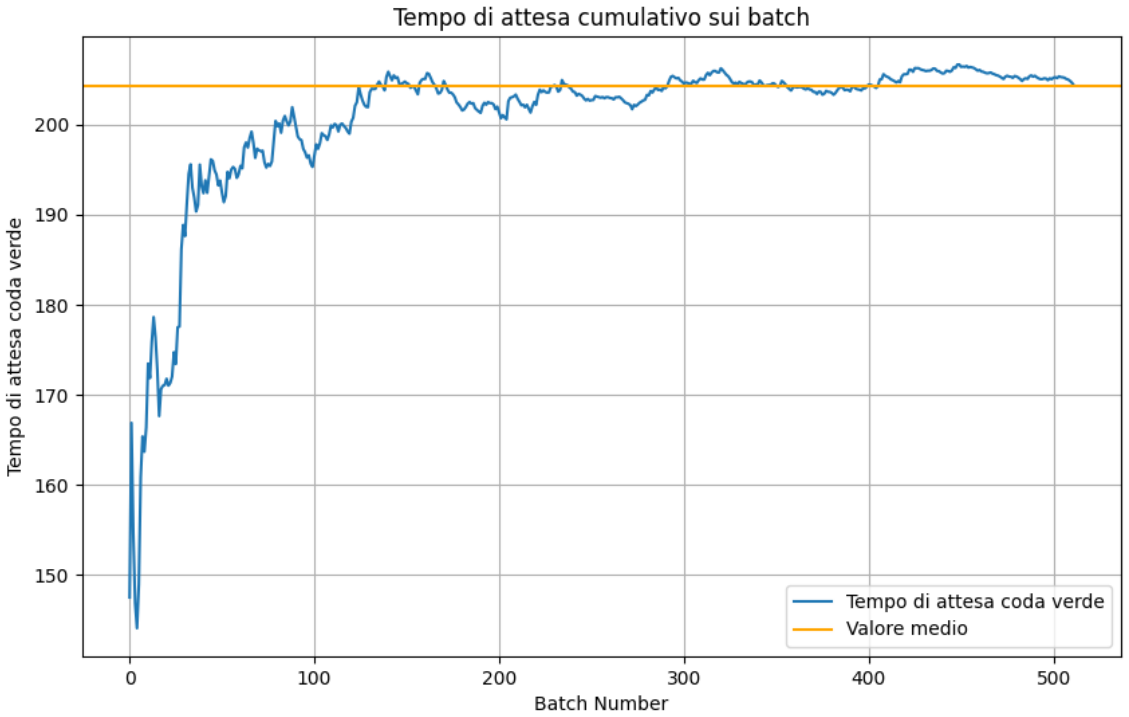
Simulazione ad orizzonte infinito

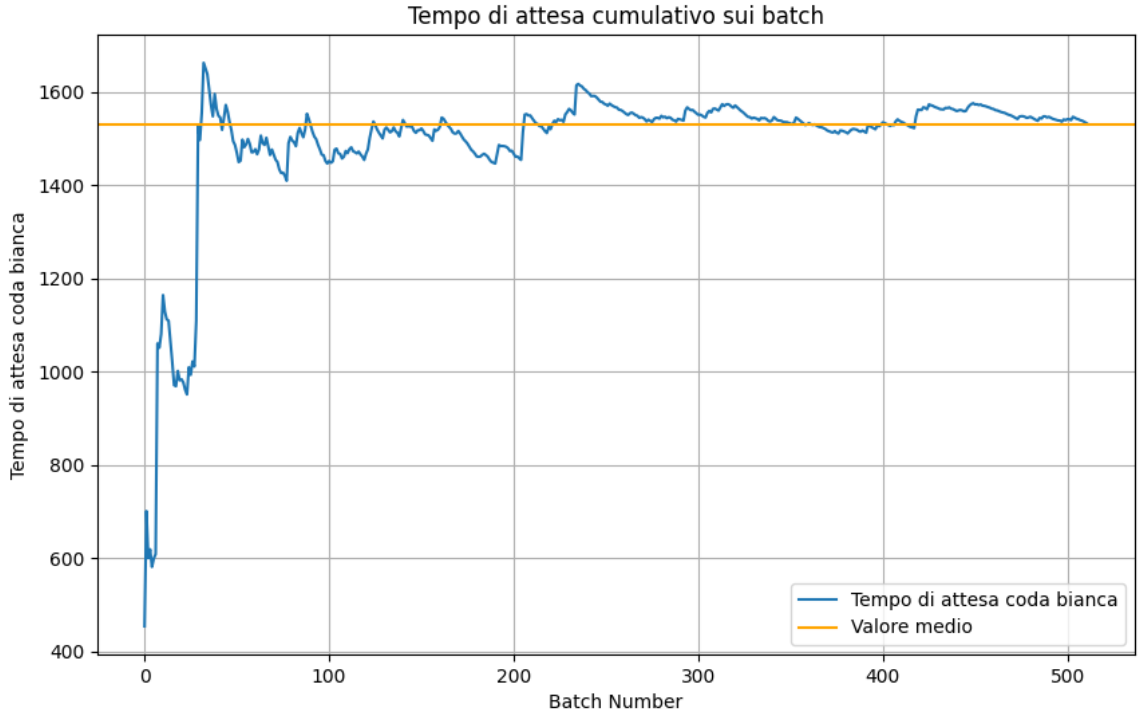




6. MODELLO MIGLIORATIVO

Simulazione ad orizzonte infinito





6. MODELLO MIGLIORATIVO

Risultati

Il modello migliorativo:

- Ottimizza la gestione delle risorse.
- Riduce le violazioni dei tempi di attesa.
- Miglioramenti differenziati per livello di priorità, ma con riduzione complessiva delle attese.

Table 6: Violazioni

Coda	**Numero violazioni**	**Percentuale**
Rossa	11	8.11×10^{-3} %
Arancione	112692	26.71 %
Azzurra	289113	35.03 %
Verde	372405	56.43 %
Bianco	67891	60.8 %

6. MODELLO MIGLIORATIVO

Considerazioni finali

Il modello migliorativo ci permette di realizzare un sistema di pronto soccorso più efficiente e reattivo per tutti i codici di gravità. Miglioramento:

- Arancione: 41.2967%
- Azzurro: 13.0769%
- Verde: 8%
- Bianco: 5.55%

7. CONSIDERAZIONI SUL PERSONALE

Scenario: Aggiunto un medico per turno

L'incremento del personale ha consentito di ridurre significativamente le violazioni, portandole vicine a 0.

Table 7: Violazioni

Coda	Numero violazioni	Percentuale
Rossa	2	1.47×10^{-3} %
Arancione	7848	1.86 %
Azzurra	2078	0.25 %
Verde	2246	0.34 %
Bianco	295	0.26 %

8. CONSIDERAZIONI SUGLI ACCESSI

Considerazioni sugli accessi

Scenario: Dimezzare il numero di accessi in codice bianco

Dimezzare gli accessi evitabili, identificati tramite il codice bianco, ci permette ridurre significativamente il numero di violazioni

Table 8: Violazioni

Coda	**Numero violazioni**	**Percentuale**
Rossa	10	7.376×10^{-3} %
Arancione	92046	21.82 %
Azzurra	197807	23.97 %
Verde	259664	39.35 %
Bianco	22385	40.13 %

9. RISULTATI

Conclusioni

- Con una gestione differente dello scheduling abbiamo un miglioramento che va dal 5% fino al 41%

- Fondamentale, se il numero di accessi resta inalterato o aumenta nei prossimi anni, è aumentare il personale

- Si evidenzia la necessità di diminuire gli ingressi mediante iniziative che facciano comprendere l'importanza di una struttura così importante che dovrebbe essere utilizzata solo in determinate circostanze

Table 9: Numero di violazioni per colore delle code per i vari modelli

Colore	**Standard**	**Migliorativo**	**Aumento Personale**	**Bianchi dimezzati**
Rosso	$1.47 \times 10^{-2}\%$	$8.11 \times 10^{-2}\%$	$1.47 \times 10^{-3}\%$	$7.376 \times 10^{-3}\%$
Arancione	45.50%	26.71%	1.86%	21.82%
Azzurro	40.3%	35.03%	0.25%	23.97%
Verde	61.34%	56.43%	0.34%	39.35%
Bianco	66.35%	60.8%	0.26%	40.13%

GRAZIE PER L'ATTENZIONE

Eugenio Di Gaetano – Marco Lorenzini

Università degli Studi di Roma "Tor Vergata"

Facoltà di Ingegneria

https://github.com/MarcoLor01/PMCSN_Project