



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):

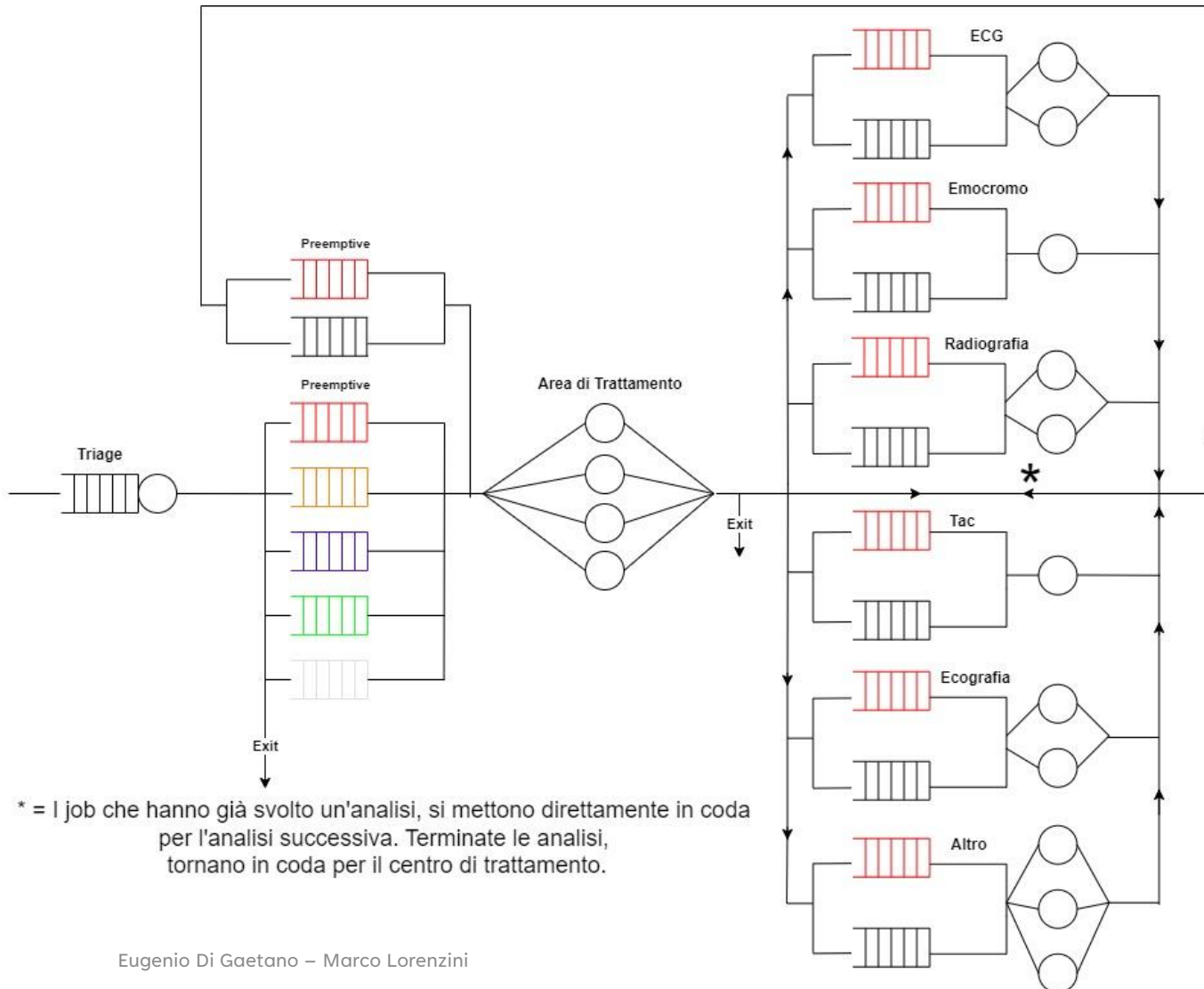
- Livello 1 (Rosso): Accesso immediato
- Livello 2 (Arancione): Accesso entro 15 minuti
- Livello 3 (Azzurro): Accesso entro 60 minuti
- Livello 4 (Verde): Accesso entro 120 minuti
- Livello 5 (Bianco): Accesso entro 240 minuti

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:

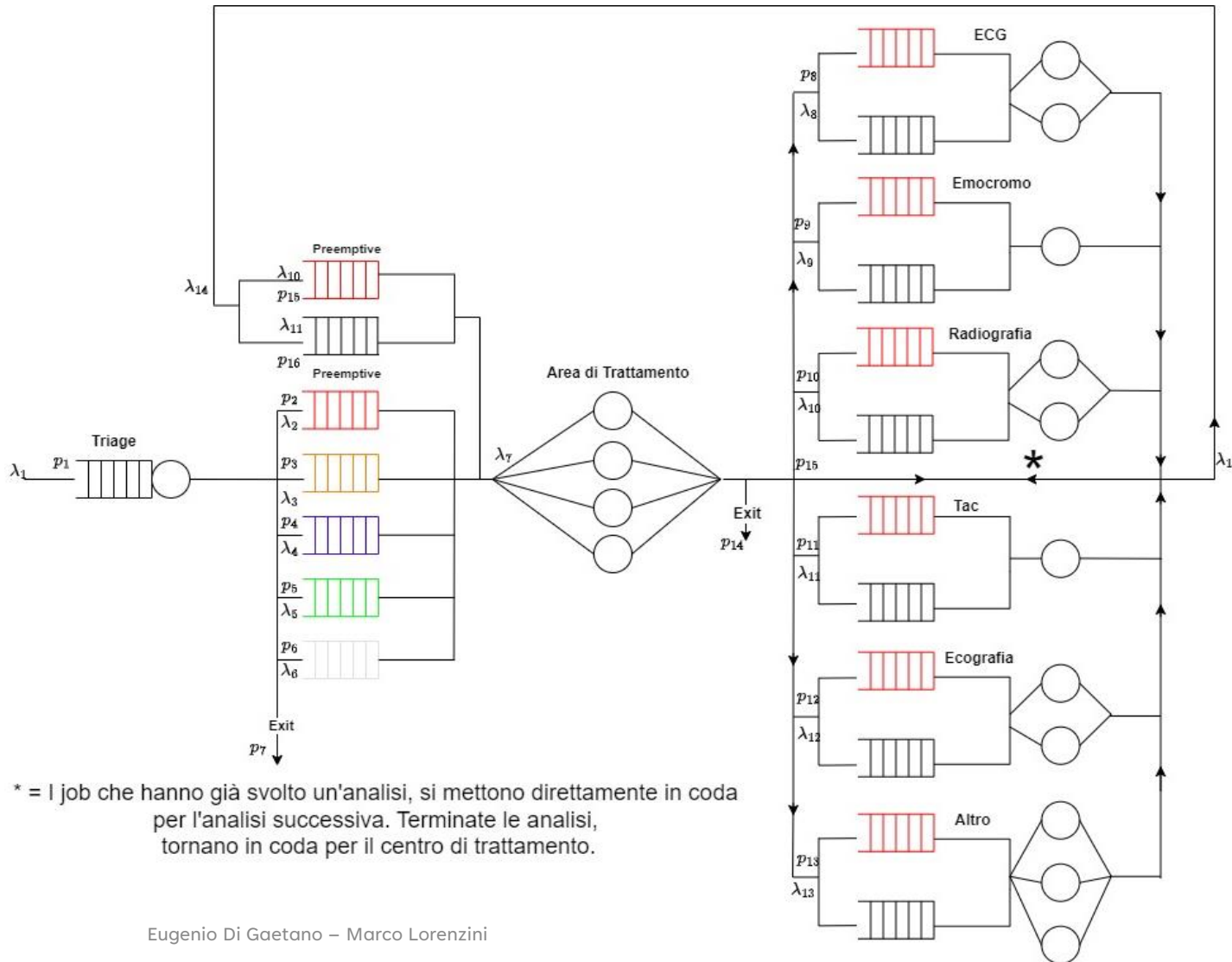
- Ingresso in un centro
- Uscita da un centro

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 Servizio:

- 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:	<i>0 minuti</i>	Codice rosso:	<i>0 ± 0 minuti</i>
Codice arancione:	<i>57 minuti</i>	Codice arancione:	<i>44.8 ± 24.21 minuti</i>
Codice azzurro:	<i>164 minuti</i>	Codice azzurro:	<i>83.65 ± 26.09 minuti</i>
Codice verde:	<i>154 minuti</i>	Codice verde:	<i>207.08 ± 33.53 minuti</i>
Codice bianco:	<i>148 minuti</i>	Codice bianco:	<i>476.53 ± 259.4 minuti</i>

5. SIMULAZIONE

Introduzione

Il QoS è basato sui tempi di attesa

Metrica utilizzata per le rappresentazioni grafiche:

- Sebbene il QoS sia espresso in termini di tempi di attesa, presentiamo i risultati grafici usando la popolazione media.
- Le due metriche sono direttamente correlate.

Vantaggio della Popolazione Media:

- Rappresentazione grafica più chiara e intuitiva rispetto ai tempi di attesa
- Possono essere tratte, con una interpretazione più immediata, le stesse conclusioni.

5. SIMULAZIONE

Simulazione ad orizzonte finito

- Tecnica della replicazione
- Durata simulazione: 1 settimana (10080 minuti)
- 64 Replicazioni
- Intervallo di confidenza: 95%

```
Il colore: ROSSO non ha sforamenti.
```

```
Ci sono state mediamente 60 violazioni in ogni ripetizione per il colore ARANCIONE . La media per singola violazione è di: 53.10805846560702 .  
Con una percentuale di job che violano il QoS di: 0.33330461008186124
```

```
Ci sono state mediamente 137 violazioni in ogni ripetizione per il colore AZZURRO . La media per singola violazione è di: 113.5476182401635 .  
Con una percentuale di job che violano il QoS di: 0.385114119244754
```

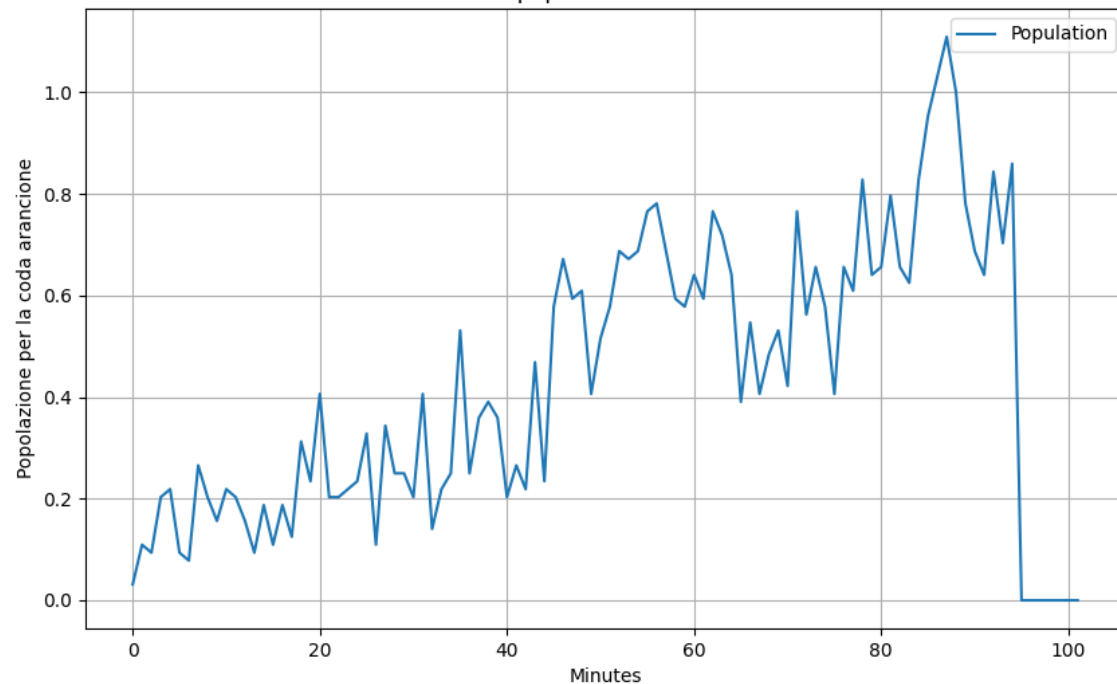
```
Ci sono state mediamente 162 violazioni in ogni ripetizione per il colore VERDE . La media per singola violazione è di: 213.01483384305232 .  
Con una percentuale di job che violano il QoS di: 0.576833720544308
```

```
Ci sono state mediamente 28 violazioni in ogni ripetizione per il colore BIANCO . La media per singola violazione è di: 287.7465859114174 .  
Con una percentuale di job che violano il QoS di: 0.5930764206401045
```

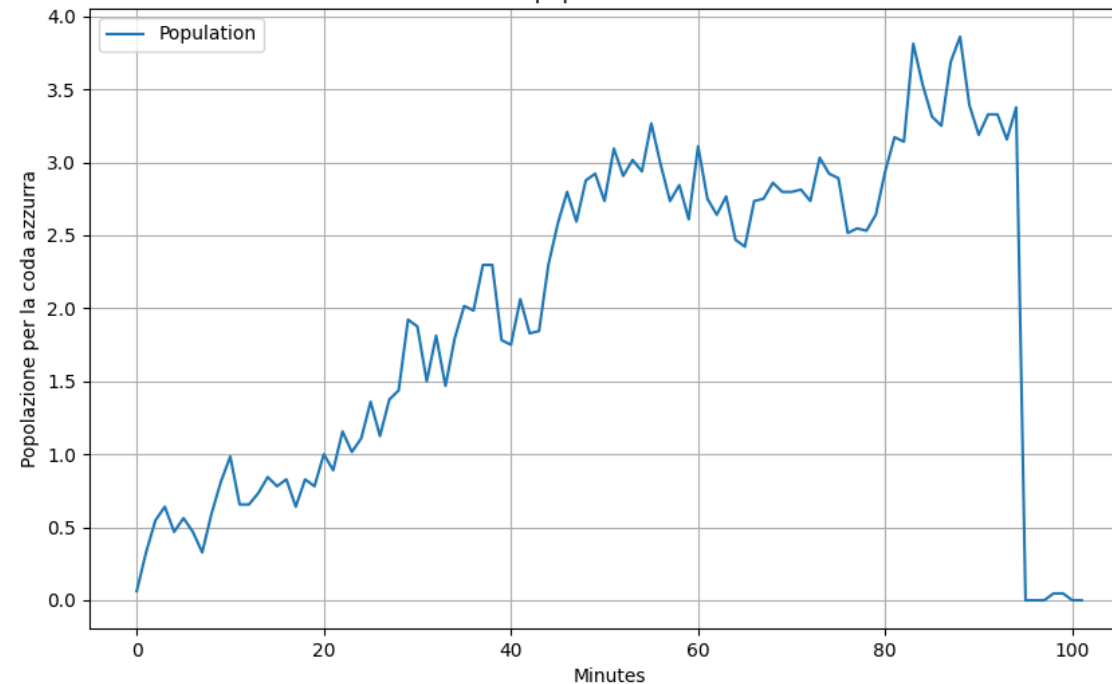
5. SIMULAZIONE

Simulazione ad orizzonte finito

Andamento della popolazione nella coda arancione



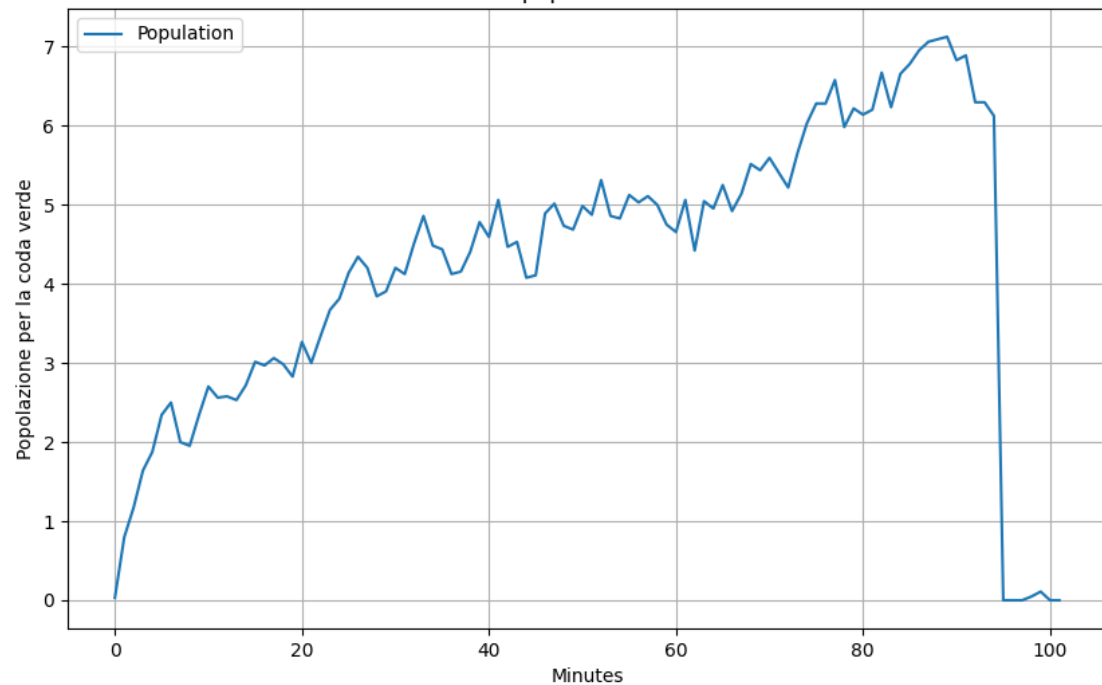
Andamento della popolazione nella coda azzurra



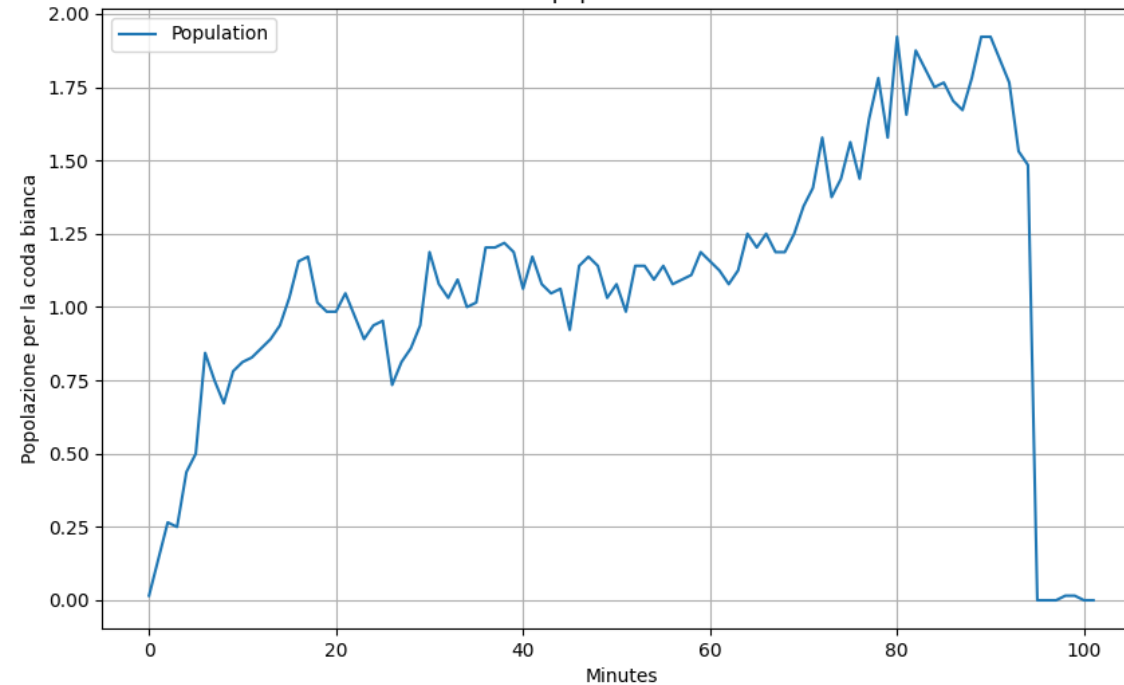
5. SIMULAZIONE

Simulazione ad orizzonte finito

Andamento della popolazione nella coda verde



Andamento della popolazione nella coda bianca



5. SIMULAZIONE

Simulazione ad orizzonte infinito

- Tecnica batch means
- Numero di batch(b): 64
- Elementi per batch(k): 512
- Intervallo di confidenza: 95%

```
Il colore: ROSSO non ha sforamenti.
```

```
Ci sono state 2784 violazioni per il colore ARANCIONE . La media per singola violazione è di: 85.18889201780038 .  
Con una percentuale di job che violano il QoS di: 0.3845835060091173
```

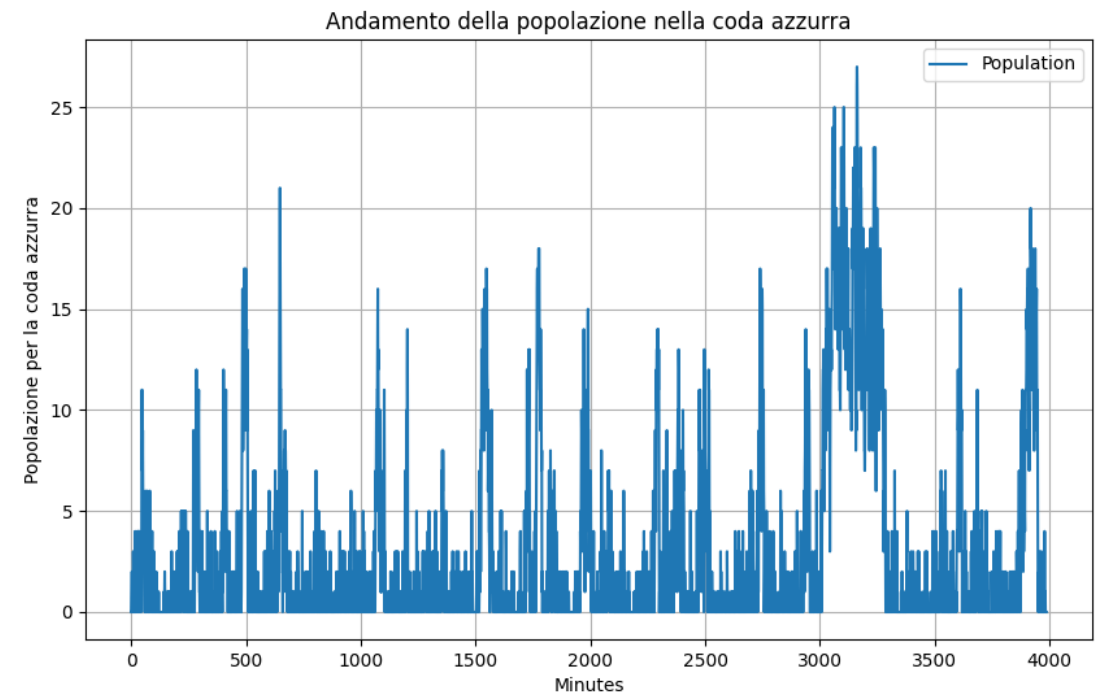
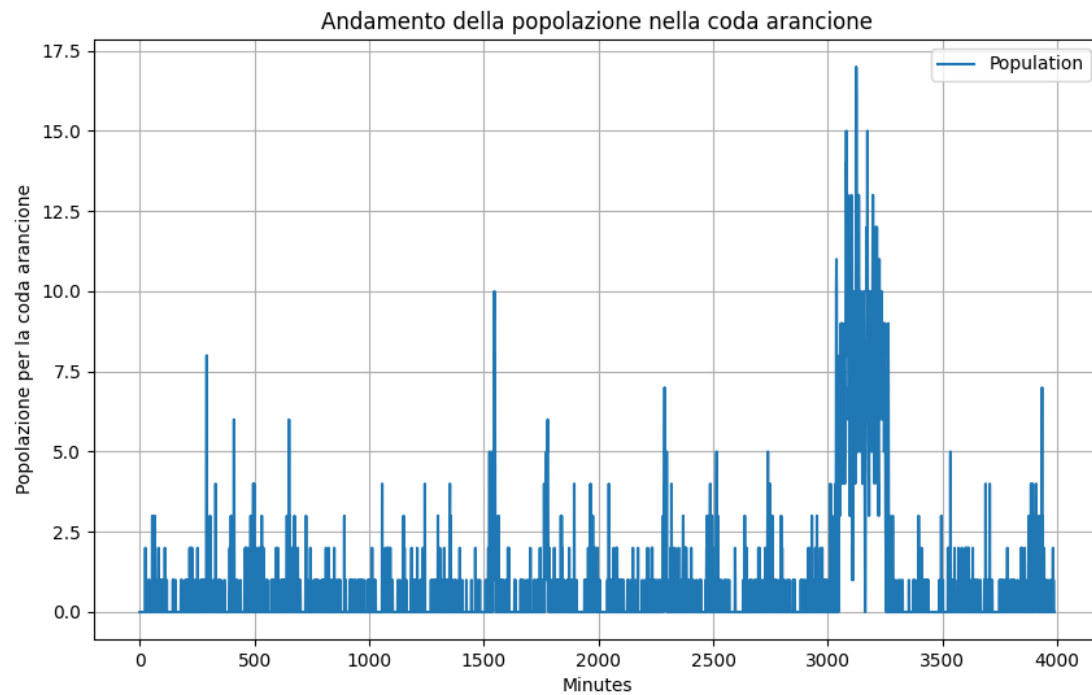
```
Ci sono state 6348 violazioni per il colore AZZURRO . La media per singola violazione è di: 143.4526718730426 .  
Con una percentuale di job che violano il QoS di: 0.4462565905096661
```

```
Ci sono state 7378 violazioni per il colore VERDE . La media per singola violazione è di: 241.3679941517809 .  
Con una percentuale di job che violano il QoS di: 0.6539620634639248
```

```
Ci sono state 1275 violazioni per il colore BIANCO . La media per singola violazione è di: 540.0016199913874 .  
Con una percentuale di job che violano il QoS di: 0.6616502335236119
```

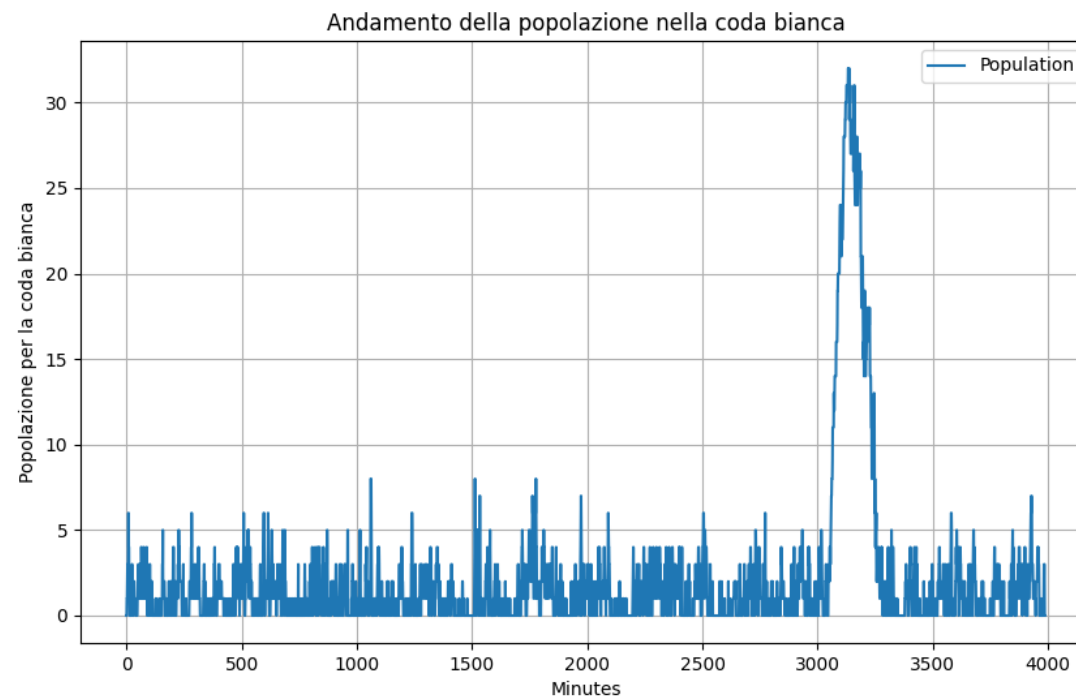
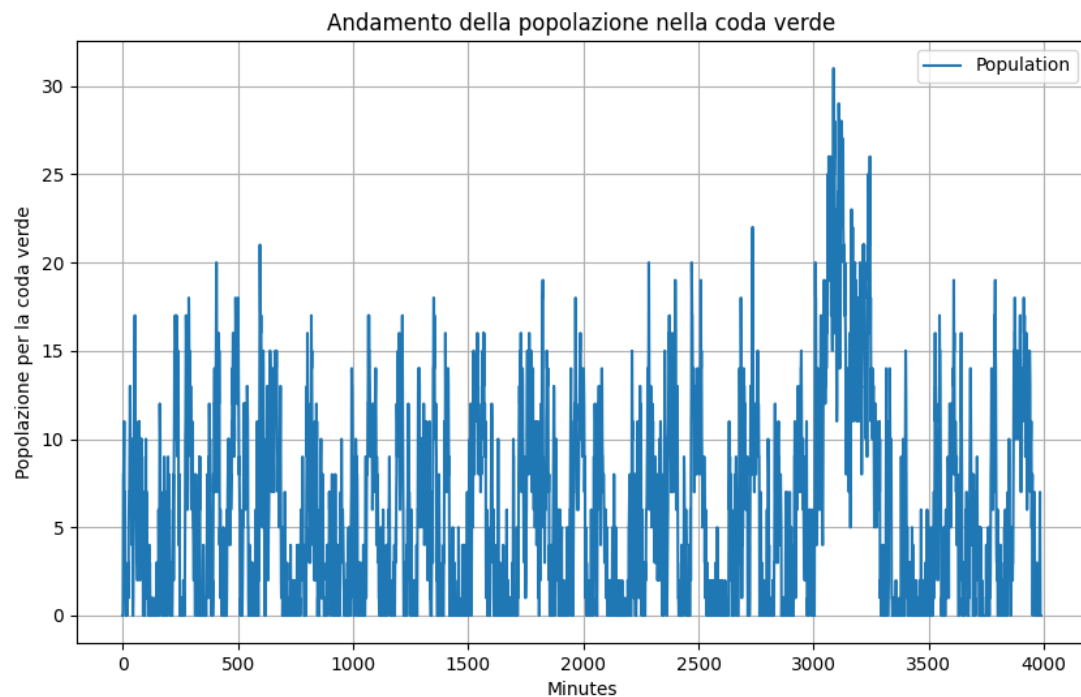

5. SIMULAZIONE

Simulazione ad orizzonte infinito



5. SIMULAZIONE

Simulazione ad orizzonte infinito



5. SIMULAZIONE

Considerazioni

Elevata percentuale di pazienti non trattati entro i tempi limite stabiliti

Orizzonte finito

Codice arancione: 33%

Codice azzurro: 38%

Codice verde: 59%

Codice bianco: 58%

Orizzonte infinito

Codice arancione: 38%

Codice azzurro: 45%

Codice verde: 65%

Codice bianco: 66%

6. MODELLO MIGLIORATIVO

Introduzione

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

```
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)
```

```
OBIETTIVO_MIGLIORATIVO = [0, 5, 20, 50, 110, 230]
```

6. MODELLO MIGLIORATIVO

Simulazione ad orizzonte finito

- Tecnica della replicazione
- Durata simulazione: 1 settimana (10080 minuti)
- 64 Replicazioni
- Intervallo di confidenza: 95%

```
Il colore: Rosso non ha sforamenti.
```

```
Ci sono state mediamente 46 violazioni in ogni ripetizione per il colore Arancione . La media per singola violazione è di: 14.591765200862701 .  
Con una percentuale di job che violano il QoS di: 0.25497630331753557
```

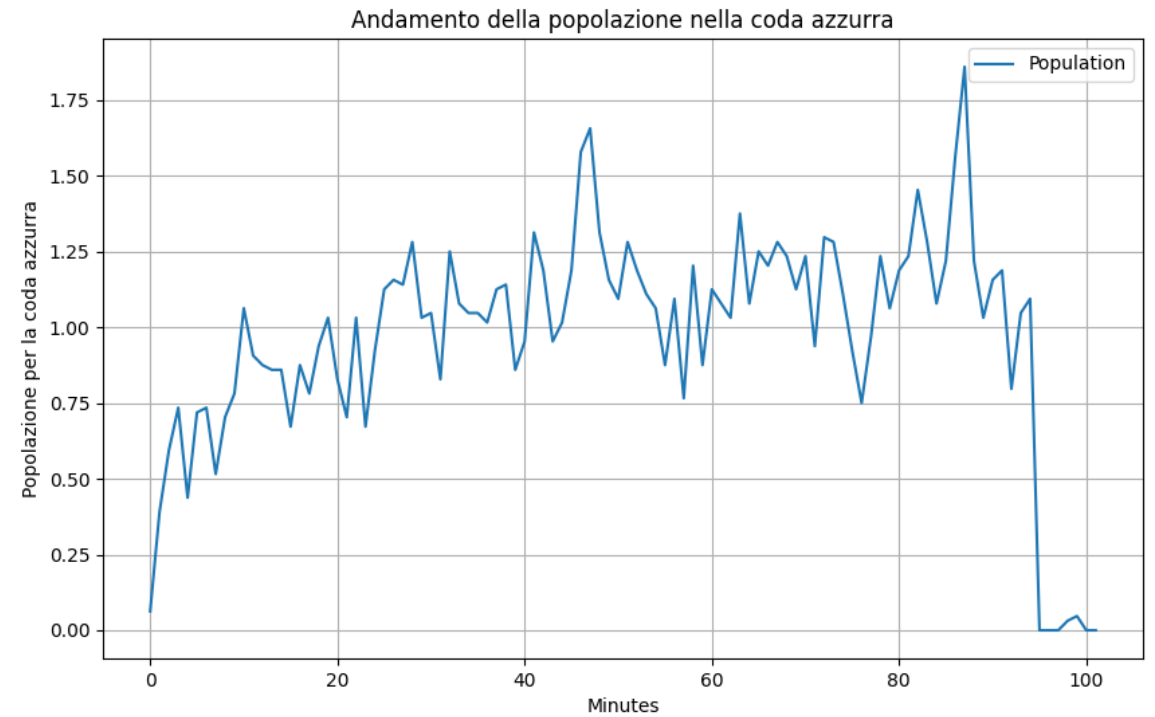
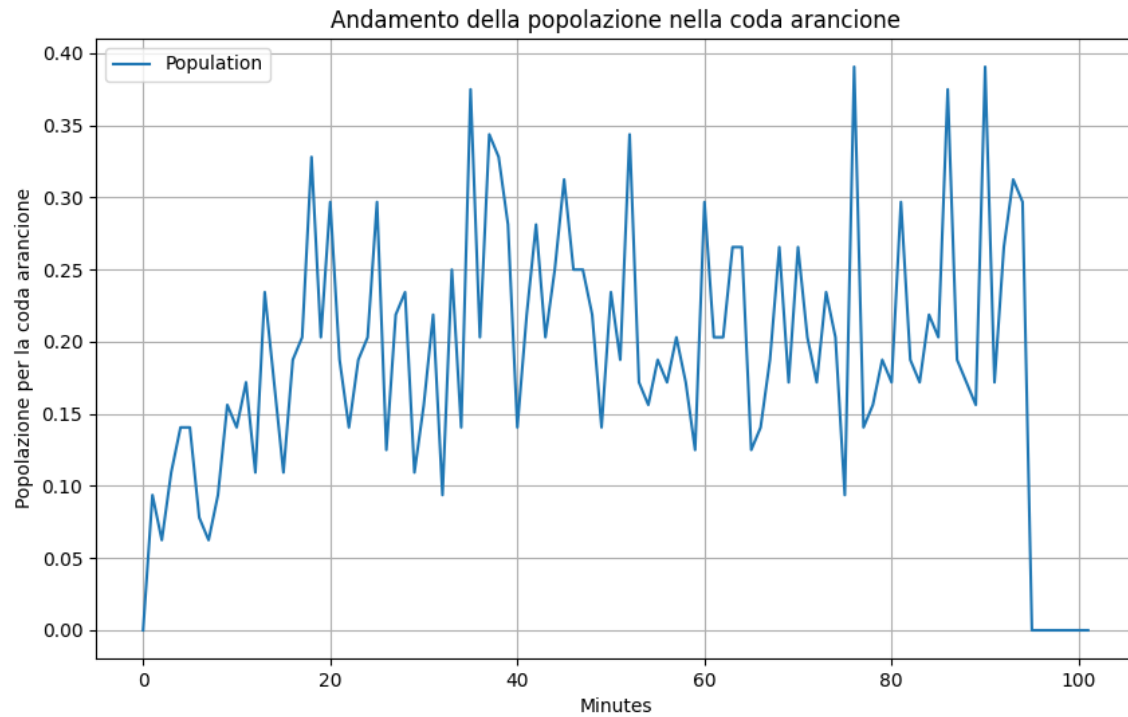
```
Ci sono state mediamente 129 violazioni in ogni ripetizione per il colore Azzurro . La media per singola violazione è di: 28.766757563254608 .  
Con una percentuale di job che violano il QoS di: 0.36430542778288866
```

```
Ci sono state mediamente 157 violazioni in ogni ripetizione per il colore Verde . La media per singola violazione è di: 170.87552269949936 .  
Con una percentuale di job che violano il QoS di: 0.5559243279123797
```

```
Ci sono state mediamente 27 violazioni in ogni ripetizione per il colore Bianco . La media per singola violazione è di: 1068.4466284611747 .  
Con una percentuale di job che violano il QoS di: 0.5685826257348139
```

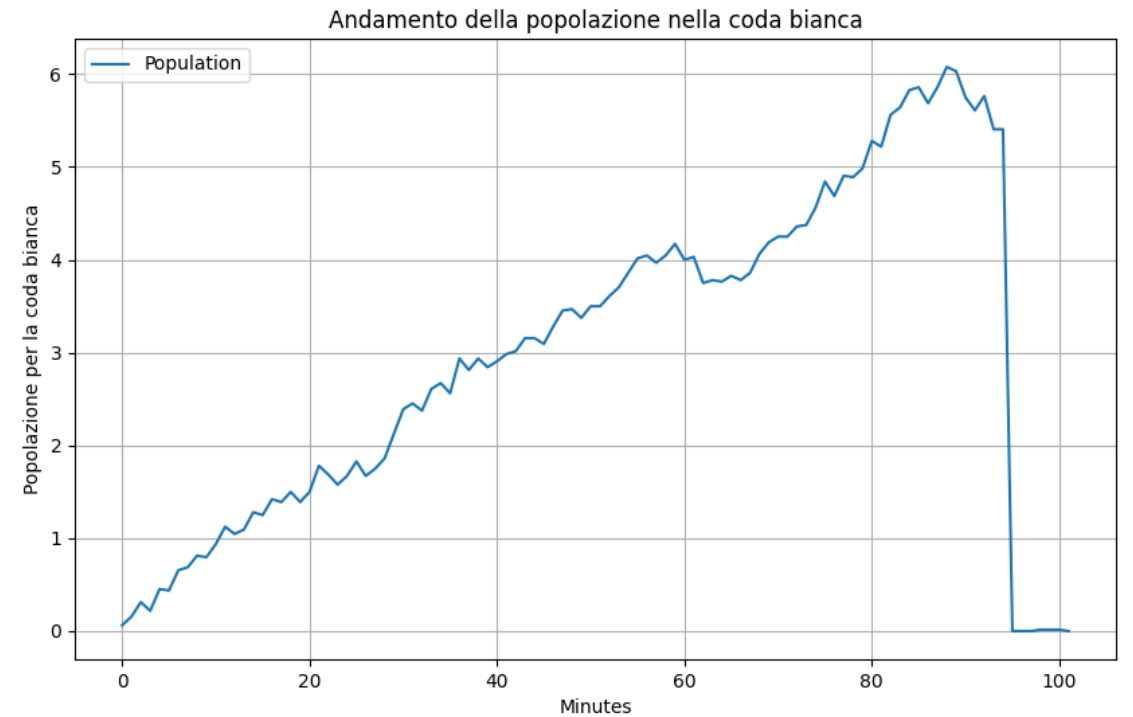
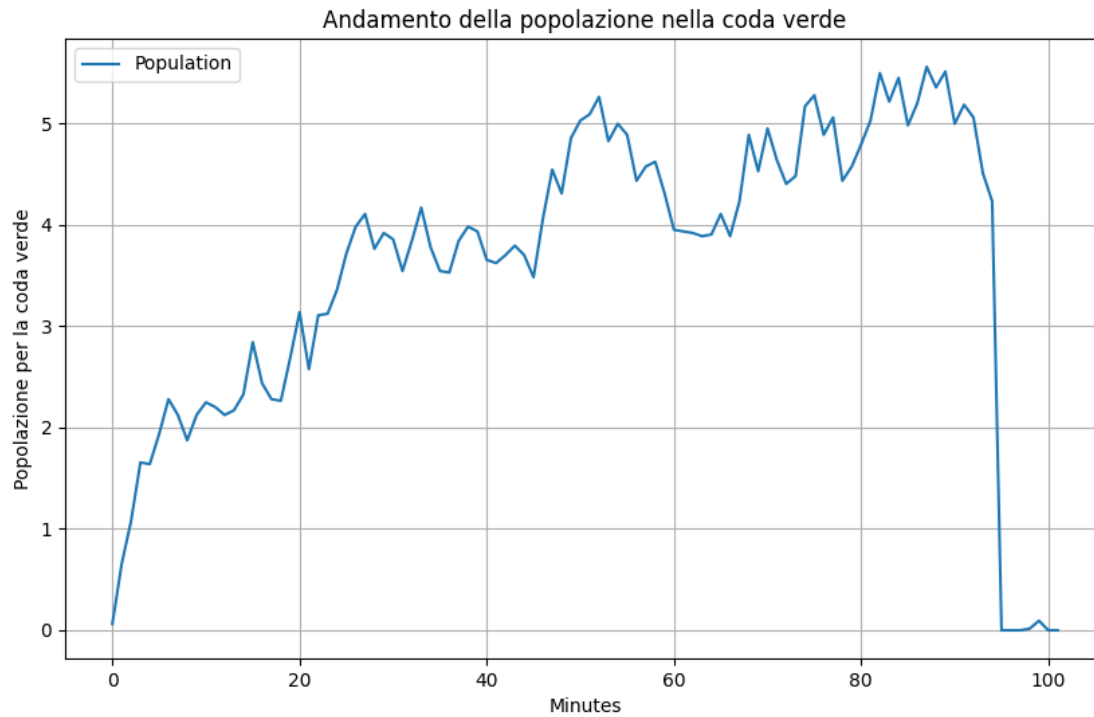
6. MODELLO MIGLIORATIVO

Simulazione ad orizzonte finito



6. MODELLO MIGLIORATIVO

Simulazione ad orizzonte finito



6. MODELLO MIGLIORATIVO

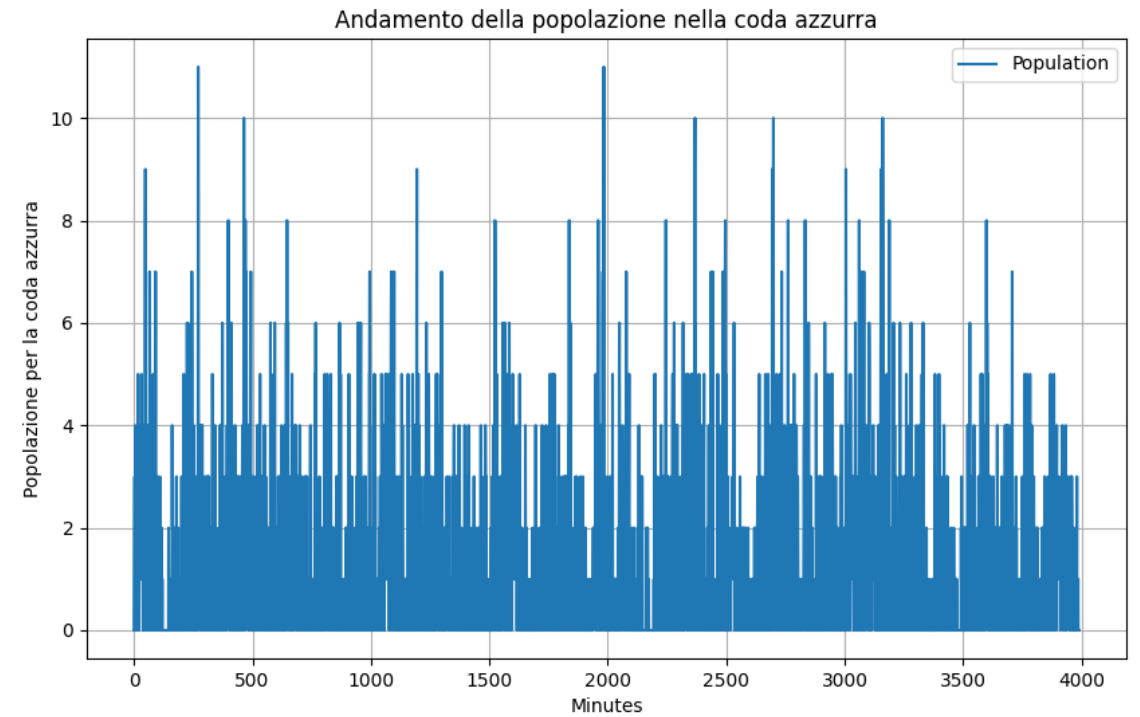
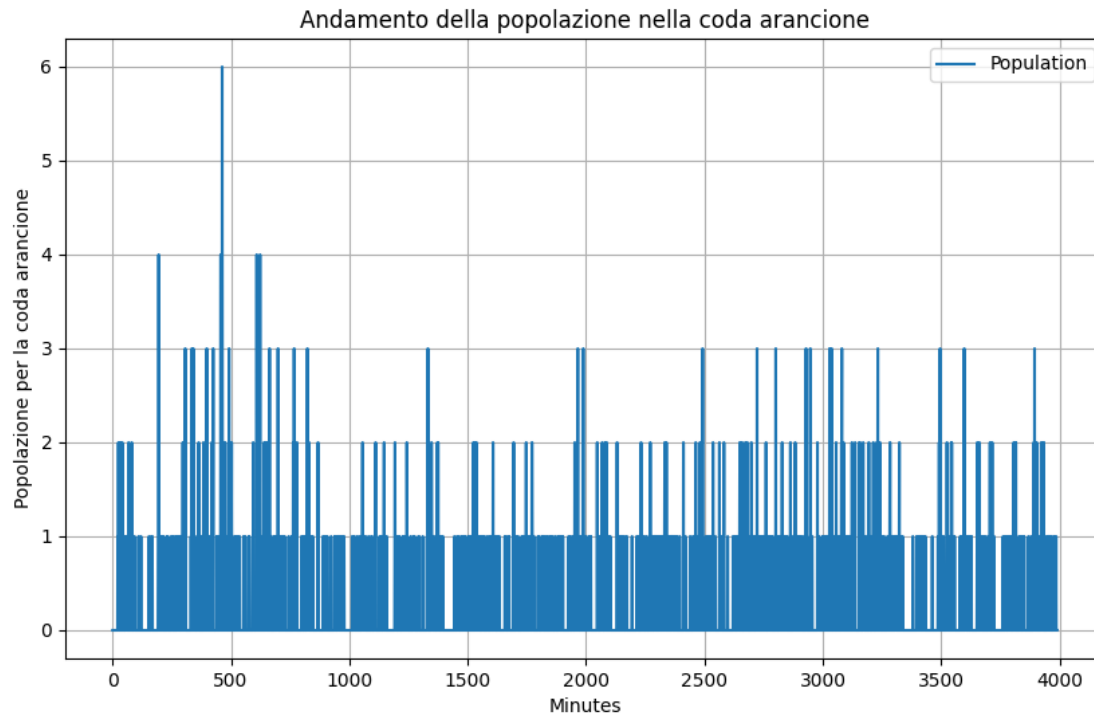
Simulazione ad orizzonte infinito

- Tecnica batch means
- Numero di batch(b): 64
- Elementi per batch(k): 512
- Intervallo di confidenza: 95%

```
Ci sono state 1 violazioni per il colore Rosso . La media per singola violazione è di: 3.215969157696236 .  
Con una percentuale di job che violano il QoS di: 0.0004424778761061947  
  
Ci sono state 1989 violazioni per il colore Arancione . La media per singola violazione è di: 15.05475728733163 .  
Con una percentuale di job che violano il QoS di: 0.2747617074181517  
  
Ci sono state 5955 violazioni per il colore Azzurro . La media per singola violazione è di: 31.68536830637251 .  
Con una percentuale di job che violano il QoS di: 0.4186291739894552  
  
Ci sono state 7078 violazioni per il colore Verde . La media per singola violazione è di: 170.1990235934102 .  
Con una percentuale di job che violano il QoS di: 0.6273710335046977  
  
Ci sono state 1224 violazioni per il colore Bianco . La media per singola violazione è di: 1778.0891712037887 .  
Con una percentuale di job che violano il QoS di: 0.6351842241826674
```

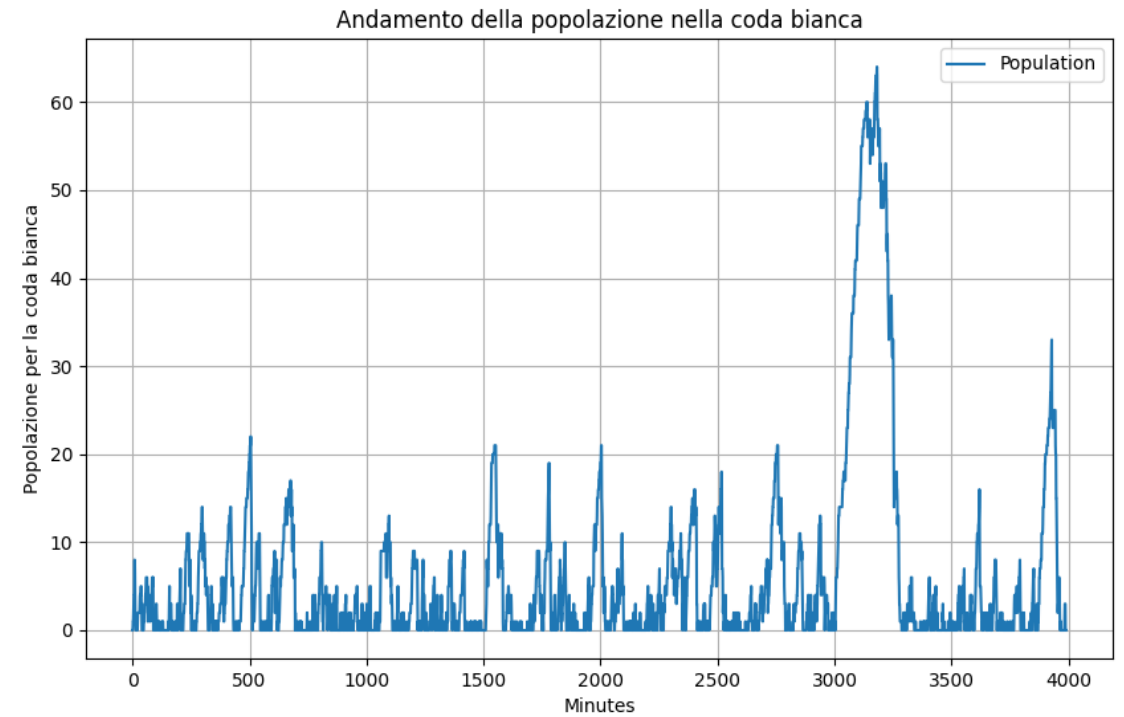
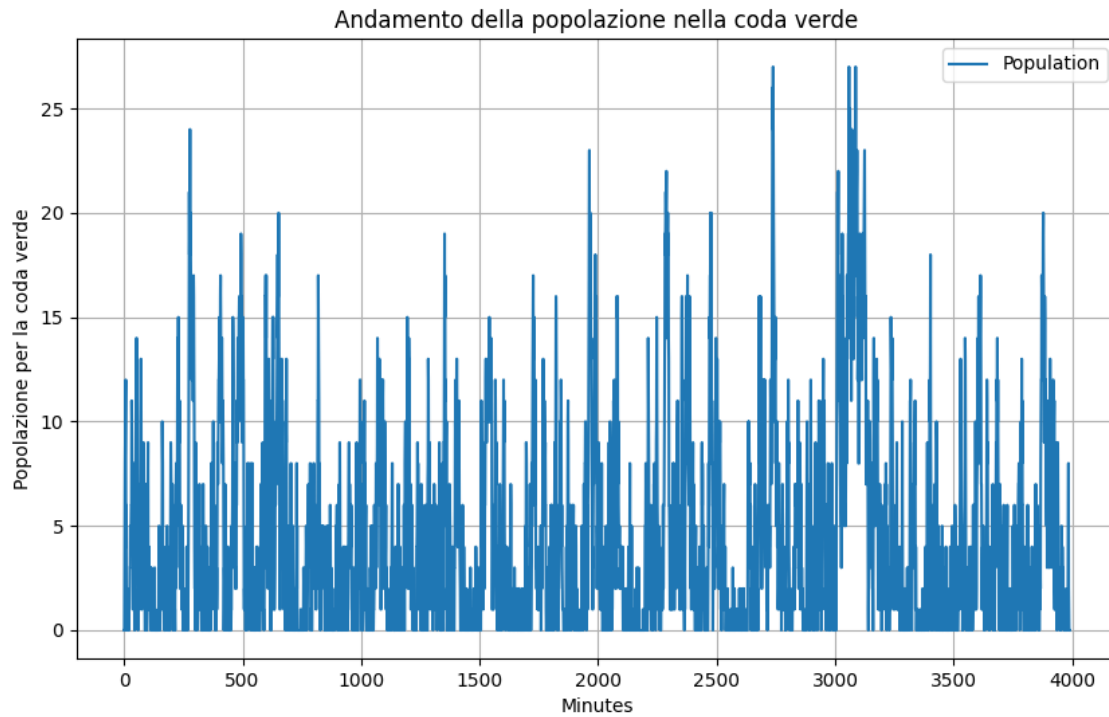

6. MODELLO MIGLIORATIVO

Simulazione ad orizzonte infinito



6. MODELLO MIGLIORATIVO

Simulazione ad orizzonte infinito



6. MODELLO MIGLIORATIVO

Considerazioni finali

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.

Risultato:

- Un sistema di pronto soccorso più efficiente e reattivo per tutti i codici di gravità.

Miglioramenti:

- Arancione: 23,55%
- Azzurro: 5,4%
- Verde: 3,62%
- Bianco: 4,13%

7. CONSIDERAZIONI SUL PERSONALE

Considerazioni sul personale

Scenario: Aggiunto un medico per turno

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

```
Il colore: Rosso non ha sforamenti.
```

```
Ci sono state mediamente 6 violazioni in ogni ripetizione per il colore Arancione . La media per singola violazione è di: 7.209691595918474 .  
Con una percentuale di job che violano il QoS di: 0.034295562257647565
```

```
Ci sono state mediamente 7 violazioni in ogni ripetizione per il colore Azzurro . La media per singola violazione è di: 10.808810883886043 .  
Con una percentuale di job che violano il QoS di: 0.019757304945897402
```

```
Ci sono state mediamente 6 violazioni in ogni ripetizione per il colore Verde . La media per singola violazione è di: 30.059434351057114 .  
Con una percentuale di job che violano il QoS di: 0.02218165726297157
```

```
Ci sono state mediamente 0 violazioni in ogni ripetizione per il colore Bianco . La media per singola violazione è di: 66.85545977033999 .  
Con una percentuale di job che violano il QoS di: 0.017635532331809273
```

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

```
Ci sono state mediamente 0 violazioni in ogni ripetizione per il colore Rosso . La media per singola violazione è di: 1.9464314626511623 .  
Con una percentuale di job che violano il QoS di: 0.0002723311546840959  
  
Ci sono state mediamente 41 violazioni in ogni ripetizione per il colore Arancione . La media per singola violazione è di: 14.633959860324097 .  
Con una percentuale di job che violano il QoS di: 0.22750775594622544  
  
Ci sono state mediamente 111 violazioni in ogni ripetizione per il colore Azzurro . La media per singola violazione è di: 28.073915586351728 .  
Con una percentuale di job che violano il QoS di: 0.31161547561670244  
  
Ci sono state mediamente 132 violazioni in ogni ripetizione per il colore Verde . La media per singola violazione è di: 158.56520134856572 .  
Con una percentuale di job che violano il QoS di: 0.46760719225449515  
  
Ci sono state mediamente 10 violazioni in ogni ripetizione per il colore Bianco . La media per singola violazione è di: 725.0169821534221 .  
Con una percentuale di job che violano il QoS di: 0.4494238156209987
```

9. RISULTATI

Conclusioni

- Con una gestione differente dello scheduling abbiamo un miglioramento che va dal 3% fino al 23%
- 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

Colore	Standard	Migliorativo	Aumento Personale	Bianchi dimezzati
Rosso	0%	0%	0%	0%
Arancione	33.33%	25.49%	3.43%	22.75%
Azzurro	38.51%	36.43%	1.98%	31.16%
Verde	57.68%	55.59%	2.22%	46.76%
Bianco	59.30%	56.85%	1.76%	44.94%



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