Corso di Performance Modeling Of Computer Systems And Networks

A.A. 2022/2023

AMA - Roma

Simone Festa, mat. 0320408
Michele Tosi, mat. 0327862

Agenda

Introduzione

Problematiche del sistema

Scopi ed obiettivi

Modello Concettuale

Modello computazionale

Modello delle specifiche

Verifica e validazione

Design degli esperimenti

Analisi del collo di bottiglia

Simulazione orizzonte finito

Simulazione orizzonte infinito

Modello Migliorativo

Introduzione

Novità

Verifica e validazione

Modelli di tipo:

Concettuale

Delle Specifiche

Computazione

Design degli esperimenti

Conclusioni

Introduzione

- **Sistema analizzato:** polo impiantistico AMA.

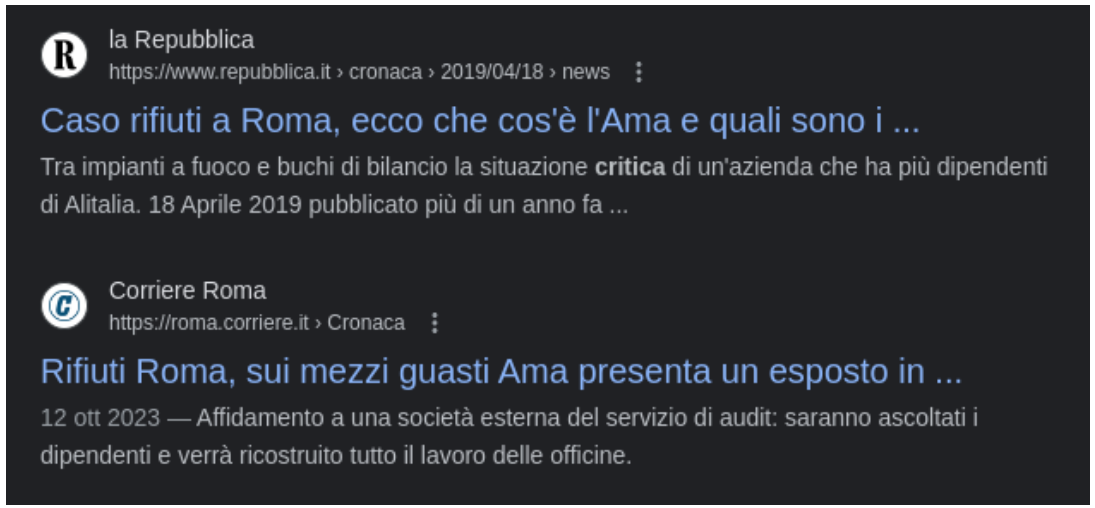
- I mezzi di raccolta di rifiuti, dopo aver completato il servizio, tornano nel polo impiantistico per lo smaltimento.

- Se i mezzi presentano un guasto, vengono riparati.

Problematiche del sistema

- Spesso l'azienda non rispetta i vincoli riguardo il numero minimo di mezzi che devono stare fuori dal sistema.
- Tempi di attesa per riparazioni molto lunghi.

Criticità: mancata gestione ottimale delle code dei mezzi (causa di disservizio pubblico).

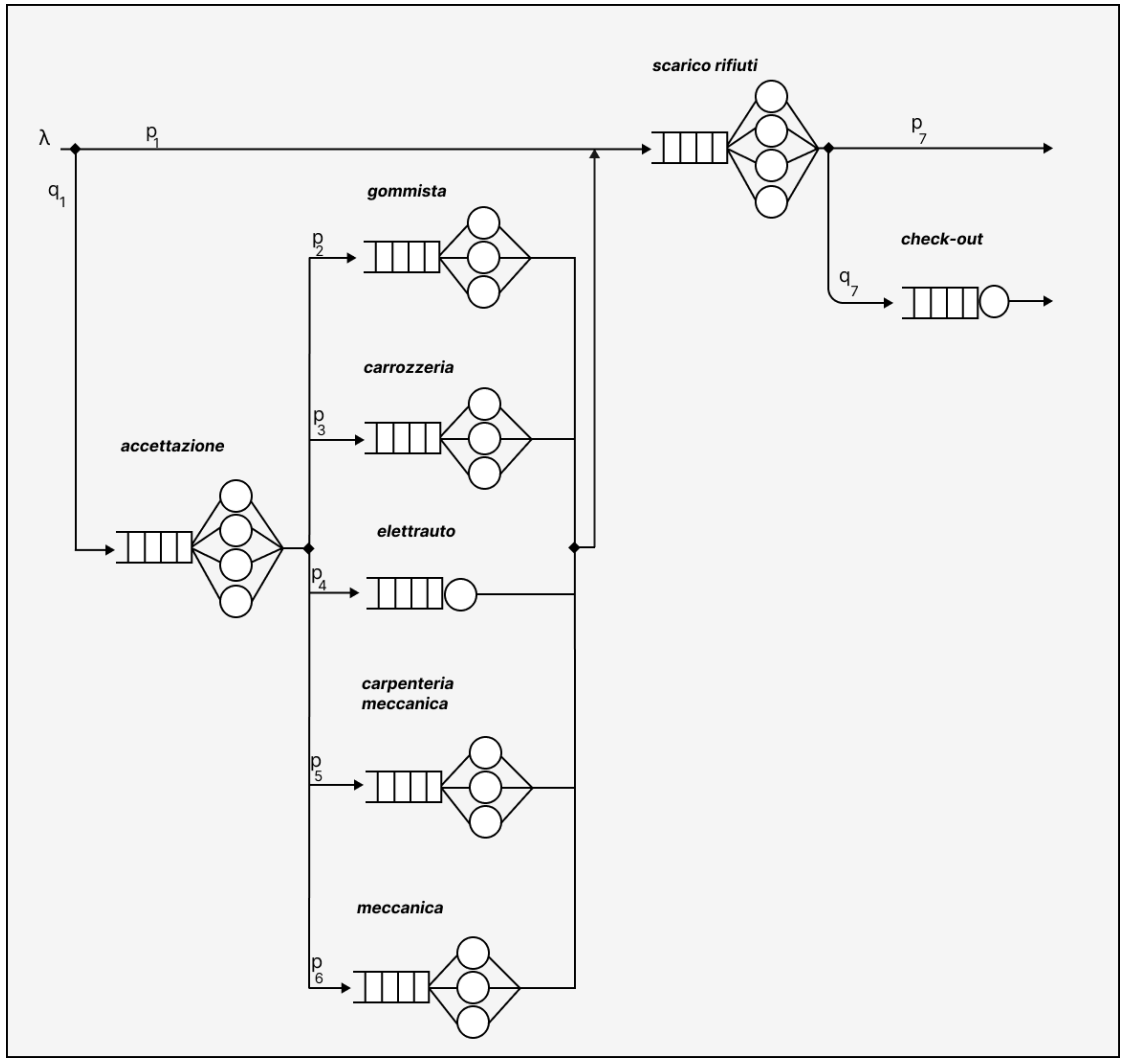


Scopi e obiettivi

- **Obiettivo:** minimizzare numero di mezzi all'interno del sistema in modo da rispettare i QoS:
  - **CSL2** fuori dal sistema devono essere più del 27%

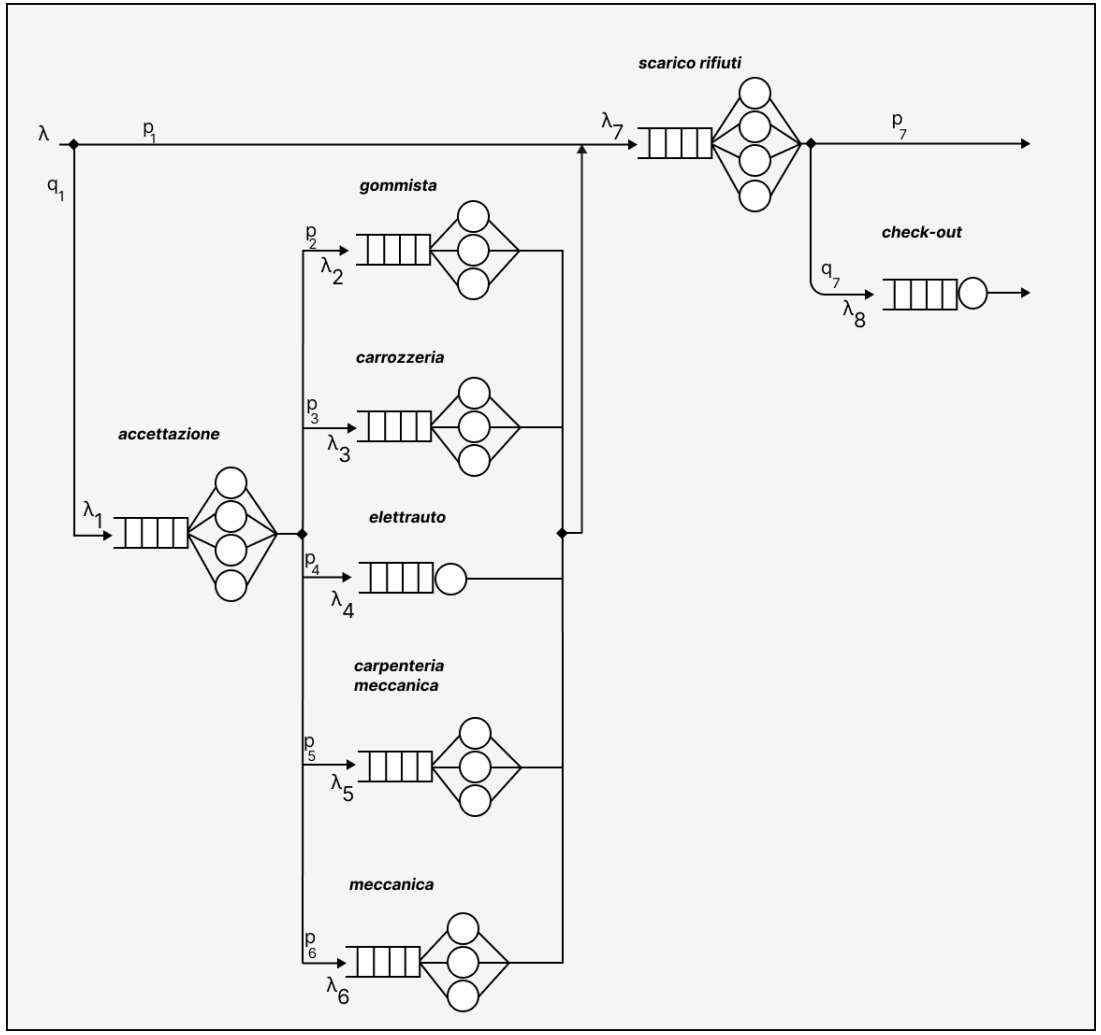
  - **CSL3** fuori dal sistema devono essere più del 64%.

Modello concettuale



- Utenti mezzi (182: 103 piccoli, 79 grandi).
- Eventi:
  - Arrivo di un mezzo in uno specifico centro.
  - Uscita di un mezzo da uno specifico centro.
- Gestione code: FIFO.

Modello delle specifiche



Equazioni di traffico:

- $\lambda_1 = (1 - p_1) \cdot \lambda$
- $\lambda_2 = p_2 \cdot \lambda_1$
- $\lambda_3 = p_3 \cdot \lambda_1$
- $\lambda_4 = p_4 \cdot \lambda_1$
- $\lambda_5 = p_5 \cdot \lambda_1$
- $\lambda_6 = p_6 \cdot \lambda_1$
- $\lambda_7 = \lambda$
- $\lambda_8 = (1 - p_7) \cdot \lambda_7$

Matrice di routing:

	Esterno	Accettazione	Gommista	Carrozzeria	Elettrauti	Carpenteria	Meccanica	Scarico	Checkout
Esterno	0	$1 - p_1$	0	0	0	0	0	p_1	0
Accettazione	0	0	p_2	p_3	p_4	p_5	p_6	0	0
Gommista	0	0	0	0	0	0	0	1	0
Carrozzeria	0	0	0	0	0	0	0	1	0
Elettrauti	0	0	0	0	0	0	0	1	0
Carpenteria	0	0	0	0	0	0	0	1	0
Meccanica	0	0	0	0	0	0	0	1	0
Scarico	p_7	0	0	0	0	0	0	0	$1 - p_7$
Checkout	1	0	0	0	0	0	0	0	0

Modello computazionale

- **Controllers:** contengono la logica dei vari nodi del sistema e un EventHandler per la gestione centralizzata degli eventi.
- **Models:** contiene codice per l'implementazione della MSQ e le costanti necessarie.
- **Utils:** contiene codice per la generazione di numeri multi-stream e produzione di statistiche di output.
- Per effettuare la simulazione del sistema è stato utilizzato l'approccio della Next-Event Simulation.

0	t	x	arrival
1	t	x	completion of service by server 1
2	t	x	completion of service by server 2
3	t	x	completion of service by server 3
4	t	x	completion of service by server 4

Event-List

```
private double t;
private int x;
private int vehicleType;
```

Implementazione entry della EventList (EventListEntryClass)

Modello computazionale (2)

- **Gestione degli arrivi esterni:** l'accettazione si preoccupa di indirizzare gli eventi esterni nelle varie officine.

```
eventHandler.getInternalEventsOfficina(off).add(new EventListEntry(event.getT(), event.getX(), event.getVehicleType()));
```

- Se i mezzi sono tutti all'interno del sistema il nuovo evento viene restituito con un tipo invalido di veicolo finché non si ha un'uscita.

```
eventHandler.decrementVType(event.getVehicleType());
```

```
int vType=rnd.getExternalVehicleType();  //vedo quale tipo di veicolo sta arrivando
if(vType==Integer.MAX_VALUE)  { // se il veicolo è pari a max_value vuol dire che non possono esserci arrivi
   eventList.get(0).setX(0);
   eventHandler.setEventsAccettazione(eventList);
   return;
}
```

- I servizi sono stati modellati come **normali troncate**.

```
public double idfTruncatedNormal(double m, double s, double lowerBound, double upperBound, double r){
   double a= cdfNormal(m, s, lowerBound-1);
   double b= 1.0-cdfNormal(m, s, upperBound);
   double u=idfUniform(a,1.0-b,r);
   return idfNormal(m, s, u);
}
```

Verifica

- In questa fase si è fatto uso di tempi di servizio esponenziali per permettere il calcolo.
- Seed: 123456789.
- Intervallo di confidenza: 95%.
- Criteri di consistenza considerati:
  - $E[T_s] = E[T_q] + E[S_i]$
  - $E[N_s] = E[N_q] + m \cdot \rho$
  - $0 < \rho \leq 1$

CENTRI	Popolazione in coda	Tempo di attesa	Popolazione nel sistema	Tempo di risposta	Utilizzazione
Accettazione	0,007±8,107*10-4	3,928±0,487	1,008±0,023	605,64±13,18	0,250±0,006
Gommista	0,525±0,034	1.048,401±65,237	2,328±0,066	4.661,33±123,75	0,601±0,014
Carrozzeria	0,030±0,002	183,600±13,003	0,923±0,021	5.598,75±123,27	0,298±0,007
Elettrauto	6,882±0,710	41.520,13±4097,79	7,766±0,720	46.890,29±4138,81	0,884±0,020
Carpenteria	0,511±0,031	1.547,12±91,04	2,285±0,062	6.936,14±178,04	0,591±0,013
Meccanica	8,023±1,485	15.676,39±2.457,30	10,439±1,513	21.076,98±2.494,29	0,899±0,020
Scarico	3,297±0,343	589,588±58,773	6,634±0,385	1.190,85±64,50	0,834±0,019
Check-out	1,296±0,061	2.344,62±102,33	1,962±0,072	3.551,079±119,403	0,665±0,015

Risultati ottenuti dalla simulazione

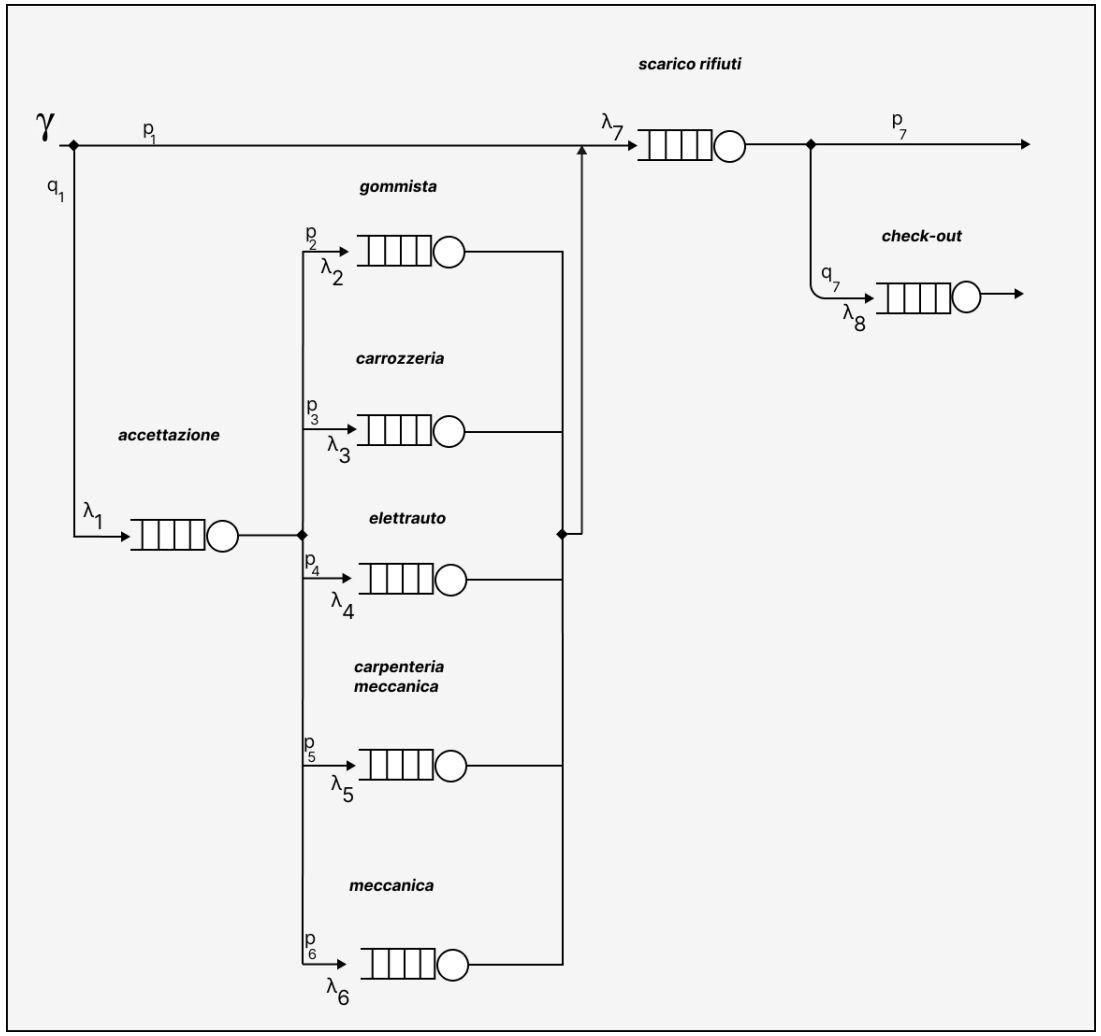
CENTRI	Popolazione in coda	Tempo di attesa	Popolazione nel sistema	Tempo di risposta	Utilizzazione
Accettazione	0,0065	3,9342	0,9965	603,9342	0,2475
Gommista	0,5078	1.025,8806	2,2898	4.625,8806	0,5940
Carrozzeria	0,0289	174,8555	0,9199	5.574,8555	0,2970
Elettrauto	6,6386	40.233,8164	7,5296	45.633,8164	0,8910
Carpenteria	0,5078	1.538,8209	2,2898	6.938,8209	0,3471
Meccanica	6,5493	13.230,9973	9,2223	18.630,9973	0,8910
Scarico	3,0273	550,4225	6,3273	1.150,4225	0,8250
Check-out	1,2812	2.329,4100	1,9410	3.529,4100	0,6600

Risultati ottenuti analiticamente

Validazione

- Arrivi al sistema con normali troncate nell'arco delle 24 compresi tra 466 e 521, valore coerente con i 480 arrivi del sistema reale.
- Nello scarico confluiscono tutti i mezzi presenti nel sistema, una volta chiuse le porte continuano ad arrivare i mezzi dalle officine.
- Nell'accettazione ci si aspetta una popolazione ridotta e costante grazie ai tempi di servizio ridotti e ai numerosi serventi.
- Nelle officine più utilizzate (meccanica ed elettrauto) ci aspettiamo la maggior parte della popolazione del sistema (lunghi tempi di servizio).

Analisi del collo di bottiglia



- Servizio medio dei centri del sistema:

$S_1 = \frac{1}{\mu_1} = 600\,\text{sec}$ $S_5 = \frac{1}{\mu_5} = 5400\,\text{sec}$

$S_2 = \frac{1}{\mu_2} = 3600\,\text{sec}$ $S_6 = \frac{1}{\mu_6} = 5400\ sec$

$S_3 = \frac{1}{\mu_3} = 5400\,\text{sec}$ $S_7 = \frac{1}{\mu_7} = 600\ sec$

$S_4 = \frac{1}{\mu_4} = 5400\,\text{sec}$ $S_8 = \frac{1}{\mu_8} = 1200\ sec$

- Numero medio di visite ai centri del sistema:

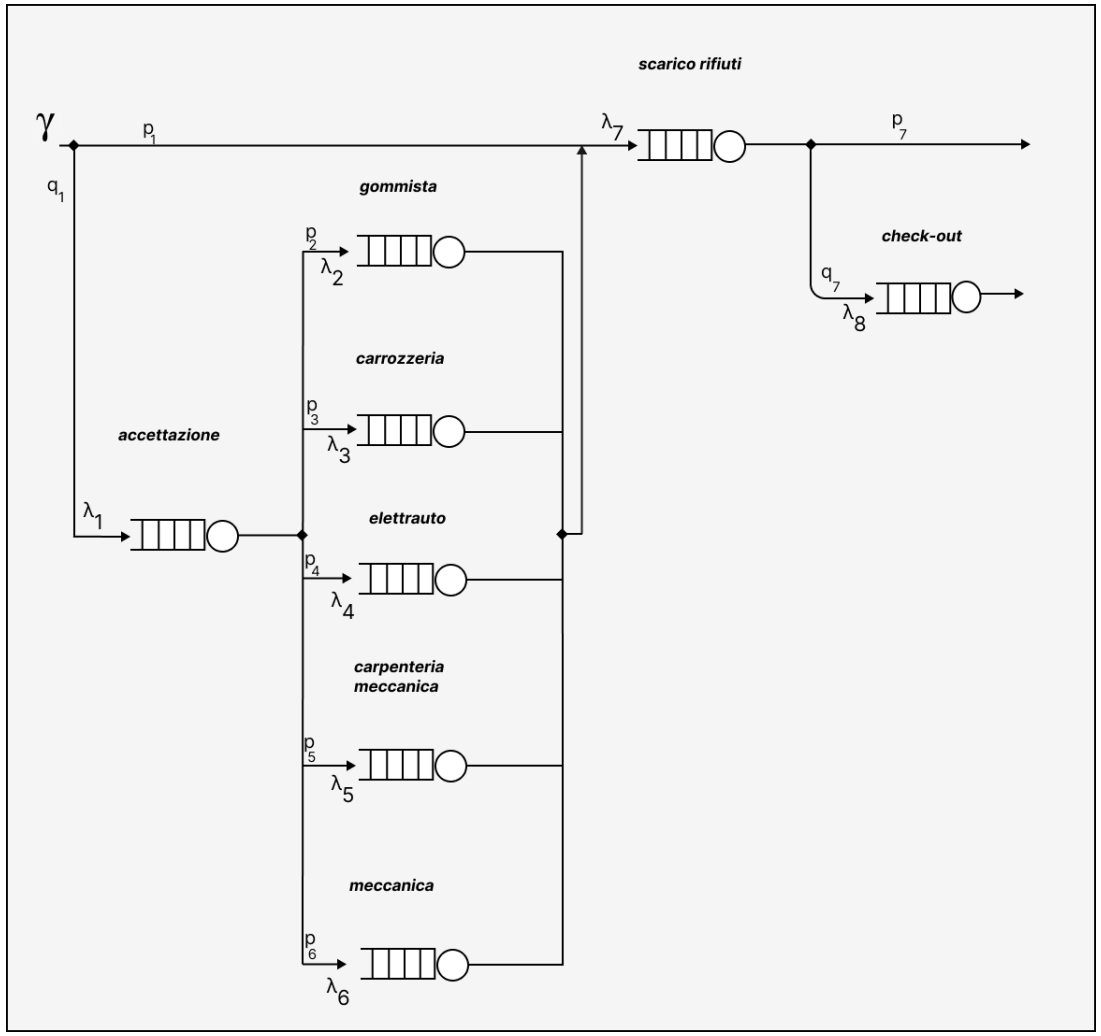
$v_1 = \frac{\lambda_1}{\gamma} = 0.3$ $v_5 = \frac{\lambda_5}{\gamma} = 0.06$

$v_2 = \frac{\lambda_2}{\gamma} = 0.09$ $v_6 = \frac{\lambda_6}{\gamma} = 0.09$

$v_3 = \frac{\lambda_3}{\gamma} = 0.03$ $v_7 = \frac{\lambda_7}{\gamma} = 1$

$v_4 = \frac{\lambda_4}{\gamma} = 0.03$ $v_8 = \frac{\lambda_8}{\gamma} = 0.1$

Analisi del collo di bottiglia



- Domanda media per i centri del sistema:

$D_1 = v_1 \cdot S_1 = 180\ sec$ $\quad D_5 = v_5 \cdot S_5 = 324\ sec$

$D_2 = v_2 \cdot S_2 = 324\ sec$ $\quad D_6 = v_6 \cdot S_6 = 486\ sec$

$D_3 = v_3 \cdot S_3 = 162\ sec$ $\quad D_7 = v_7 \cdot S_7 = 600\ sec$

$D_4 = v_4 \cdot S_4 = 162\ sec$ $\quad D_8 = v_8 \cdot S_8 = 120\ sec$

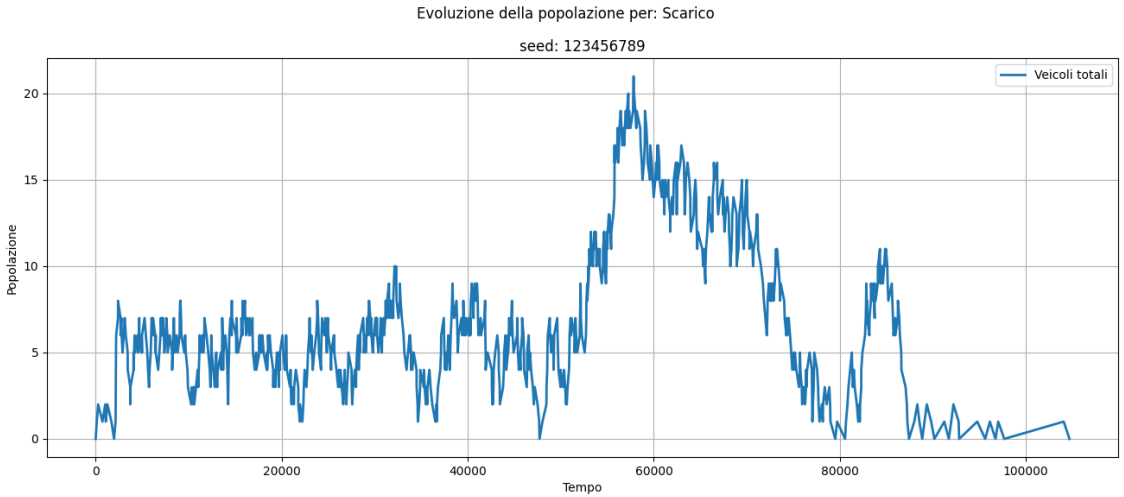
- I valori calcolati sono coerenti con quelli ottenuti dalla simulazione:

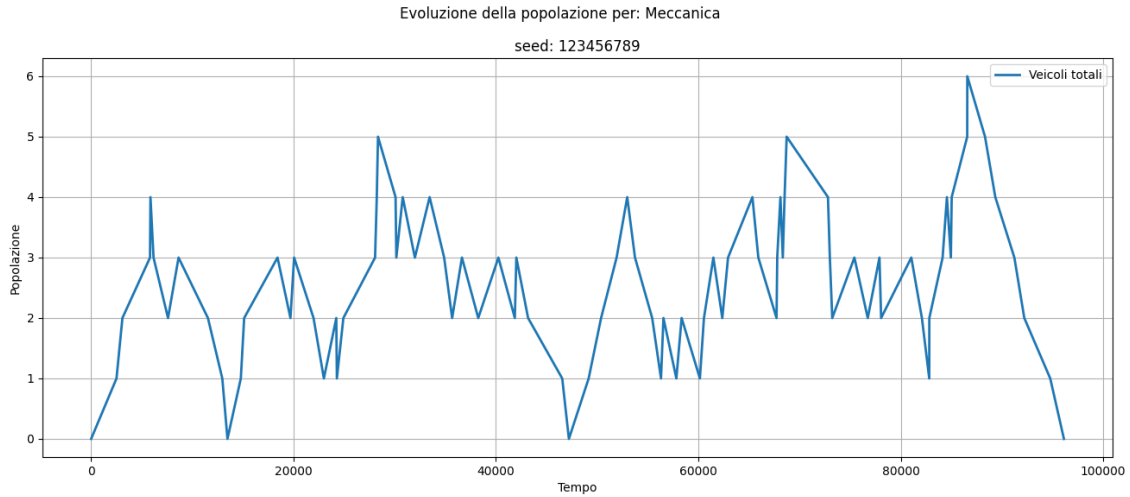
	Accettazione	Gommista	Carrozzeria	Elettrauti	Carpenteria	Meccanica	Scarico	Checkout
V	0.300158	0.090122	0.029928	0.029888	0.059984	0.090235	1	0.099851
D (sec)	180.094748	324.439330	161.610493	161.395675	323.915256	487.270761	600	119.821711

- Da questi risultati si nota che è lo scarico il collo di bottiglia.

Simulazione ad orizzonte finito

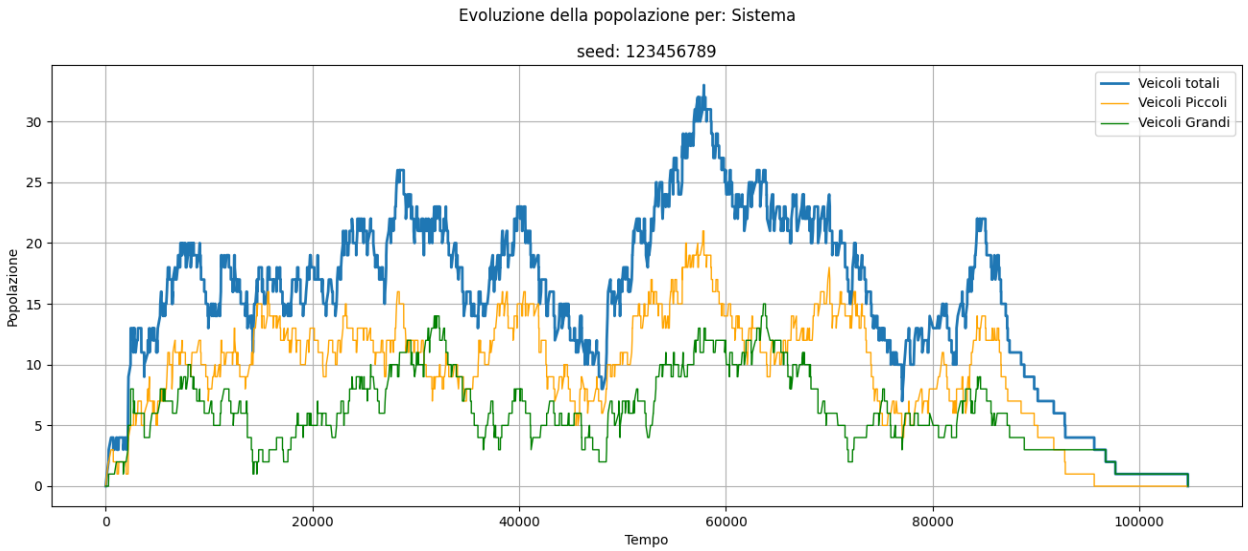
- Uso della tecnica di **Replicazione**, 96 repliche indipendenti.
- **Intervallo di confidenza** calcolato come $t^* \cdot \left(\frac{\mathrm{s}}{\sqrt{n-1}} \right)$





Replicazioni indipendenti, run n° 1 dei centri "Scarico" e "Meccanica"

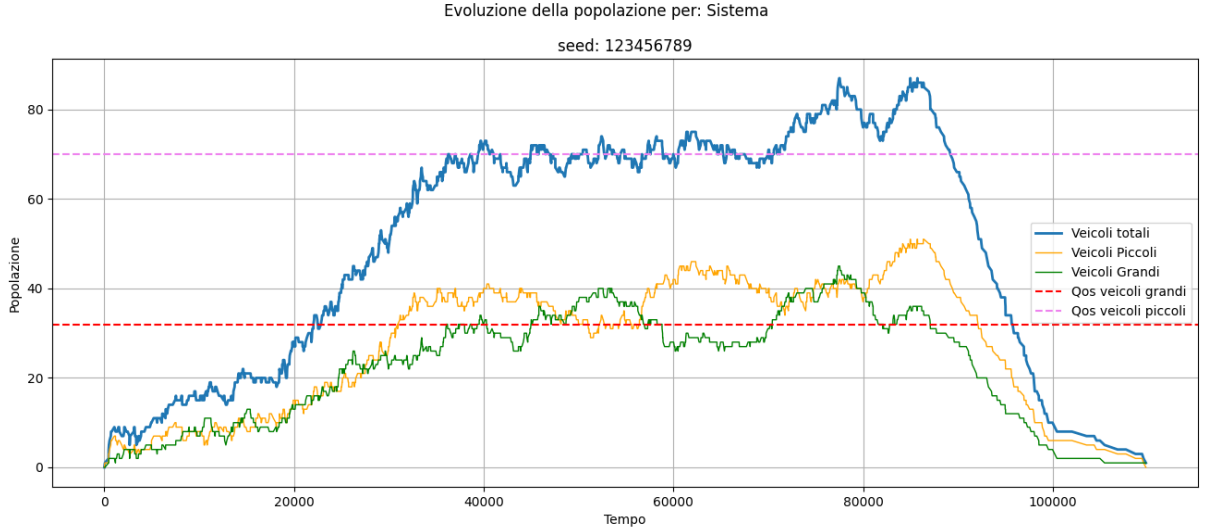
Simulazione ad orizzonte finito



Run n° 1 della simulazione

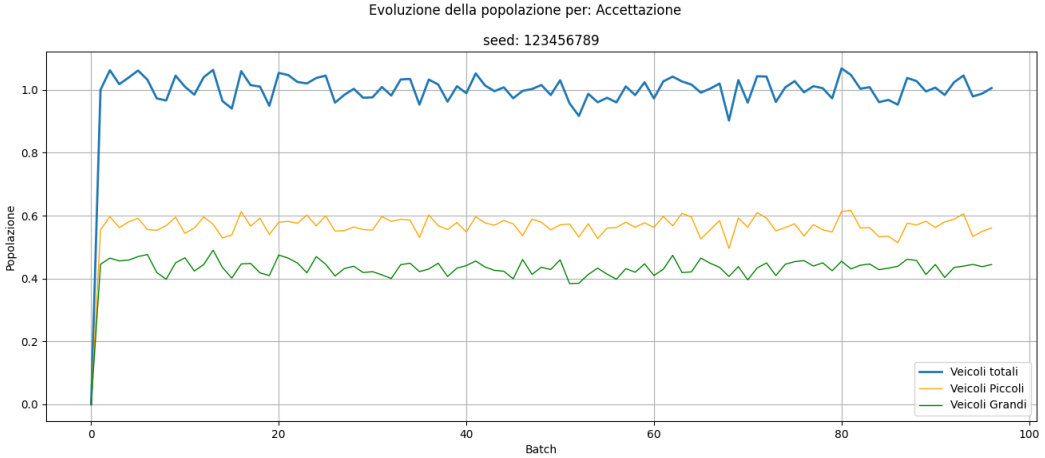
QoS violati 536 volte, a causa dei veicoli di tipo 2 (mezzi grandi) nelle 96 replicazioni totali.

Run n° 92 della simulazione

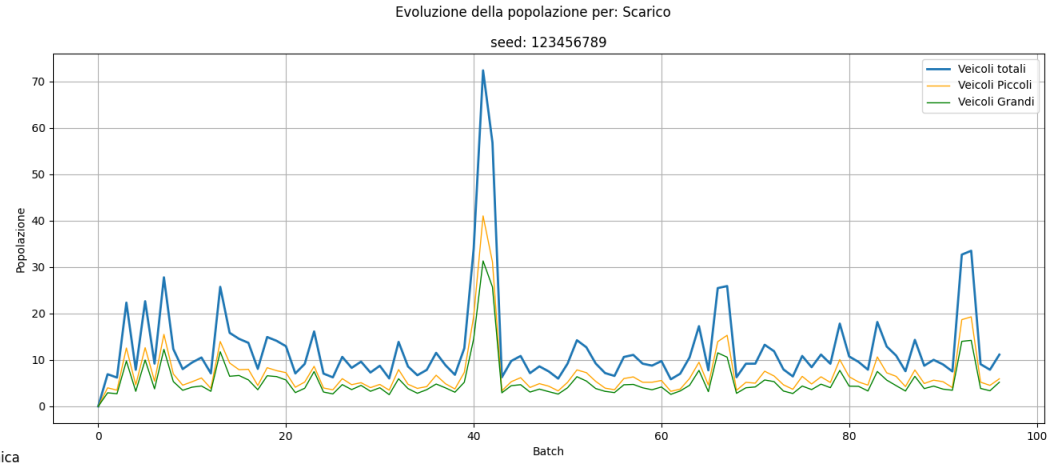


Simulazione ad orizzonte infinito

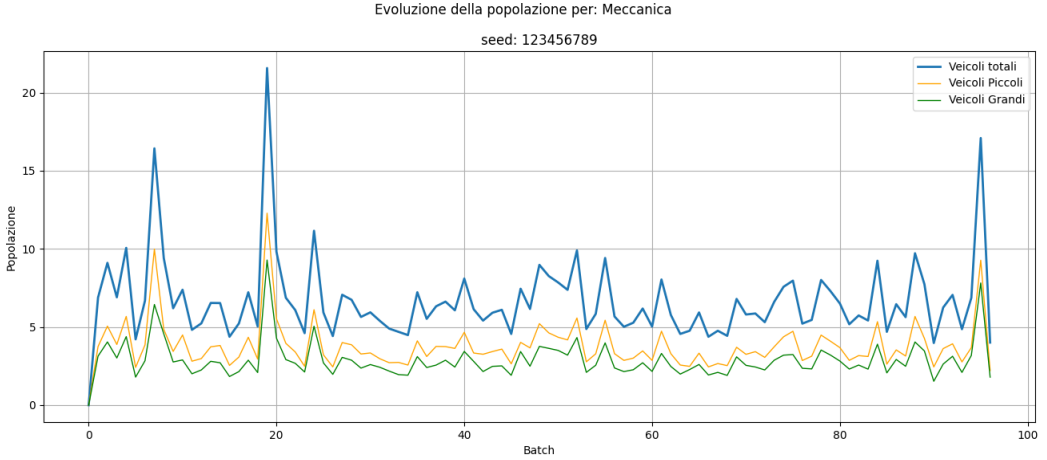
- Utilizzo della tecnica **"Batch Means"**, con *B = 1080*, *K = 96*
- Riduce l'influenza dello stato iniziale.



E[Ns] = 1.004 ± 0.022



E[Ns] = 12.657 ± 2.006

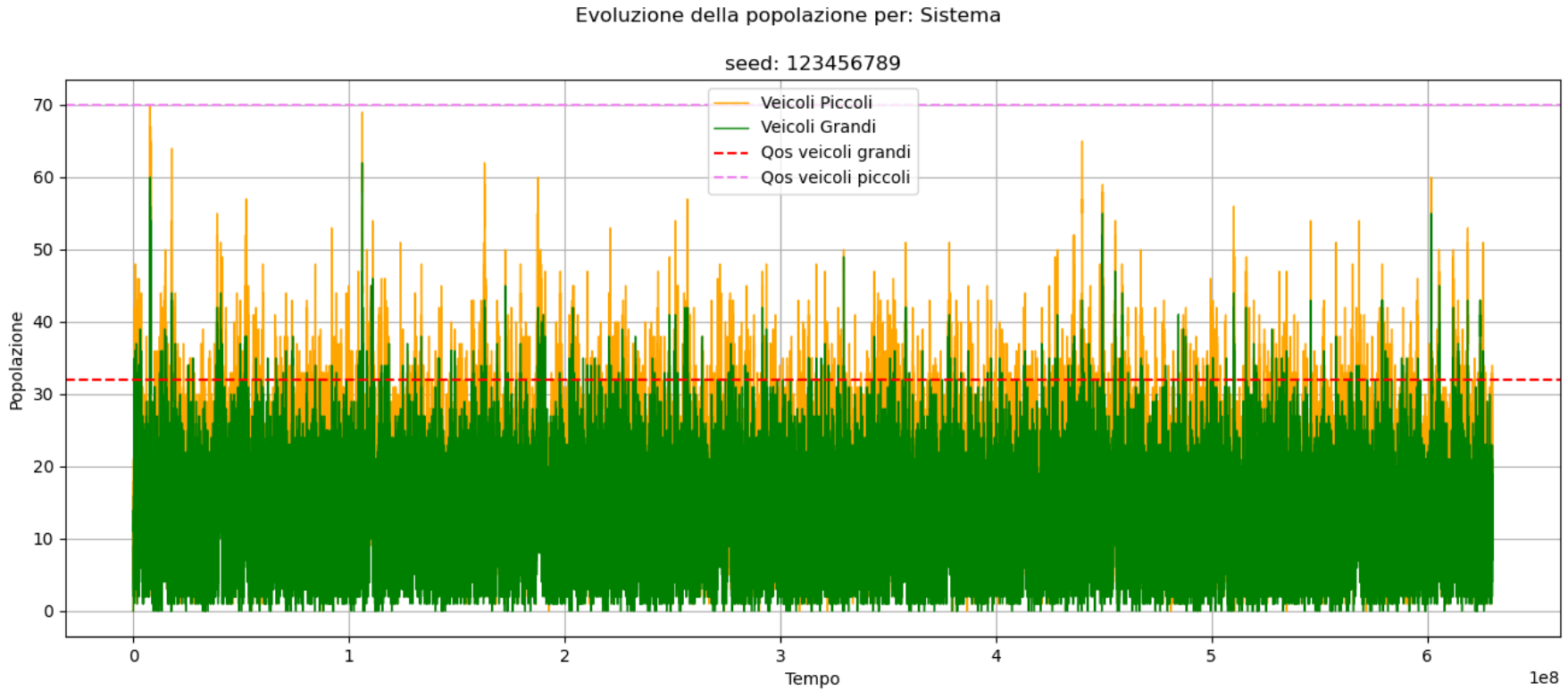


E[Ns] = 6.728 ± 0.553

Simulazione ad orizzonte infinito

```
Numero di volte in cui è stato superato il limite di veicoli nel sistema: 59285
Per i veicoli di tipo 1: 0 e per i veicoli di tipo 2:  59285
```

In media, nel sistema, troveremo 16 veicoli di piccole dimensioni e 12 veicoli di grandi dimensioni.



Modello migliorativo

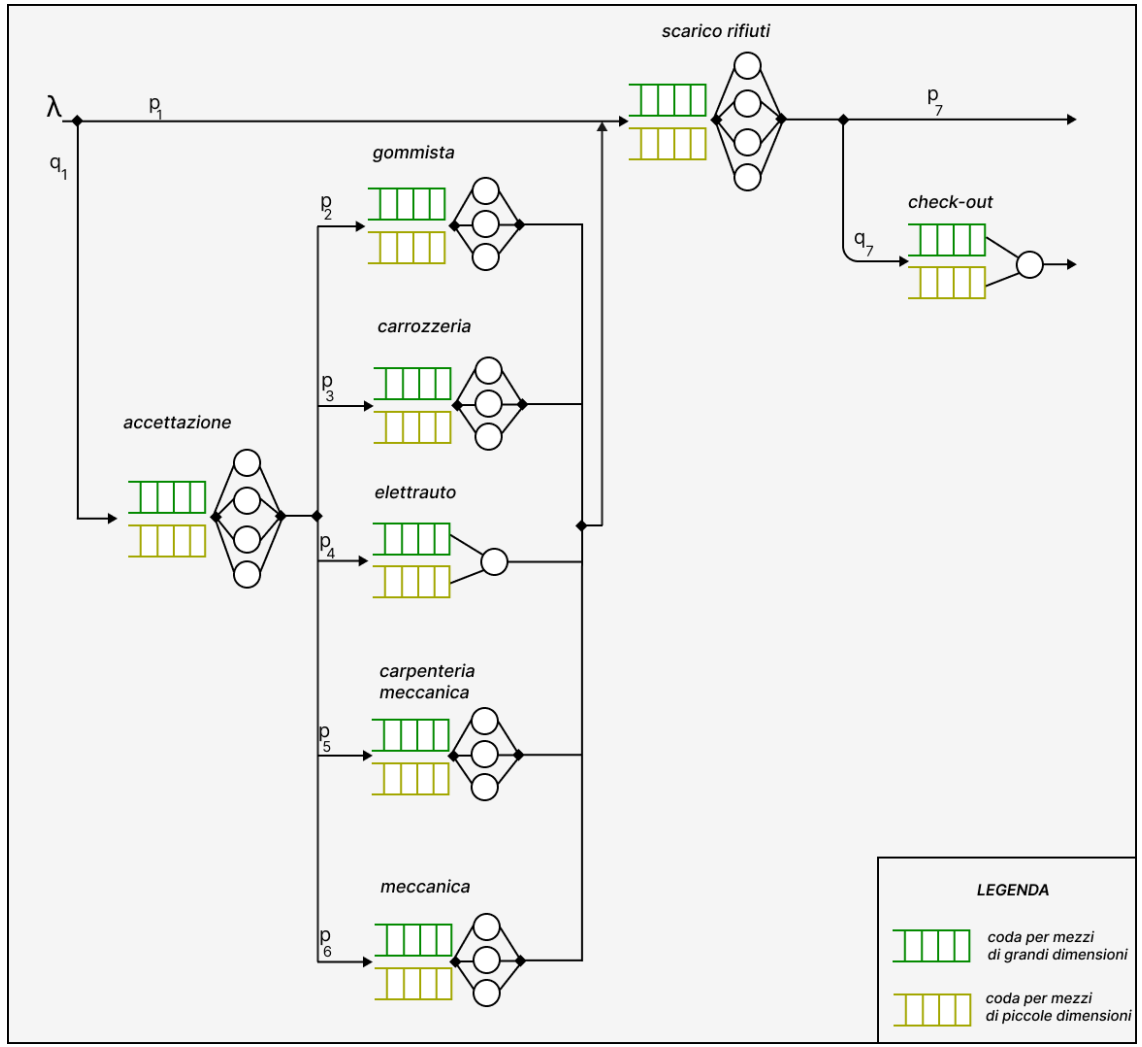
- La politica FIFO *favorisce i veicoli di piccole dimensioni*.
- Servizio sottodimensionato, per le aree che necessitano di veicoli grandi.

- **Proposta**: Rinnovare la gestione dei mezzi in attesa di un servizio.
- **Come**: Introducendo un controller generale che aggiorni la priorità *dinamicamente*, in base ai QoS.
- **Perché**: Necessità di un approccio che non vada a *snaturare* il polo impiantistico. Introduzione di veicoli e strumentazione non sono proposte reali.
- **Obiettivo**: rispettare i QoS illustrati precedentemente.

Veicolo di tipo 1

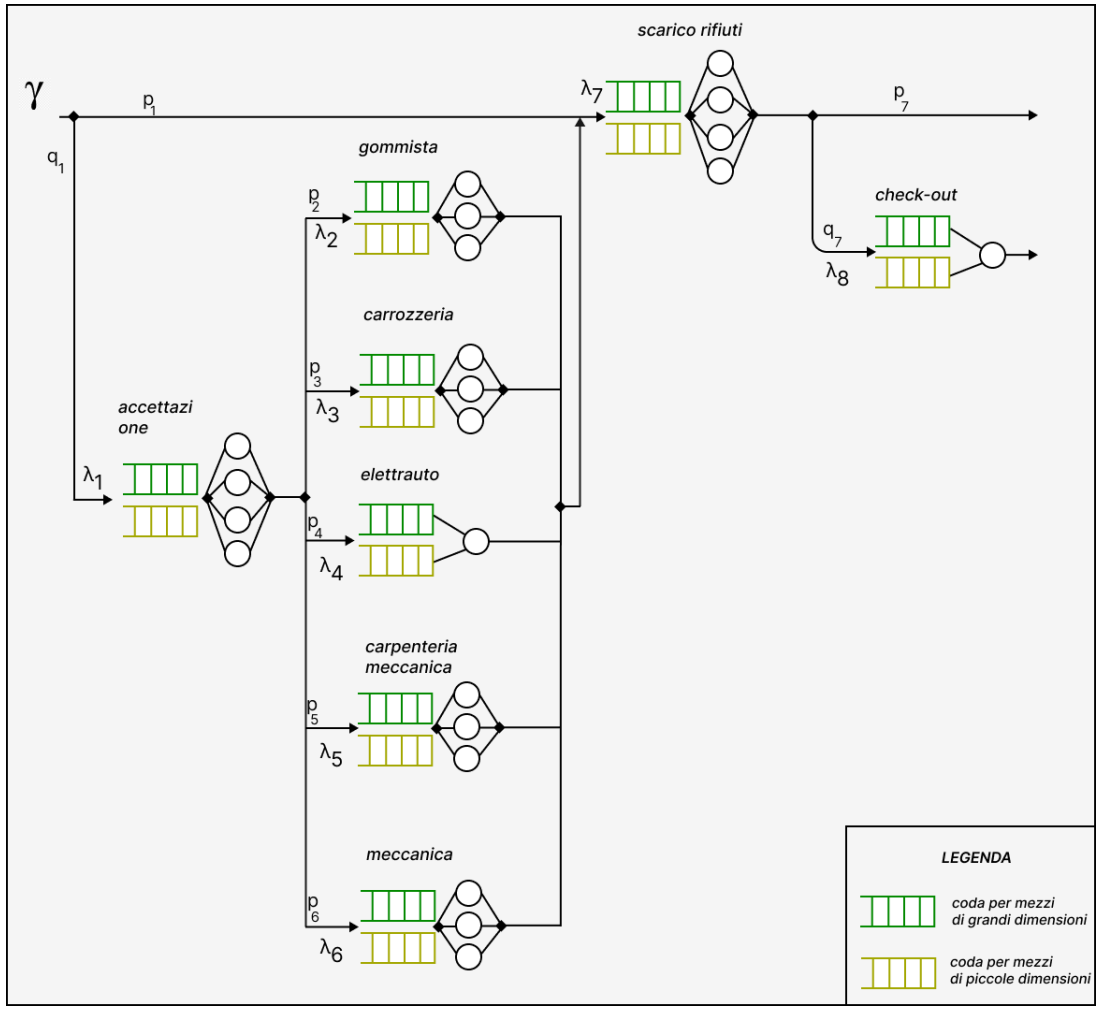
Veicolo di tipo 2

Modello concettuale



- Utenti mezzi (182: 103 piccoli, 79 grandi).
- Eventi:
  - Arrivo di un mezzo in uno specifico centro.
  - Uscita di un mezzo da uno specifico centro.
- Gestione code: priorità **dinamica** ed **adattativa** a seconda dello stato del sistema.

Modello delle specifiche



Equazioni di traffico:

- $\lambda_1 = (1 - p_1) \cdot \lambda$
- $\lambda_2 = p_2 \cdot \lambda_1$
- $\lambda_3 = p_3 \cdot \lambda_1$
- $\lambda_4 = p_4 \cdot \lambda_1$
- $\lambda_5 = p_5 \cdot \lambda_1$
- $\lambda_6 = p_6 \cdot \lambda_1$
- $\lambda_7 = \lambda$
- $\lambda_8 = (1 - p_7) \cdot \lambda_7$

Matrice di routing:

	Esterno	Accettazione	Gommista	Carrozzeria	Elettrauti	Carpenteria	Meccanica	Scarico	Checkout
Esterno	0	$1 - p_1$	0	0	0	0	0	p_1	0
Accettazione	0	0	p_2	p_3	p_4	p_5	p_6	0	0
Gommista	0	0	0	0	0	0	0	1	0
Carrozzeria	0	0	0	0	0	0	0	1	0
Elettrauti	0	0	0	0	0	0	0	1	0
Carpenteria	0	0	0	0	0	0	0	1	0
Meccanica	0	0	0	0	0	0	0	1	0
Scarico	p_7	0	0	0	0	0	0	0	$1 - p_7$
Checkout	1	0	0	0	0	0	0	0	0

Modello computazionale

- Invariato rispetto al precedente a parte per la gestione della priorità nelle code.
- Viene introdotto un contatore dei mezzi del sistema per ciascun tipo di veicoli.
- La classe *EventHandler* viene utilizzata come controller centrale per gestire la priorità delle code.

```
public int getNextEventFromQueue(List<EventListEntry> queue)
{
    for(int i=0;i<queue.size();i++){
        if(queue.get(i).getVehicleType()==this.priorityClass){
            return i;
        }
    }
    return 0;
}
```

Metodo per ottenere prossimo evento da gestire

```
if(eventHandler.getNumberV1()>MAX_VEICOLI1 || eventHandler.getNumberV2()>MAX_VEICOLI2){
    eventHandler.incrementSuperatoMax();
    if (eventHandler.getNumberV1()>MAX_VEICOLI1) eventHandler.incrementSuperatoMaxVeicoli1();
    else eventHandler.incrementSuperatoMaxVeicoli2();
}
if(eventHandler.getNumberV1()>=(MAX_VEICOLI1*BOUNDV1)){
    eventHandler.setPriorityClassV1();
}
if(eventHandler.getNumberV2()>=(MAX_VEICOLI2*BOUNDV2)){
    eventHandler.setPriorityClassV2();
}

if(eventHandler.getNumberV1()>MAX_VEICOLI1*BOUNDV1 && eventHandler.getNumberV2()>MAX_VEICOLI2*BOUNDV2){
    if ( (eventHandler.getNumberV1() - MAX_VEICOLI1*BOUNDV1) >= (eventHandler.getNumberV2() - MAX_VEICOLI2*BOUNDV2)){
        eventHandler.setPriorityClassV1();
    } else eventHandler.setPriorityClassV2();
}
```

Metodo per modificare priorità

Fase di verifica

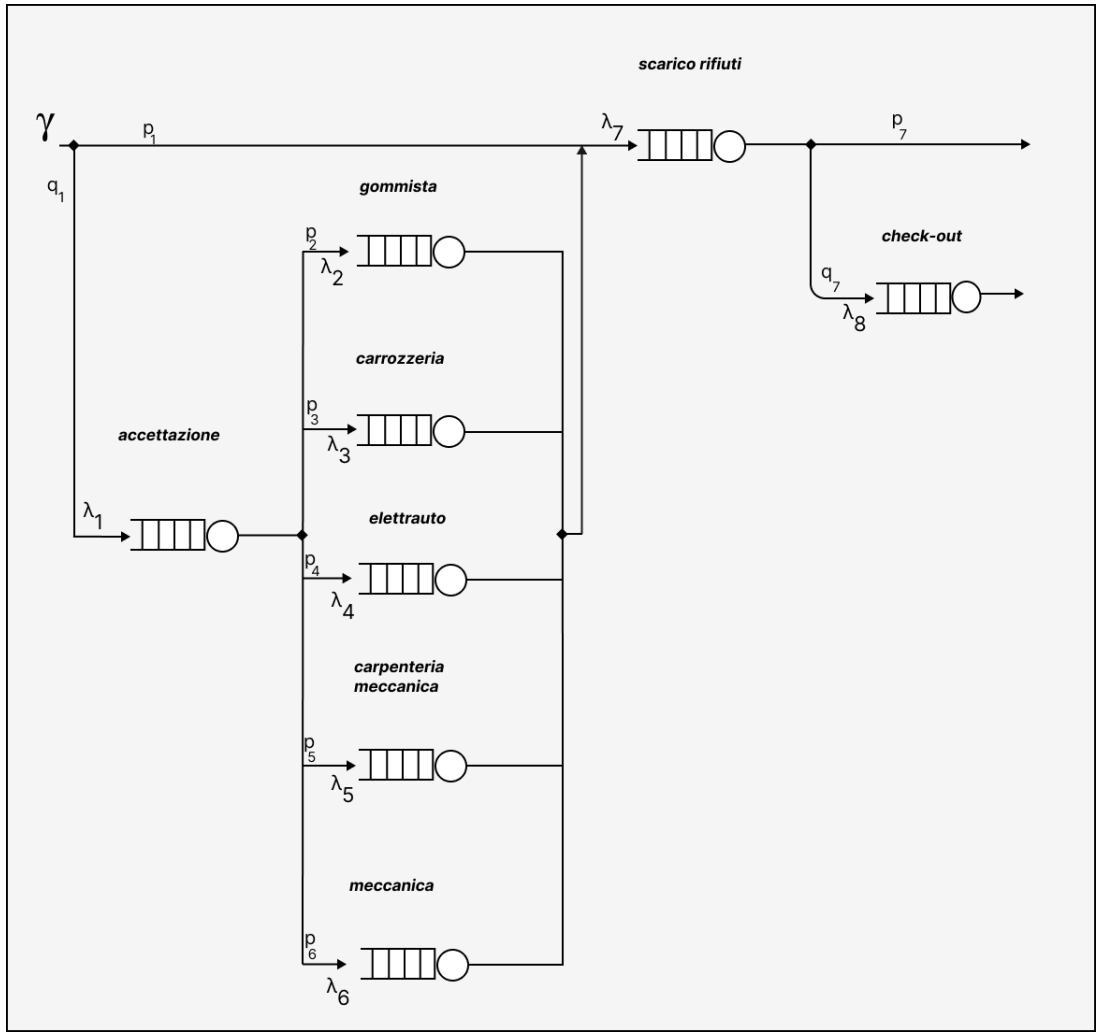
- Sfruttando distribuzioni esponenziali torniamo alla verifica precedente.
- L'introduzione di classi di priorità apporta miglioramenti solo nel caso locale.
- I tempi globali rimangono gli stessi.
- Criteri di consistenza considerati:
  - $E[T_s] = E[T_q] + E[S_i]$
  - $E[N_s] = E[N_q] + m \cdot \rho$
  - $0 < \rho \leq 1$

Validazione

- I tempi di interarrivo sono gli stessi ottenuti precedentemente dai report AMA.

- Viene modificato solo l'ordine con cui gli eventi sono serviti, si mantiene quindi consistenza tra dati reali e simulazione:
  - Nello scarico confluiscono tutti i mezzi presenti nel sistema, una volta chiuse le porte continuano ad arrivare i mezzi dalle officine.
  - Nell'accettazione ci si aspetta una popolazione ridotta e costante grazie ai tempi di servizio ridotti e ai numerosi serventi.
  - Nelle officine più utilizzate (meccanica ed elettrauto) ci aspettiamo la maggior parte della popolazione del sistema (lunghi tempi di servizio).

Analisi del collo di bottiglia



- Servizio medio dei centri del sistema:

$S_1 = \frac{1}{\mu_1} = 600\,\text{sec}$ $S_5 = \frac{1}{\mu_5} = 5400\,\text{sec}$

$S_2 = \frac{1}{\mu_2} = 3600\,\text{sec}$ $S_6 = \frac{1}{\mu_6} = 5400\ sec$

$S_3 = \frac{1}{\mu_3} = 5400\,\text{sec}$ $S_7 = \frac{1}{\mu_7} = 600\ sec$

$S_4 = \frac{1}{\mu_4} = 5400\,\text{sec}$ $S_8 = \frac{1}{\mu_8} = 1200\ sec$

- Numero medio di visite ai centri del sistema:

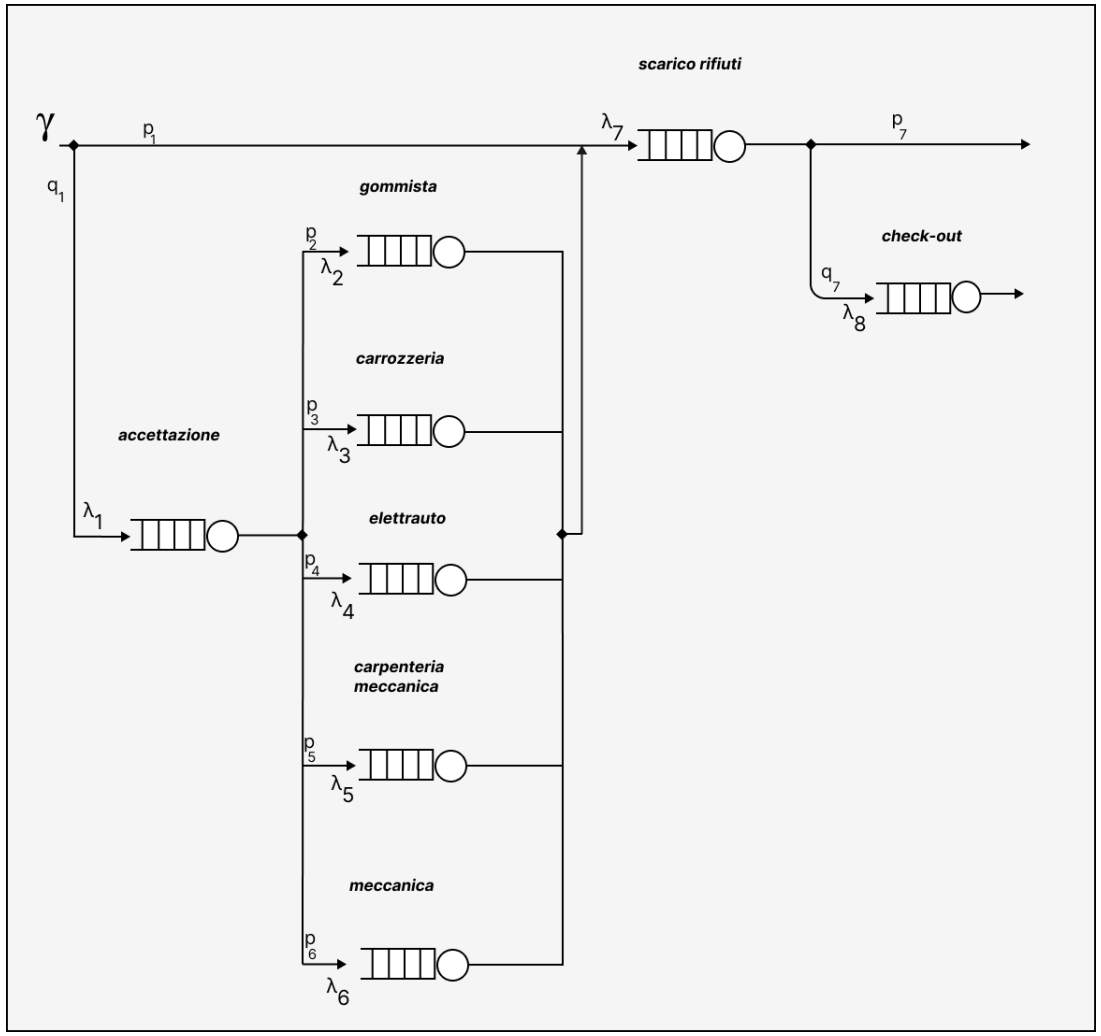
$v_1 = \frac{\lambda_1}{\gamma} = 0.3$ $v_5 = \frac{\lambda_5}{\gamma} = 0.06$

$v_2 = \frac{\lambda_2}{\gamma} = 0.09$ $v_6 = \frac{\lambda_6}{\gamma} = 0.09$

$v_3 = \frac{\lambda_3}{\gamma} = 0.03$ $v_7 = \frac{\lambda_7}{\gamma} = 1$

$v_4 = \frac{\lambda_4}{\gamma} = 0.03$ $v_8 = \frac{\lambda_8}{\gamma} = 0.1$

Analisi del collo di bottiglia



- Domanda media per i centri del sistema:

$D_1 = v_1 \cdot S_1 = 180\ sec$ $\quad D_5 = v_5 \cdot S_5 = 324\ sec$

$D_2 = v_2 \cdot S_2 = 324\ sec$ $\quad D_6 = v_6 \cdot S_6 = 486\ sec$

$D_3 = v_3 \cdot S_3 = 162\ sec$ $\quad D_7 = v_7 \cdot S_7 = 600\ sec$

$D_4 = v_4 \cdot S_4 = 162\ sec$ $\quad D_8 = v_8 \cdot S_8 = 120\ sec$

- I valori calcolati sono coerenti con quelli ottenuti dalla simulazione:

	Accettazione	Gommista	Carrozzeria	Elettrauti	Carpenteria	Meccanica	Scarico	Checkout
V	0.300158	0.090122	0.029928	0.029888	0.059984	0.090235	1	0.099851
D (sec)	180.094748	324.439330	161.610493	161.395675	323.915256	487.270761	600	119.821711

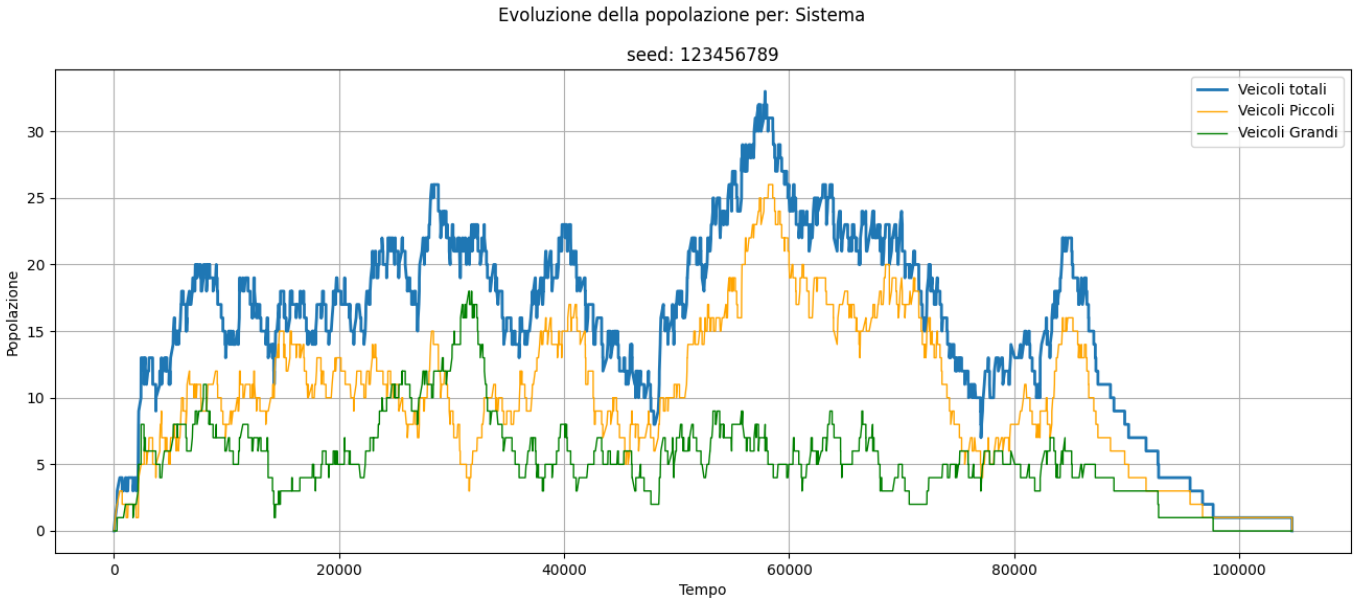
- Da questi risultati si nota che è lo scarico il collo di bottiglia.

Simulazione ad orizzonte finito

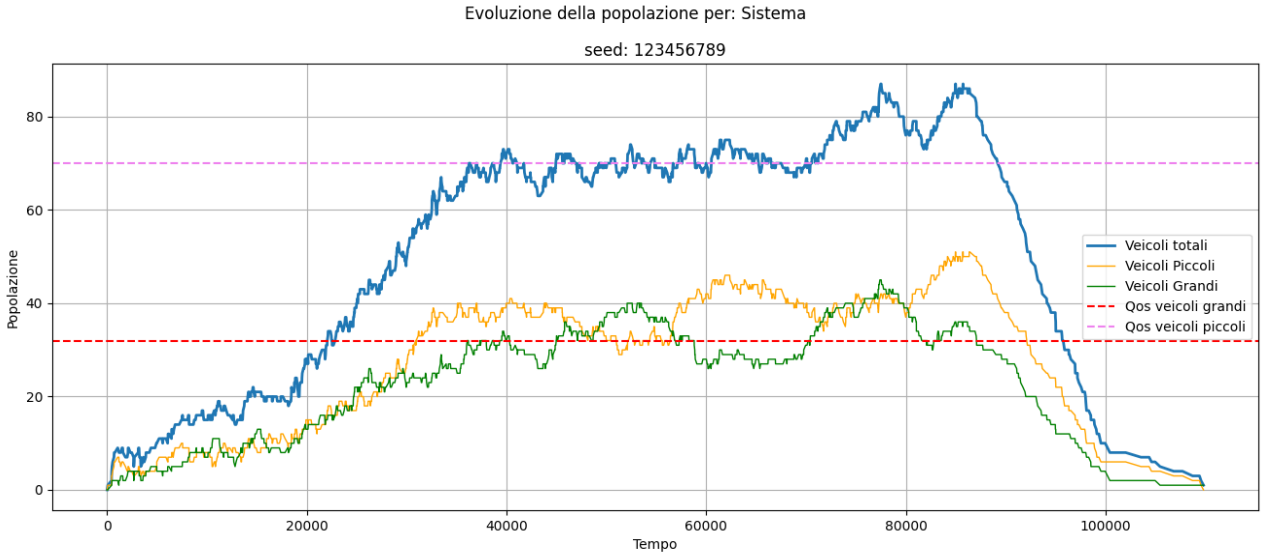
- Uso della tecnica di **Replicazione**, 96 repliche indipendenti.
- **Intervallo di confidenza** calcolato come $t^* \cdot \left(\frac{s}{\sqrt{n-1}} \right)$

QoS non vengono mai violati nelle 96 replicazioni totali.

N.B. modello transiente poco accurato rispetto al funzionamento dell'azienda

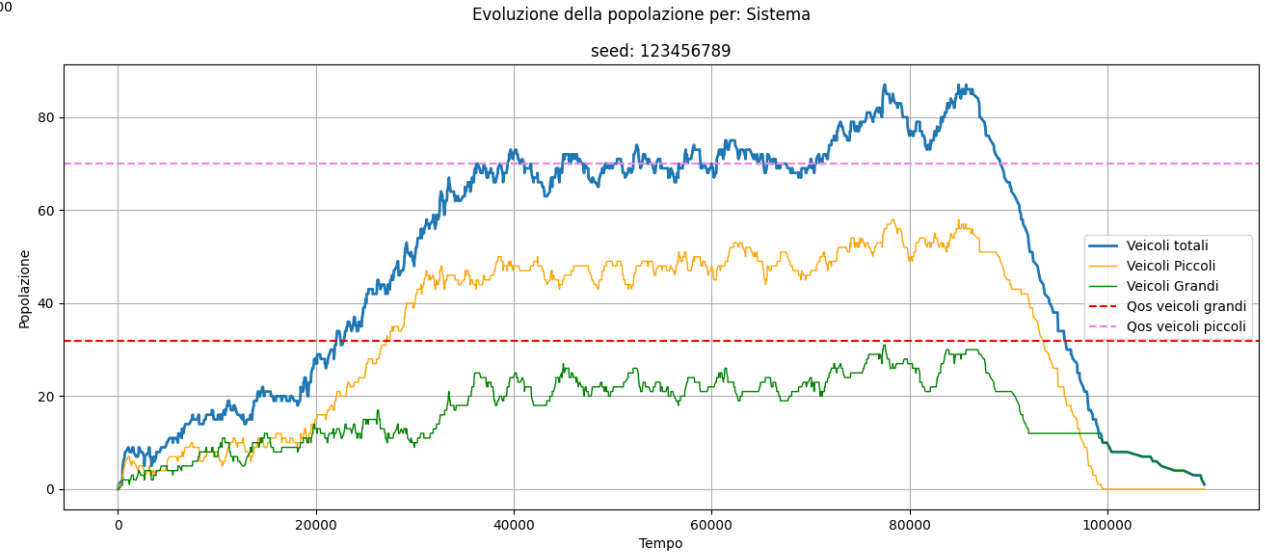


Simulazione ad orizzonte finito



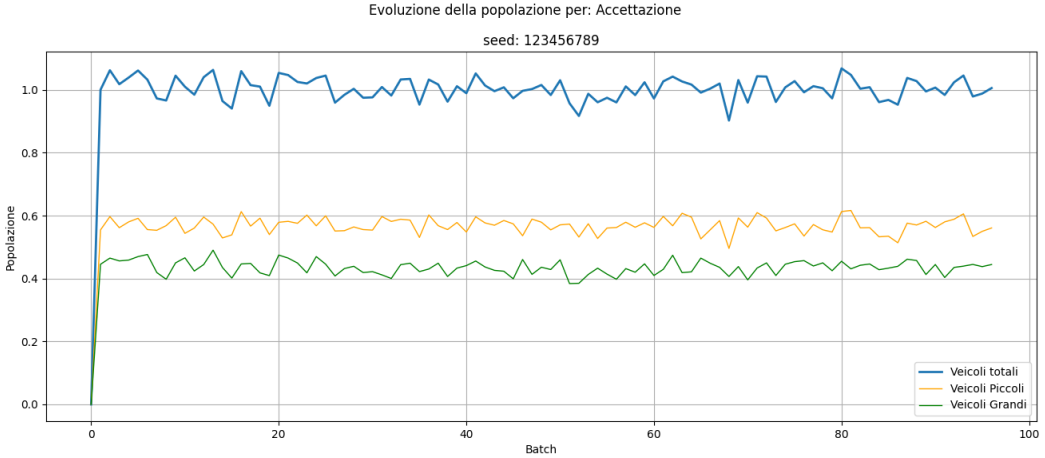
Modello base: run n° 92

Modello migliorativo: run n° 92

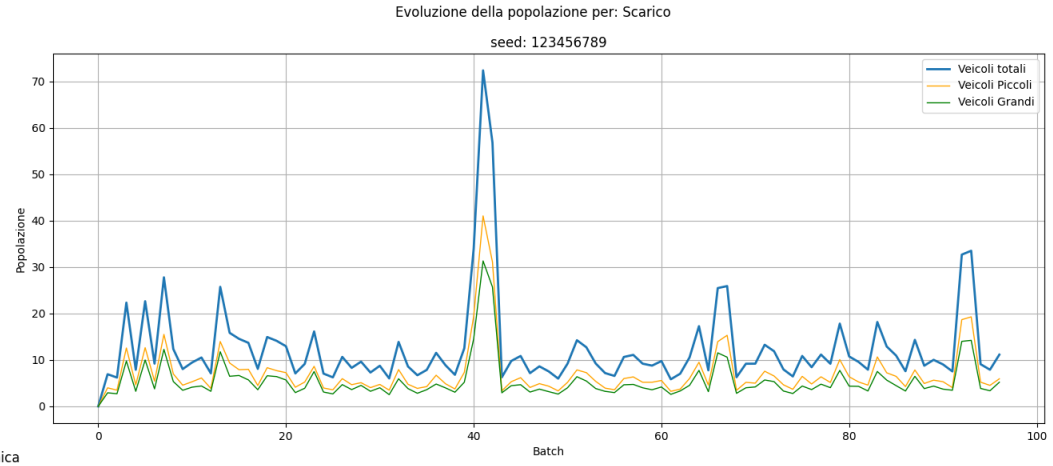


Simulazione ad orizzonte infinito

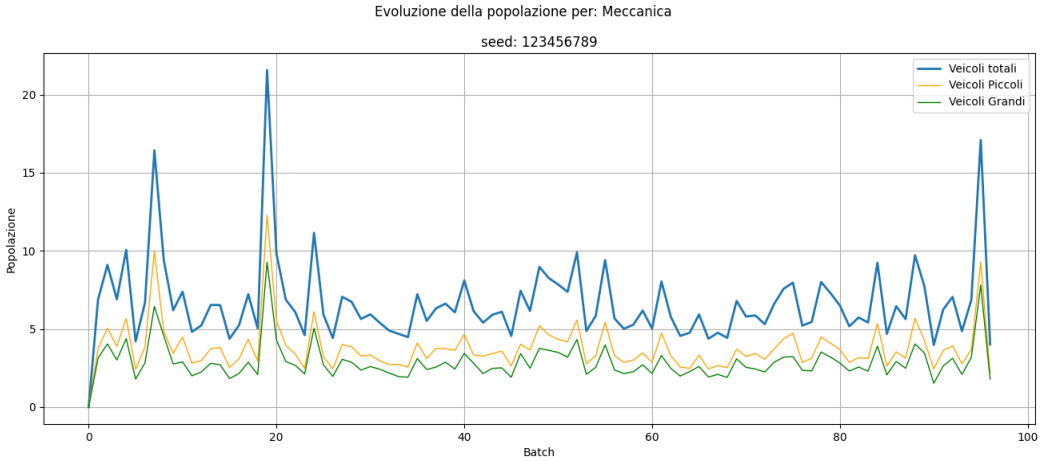
- Utilizzo della tecnica **"Batch Means"**, con *B = 1080*, *K = 96*
- Riduce l'influenza dello stato iniziale.



E[Ns] = 1.004 ± 0.022



E[Ns] = 12.657 ± 2.006

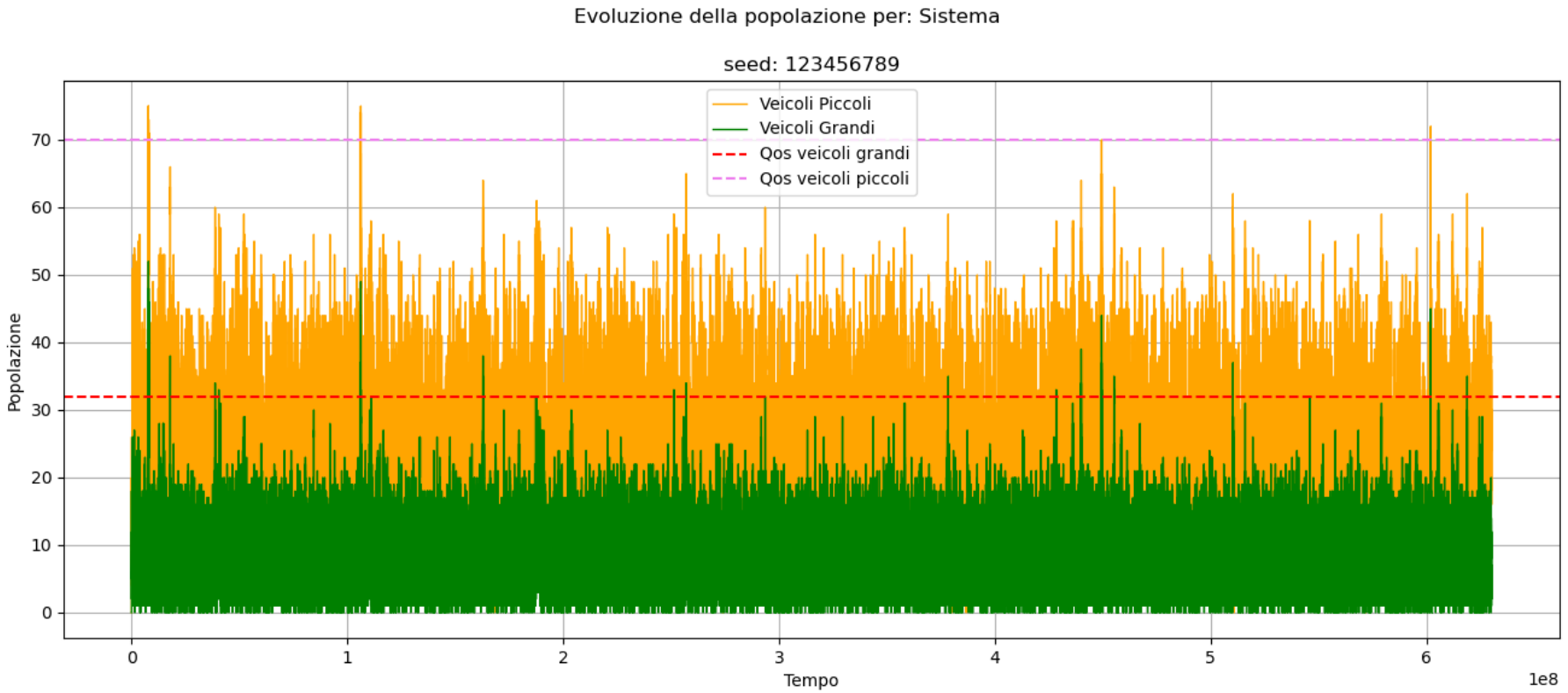


E[Ns] = 6.728 ± 0.553

Simulazione ad orizzonte infinito

```
Numero di volte in cui è stato superato il limite di veicoli nel sistema: 10184
Per i veicoli di tipo 1: 798 e per i veicoli di tipo 2:  9386
```

In media, nel sistema, troveremo 20 veicoli di piccole dimensioni e 8 veicoli di grandi dimensioni.



Conclusione

- Gestione diversa delle code porta a un miglioramento dell'83%.
- Lieve peggioramento per i veicoli di tipo 1 ma netto miglioramento di quelli di tipo 2.
- Si è fatto uso di priorità dinamica poiché:
  - Veicoli di tipo 1 non riuscirebbero, a causa della loro capacità minore, a fornire la stessa efficienza di quelli di tipo 2.
  - Veicoli di tipo 2 non possono completare tutti i servizi dei veicoli di tipo 1.

Grazie per l'attenzione!

Simone Festa - Michele Tosi
Università degli Studi di Roma "Tor Vergata"
Facoltà di Ingegneria

https://github.com/simonefesta/AMA-ROMA_PMCSN