

Gestione Di Una Mensa Aziendale

Di Battista Mattia (0304938)¹ and Rossi Giacomo Lorenzo (0292400)²

¹mattia.dibattista@alumni.uniroma2.eu

²giacomolorenzo.rossi@alumni.uniroma2.eu

Sommario

In questo report è presentato il caso di studio di una mensa aziendale come progetto per il corso di **Performance Modeling Of Computer Systems And Networks**. L'obiettivo è analizzare l'attività del sistema nel tempo rapportandolo con le metriche di qualità definite.

1 Introduzione

In questa sezione è data una descrizione del sistema in analisi e gli obiettivi che lo studio si propone.

1.1 Descrizione

Il sistema considerato è una mensa aziendale composta da una zona ristorazione, dove un utente può scegliere il suo pasto e acquistarlo, e una successiva area dove mangiare costituita da diversi posti a tavola.

Le tipologie di piatti che un utente può scegliere sono:

- *primo*;
- *secondo e/o contorno*;
- *dessert e/o frutta*;

Per semplificare il modello, si assume che un utente non possa consumare soltanto un *dessert e/o frutta*. Dunque le possibili combinazioni di pietanze che un utente può scegliere sono:

- {*primo*};
- {*secondo e/o contorno*};
- {*primo, secondo e/o contorno*};
- {*primo, dessert e/o frutta*};
- {*secondo e/o contorno, dessert e/o frutta*};

- $\{primo, secondo e/o contorno, dessert e/o frutta\}$;

Dopo aver scelto una o più portate, ogni dipendente si dirige alle casse per il pagamento, distinte in:

- *fast*, riservate a chi sceglie una sola portata;
- *standard*, riservate a chi sceglie più di una portata;

Questa distinzione comporta che un utente paga nelle casse *fast* se proviene direttamente dal *primo*, o se proviene dal *secondo e/o contorno* senza però essere passato per il *primo*. D'altro canto, un dipendente arriva necessariamente alle casse *standard* se ha preso *dessert e/o frutta*.

Infine, l'utente sceglie un posto a sedere per consumare il pasto. Se tutti posti sono occupati, l'utente decide di mangiare altrove.

In Fig. 1 è riportato lo schema di suddivisione della mensa.

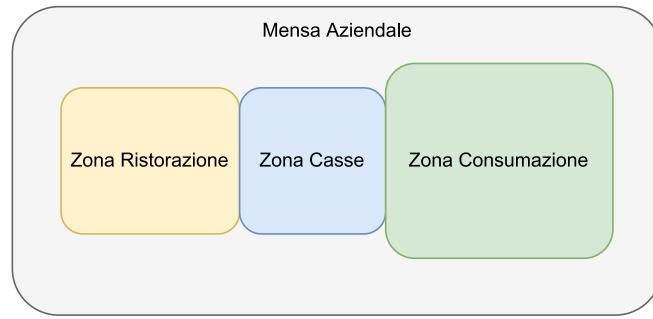


Figura 1: Schema di suddivisione della mensa.

1.2 Obiettivi

L'obiettivo è analizzare le metriche (e.g., tempi di attesa in coda, utilizzazioni) del modello, considerando i seguenti requisiti di qualità:

- QoS-1: il tempo di risposta del sistema deve essere inferiore a 15 min¹;
- QoS-2: la percentuale di utenti che non trovano un posto a sedere deve essere inferiore al 5%;

2 Modello Concettuale

Nella seguente sezione è presentato il modello concettuale descrivendone i diversi blocchi, le variabili di stato e gli eventi.

¹Il requisito è dovuto ai tempi limitati che un dipendente ha solitamente a disposizione per mangiare in una azienda.

2.1 Descrizione del Modello

Come discusso in Sec. 1.1 i blocchi del sistema sono²:

1. *PRIMO*;
 2. *SECONDO*;
 3. *DESSERT*;
 4. *CASSA FAST*;
 5. *CASSA STD*;
 6. *CONSUMAZIONE*

I primi cinque blocchi sono modellati come $M/M/m$, mentre il centro *CONSUMAZIONE* è implementato come $M/M/m/m$. In Fig. 2 è mostrato il modello concettuale. In seguito verrà descritto nel dettaglio il procedimento per determinare il numero di serventi in ogni blocco.

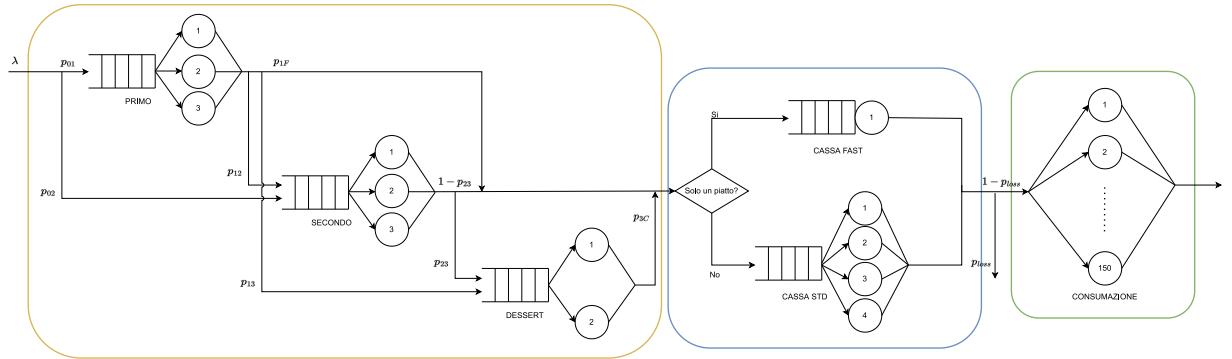


Figura 2: Modello concettuale per una mensa aziendale.

2.2 Variabili di stato

Le **variabili di stato** per definire lo stato del nodo sono:

- Lo stato di attività per ogni servente di ciascun centro (i.e., idle, busy), indicato con:

$$x_{ij} = \text{idle}/\text{busy}, \quad \forall i \in \{1, 2, 3, F, S, C\}, \quad \forall j \in \{1, 2, \dots, m_i\}$$

Dove i rappresenta il nodo, j un particolare servente del nodo e m_i il numero di serventi per il nodo.

²Per comodità, nella discussione che segue *secondo e/o contorno e dessert e/o frutta* sono abbreviati rispettivamente con *SECONDO* e *DESSERT*.

- Il numero di job totali in ogni centro, indicato come:

$$l_i(t), \forall i \in \{1, 2, 3, F, C, S\}$$

- Il numero di job in coda in ogni centro, indicato come:

$$q_i(t), \forall i \in \{1, 2, 3, F, C, S\}$$

Ne segue che il numero di job in servizio nel centro i è ricavato con la seguente formula:

$$x_i(t) = l_i(t) - q_i(t), \forall i \in \{1, 2, 3, F, C, S\}$$

Ciò consente di evitare di introdurre il numero di job in servizio come variabile di stato, semplificando il modello.

2.3 Eventi

Gli eventi simulati nel sistema sono:

- Arrivo di un utente dall'esterno nel centro *PRIMO* o *SECONDO*;
- Completamento del servizio di un utente nel servente j per un particolare centro i ;

3 Modello delle specifiche

In questa sezione verranno presentati i dati di input al sistema e le statistiche prodotte, nonché i passaggi matematici per ottenerle.

3.1 Dati di Input

I tassi di arrivo e completamento per ciascun utente nel sistema sono stati modellati con una distribuzione **esponenziale**³, mentre la disciplina di scheduling in ciascuna coda è **FIFO**. Infine i servizi sono **non-preemptive** e **work-conserving**.

³Ciò è dovuto al fatto che ogni evento del sistema (i. e. arrivo o completamento) è casuale.

3.1.1 Probabilità di routing

Per prima cosa si riportano le **probabilità di routing** del sistema.

- Probabilità di routing dall'esterno:

$$\begin{aligned} - P(\text{entrata nel PRIMO}) &= p_{0,1} = 0.75; \\ - P(\text{entrata nel SECONDO}) &= p_{0,2} = 0.25; \end{aligned}$$

- Probabilità di routing dal *PRIMO*:

$$\begin{aligned} - P(\text{entrata nelle casse FAST}) &= p_{1,F} = 0.20; \\ - P(\text{entrata nel SECONDO}) &= p_{1,2} = 0.55; \\ - P(\text{entrata nel DESSERT}) &= p_{1,3} = 0.25; \end{aligned}$$

- Probabilità di routing dal *SECONDO*:

$$\begin{aligned} - P(\text{entrata nel DESSERT}) &= p_{2,3} = 0.45; \\ - P(\text{entrata nelle casse}) &= 1 - p_{2,3} = 0.55; \\ * P(\text{entrata nelle casse FAST}) &= p_{2,F} = 0.1375; \\ * P(\text{entrata nelle casse STANDARD}) &= p_{2,C} = 0.4125; \end{aligned}$$

- Probabilità di routing daL *DESSERT*:

$$- P(\text{entrata nelle casse STANDARD}) = p_{3,C} = 1;$$

Dai dati appena descritti è possibile derivare le seguenti ulteriori probabilità.

- Probabilità che un utente scelga una pietanza:

$$- P(\#\text{pietanze} = 1) = p_{0,1} \cdot p_{1,F} + p_{0,2} \cdot (1 - p_{2,3}) = 0.75 \cdot 0.2 + 0.25 \cdot 0.55 = 0.2875;$$

- Probabilità che un utente scelga almeno due pietanze:

$$\begin{aligned} - P(\#\text{pietanze} > 1) &= P(\#\text{pietanze} = 2) + P(\#\text{pietanze} = 3) = \\ &= (p_{0,1} \cdot p_{1,2} \cdot (1 - p_{2,3}) + p_{0,2} \cdot p_{2,3} + p_{0,1} \cdot p_{1,3}) + (p_{0,1} \cdot p_{1,2} \cdot (1 - p_{2,3})) = \\ &= (0.75 \cdot 0.55 \cdot 0.55 + 0.25 \cdot 0.45 + 0.75 \cdot 0.25) + (0.75 \cdot 0.55 \cdot 0.45) = \\ &= 0.7125; \end{aligned}$$

- In alternativa:

$$- P(\#\text{pietanze} > 1) = 1 - P(\#\text{pietanze} = 1) = 1 - 0.2875 = 0.7125;$$

3.1.2 Tasso di arrivo

L'orario di operatività della mensa è [12:00-15:00] con un numero di utenti medi giornalieri pari a 2500, ottenendo così un **tasso di arrivo** pari a:

$$\lambda = \frac{4 \cdot 60 \cdot 60}{2500} = 0.231481 \frac{\text{arrivi}}{\text{s}}$$

Inizialmente nessun utente è presente nel sistema e al termine dell'operatività, si attende che tutti finiscano di pranzare per concludere la simulazione. Per calcolare i singoli tassi di arrivo, si sfruttano lo schema della rete, le probabilità di routing viste nel paragrafo precedente e il tasso di arrivo globale nella rete λ :

- Per il *PRIMO*, $\lambda_1 = \lambda \cdot p_{01} = 0.1735$
- Per il *SECONDO*, $\lambda_2 = \lambda \cdot p_{02} + \lambda_1 \cdot p_{12} = 0.1534$
- Per il *DESSERT*, $\lambda_3 = \lambda_1 \cdot p_{13} + \lambda_2 \cdot p_{23} = 0.1124$
- Per le *CASSE FAST*, $\lambda_F = \lambda_1 \cdot p_{1F} + \lambda_2 \cdot p_{2F} = 0.0558$
- Per le *CASSE STD*, $\lambda_C = \lambda_1 \cdot p_{1C} + \lambda_2 \cdot p_{2C} + \lambda_3 \cdot p_{3C} = 0.1757$
- Per *CONSUMAZIONE*, $\lambda_S = \lambda \cdot (1 - p_{loss}) = 0.2256 \cdot (1 - 0.025197) = 0.2256$

Il calcolo della probabilità di perdita per il centro *CONSUMAZIONE* verrà spiegato nel paragrafo 3.2.2.

3.1.3 Tempi di servizio

I **tempi di servizio** per ciascun servente di ogni centro sono riportati in Tab. 3.1.3.

Tipologia servente	Tempo di servizio medio (s)	tasso di servizio (1/s)
<i>PRIMO</i>	15	0.06667
<i>SECONDO</i>	15	0.06667
<i>DESSERT</i>	10	0.10000
<i>CASSA FAST</i>	11	0.0909
<i>CASSA STD</i>	18	0.05554
<i>CONSUMAZIONE</i>	600	0.00166

3.1.4 Utilizzazioni

Noti i tassi di arrivo e di servizio per i singoli centri, adesso è possibile determinare le **utilizzazioni**, verificare quali centri sono **ergodici** e quali no e infine determinare il numero minimo di server affinché tutta la rete sia stabile.

Le utilizzazioni dei singoli centri $M/M/m$ possono essere calcolate con la seguente formula generale:

$$\rho_i = \frac{\lambda_i \cdot E(S_i)}{m_i}$$

Per il centro *CONSUMAZIONE* (con tipologia $M/M/m/m$) va considerato il tasso di arrivo dei soli utenti accettati (i clienti rigettati non vengono inclusi):

$$\rho_S = \frac{\lambda_S \cdot E(S_i)}{m_i}$$

Sia R l'utilizzo di risorse, dato dalla seguente formula:

$$R = \frac{\lambda}{\mu}$$

Questo valore è necessario per calcolare il numero minimo m di server affinché il singolo blocco sia stabile:

$$m = \lceil R \rceil$$

La tabella seguente riassume quali centri siano ergodici e quali no, e il numero minimo di server m affinché il centro sia ergodico:

Centro	λ	μ	Ergodico?	R	m	ρ
<i>PRIMO</i>	0.1735	0.06667	No	2.60228	3	0.86746
<i>SECONDO</i>	0.1534	0.06667	No	2.30086	3	0.76698
<i>DESSERT</i>	0.1124	0.10000	No	1.12396	2	0.56197
<i>CASSA FAST</i>	0.0558	0.0909	Sì	0.61388	1	0.61388
<i>CASSA STD</i>	0.1757	0.05554	No	3.16356	4	0.79086
<i>CONSUMAZIONE</i>	0.2256	0.00166	No	135.90434	136	0.99734

Notiamo che l'utilizzazione per *CONSUMAZIONE* è elevata, quindi si è deciso di aumentare il numero di serventi a 150.

3.2 Statistiche di Output

Vengono ora riportati i calcoli per le statistiche di output di interesse (i.e., tempo di risposta e probabilità di perdita).

3.2.1 Tempi di Risposta

Per l'unico blocco $M/M/1$ (i.e., *CASSA_FAST*) è usata la seguente formula:

$$E(T_{S,F}) = \frac{1}{\mu - \lambda} = \frac{1}{0.0909 - 0.0558} = 28.49011$$

Nei blocchi $M/M/m$ è invece necessario utilizzare la formula di **Erlang-C** con coda infinita per ottenere prima la probabilità di blocco⁴ e poi ricavare il tempo di risposta. Le formule sono le seguenti:

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

⁴Ovvero la probabilità che tutti i serventi del sistema siano pieni

dove $p(0)$ rappresenta la probabilità che il sistema sia vuoto, calcolata com:

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

L'utilizzazione ρ va calcolata tenendo conto del numero di server presenti nel blocco:

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

Per calcolare il tempo di risposta si utilizza la seguente formula, che somma il tempo medio in coda al tempo medio di servizio del singolo servente:

$$E(T_S) = \frac{P_Q \cdot E(S)}{1 - \rho} + E(S_i)$$

con $E(S) = 1/(m \cdot \mu)$ (tempo per liberare uno qualsiasi dei serventi) e $E(S_i) = 1/\mu$ (tempo di servizio medio del singolo servente).

Rimane da calcolare il tempo di risposta per il blocco *CONSUMAZIONE*. Poichè si tratta di un centro $M/M/m/m$ senza coda, non è necessario calcolare il tempo di attesa medio in coda, in quanto $E(T_{Q,S}) = 0$. Quindi il tempo di risposta è semplicemente dato dal tempo di servizio medio di un singolo servente:

$$E(T_{S,S}) = E(S_S) = \frac{1}{\mu_S}$$

I risultati sono visibili in Tab. 3.2.1.

Centro	Tipo	$E(T_S)$
<i>PRIMO</i>	$M/M/3$	43.8488
<i>SECONDO</i>	$M/M/3$	27.7346
<i>DESSERT</i>	$M/M/2$	14.6181
<i>CASSA FAST</i>	$M/M/1$	28.49011
<i>CASSA STD</i>	$M/M/4$	30.4489
<i>CONSUMAZIONE</i>	$M/M/150/150$	600.0

Infine, si riporta il calcolo del tempo di risposta globale dell'intera rete:

$$\begin{aligned}
 E(T_r^{(t)}) &= \sum_{i=1}^M E(T_{S,i}) \cdot v_i \\
 &= E(T_{S,1}) \cdot v_1 + E(T_{S,2}) \cdot v_2 + E(T_{S,3}) \cdot v_3 + E(T_{S,F}) \cdot v_F + E(T_{S,C}) \cdot v_C + E(T_{S,S}) \cdot v_S \\
 &= 43.8488 \cdot 0.75 + 27.7346 \cdot 0.6625 + 14.6181 \cdot 0.4856 + 28.49011 \cdot 0.2411 + 30.4489 \cdot 0.7589 + 600.0 \cdot 0.9748 = \\
 &= 673.21152 \text{ s}
 \end{aligned}$$

3.2.2 Probabilità di Perdita

La probabilità di perdita p_{loss} del blocco *CONSUMAZIONE* è data dalla formula **Erlang-B**, con $m = 150$, $\lambda = 0.231481$ e $\mu = 0.00167$:

$$p_{loss} = \frac{\frac{1}{m!} \cdot \left(\frac{\lambda}{\mu}\right)^m}{\sum_{i=0}^m \frac{1}{i!} \cdot \left(\frac{\lambda}{\mu}\right)^i} = 0.025197$$

Riassumendo, i valori teorici delle due statistiche di cui si cerca di garantire il QoS sono i seguenti:

Statistica	Valore teorico	QoS
$E(T_r)$	673.21152 s	900.0 s
p_{loss}	0.025197	0.05

4 Modello Computazionale

In quanto segue è data una descrizione del sistema a livello implementativo seguendo la **next-event simulation**⁵.

4.1 Clock di Sistema

Il primo passo della simulazione è inizializzare il clock di sistema generando il primo arrivo dall'esterno con la funzione `get_next_arrival()`, che a sua volta invoca `Exponential()`. Il primo evento di arrivo così generato viene associato ad un particolare blocco tra *PRIMO* o *SECONDO*, in base al valore generato da `Random()` e alle probabilità di routing.

La struttura dati usata per rappresentare il clock di sistema è:

```
typedef struct clock_t_n {
    double last_arrival;
    double current;
    double last;
    int type;
} timer;
```

L'attributo `current` mantiene il tempo corrente corrispondente all'evento da processare (i.e., arrivo, completamento), mentre `last_arrival` rappresenta l'istante dell'ultimo evento d'arrivo da processare, impostato a un valore INF quando la simulazione termina. L'istante dell'ultimo evento processato è `last`. Se l'evento corrente dovesse essere successivo al periodo di operatività (i.e., 3 ore) del sistema e non ci sono completamenti da eseguire, la simulazione termina.

4.2 Gestione degli Eventi

Gli eventi descritti in Sec. 2.3 sono gestiti tramite una **linked list**, i cui nodi sono ordinati in base al tempo di occorrenza degli eventi stessi. L'ordinamento è eseguito ad ogni inserimento di un nuovo evento nel metodo `insert_event_ordered()`.

⁵Si ricorda che in questa soluzione il tempo del sistema avanza in base agli eventi processati

Dopo essere stato prelevato dalla lista, un evento viene rimosso.
La struttura dati per l'implementazione degli eventi è la seguente:

```
typedef struct event_t {
    long event_id;
    block_type block_type;
    double time;
    event_type event_type;
    int target_server;
    struct event_t *next;
} event;
```

In particolare `event_id` identifica l'evento, `block_type` è il tipo di centro in cui si verifica l'evento, `time` è il tempo di occorrenza dell'evento, `event_type` è il tipo di evento (i.e. *ARRIVAL*, *IMMEDIATE_ARRIVAL*, *COMPLETION*), `target_server` il server in cui si verifica un *COMPLETION*, mentre `next` è il puntatore al prossimo evento in ordine cronologico crescente.

4.2.1 Arrivi & Completamenti

Sia gli arrivi dall'esterno che i completamenti sono generati tramite la funzione `Exponential()` presente dalla libreria `rvgs.h`.

```
double Exponential(double m)
{
    return (-m * log(1.0 - Random()));
}
```

La creazione dei valori per la distribuzione è basata sul **generatore di Lehmer**, fornito dalla libreria `rngs.h`. Per ciascuna tipologia di evento, e per ogni centro del sistema, viene usato uno **stream diverso**. In particolare è stato utilizzato lo stream 0 per gli arrivi e gli stream da 1 a 6 per i completamenti nei blocchi.

Per processare un evento d'arrivo, viene per prima cosa creato un nuovo evento della stessa tipologia. Inoltre, in ogni blocco, all'evento di arrivo segue la generazione del relativo evento di completamento, solo se il blocco ha almeno un servente libero. Se quest'ultima condizione non è soddisfatta, viene aumentano il numero di job in coda per quel centro⁶.

Ad ogni completamento corrisponde un arrivo nel centro successivo, in base alle probabilità di routing. Questi arrivi non hanno un ritardo nel passaggio da un blocco all'altro, per questo motivo definiti come *IMMEDIATE_ARRIVAL*. Dunque un evento di *IMMEDIATE_ARRIVAL* per un blocco corrisponde al completamento per un servente nel blocco antecedente.

E' importante osservare che la selezione del server, dove eseguire un completamento,⁷ avviene in modo pseudo-casuale utilizzando la funzione `Equillikely()`⁸ di `rvgs.c`.

La scelta della politica di scheduling **random** è giustificata dal fatto che quando un utente sceglie il posto a sedere, le preferenze sul posto variano a seconda dell'utente⁹.

⁶Ad esclusione del blocco *CONSUMAZIONE*, il quale non possiede una coda.

⁷Implementata nella funzione `retrieve_idle_server()`.

⁸Anch'essa basata sul generatore di Lehmer, in modo da garantire la randomicità.

⁹Alcuni possono preferire un posto vicino alle casse, altri vicino a colleghi e altri ancora in posti con meno persone.

In Alg. 1 e Alg. 2 sono riportati gli pseudo-codici rispettivamente per la gestione degli eventi di arrivo e completamento.

Algorithm 1 Pseudo-codice per la gestione un evento d'arrivo al tempo t nel centro i

```

1:  $l_i(t) ++$ 
2: if ( $\{\text{idle servers}\} \in i\} \neq \emptyset$  then
3:    $event \leftarrow \text{create\_completion\_event}()$ 
4:    $\text{insert\_event}(event)$ 
5:    $server\_service\_time += event\_completion\_time$ 
6:    $block\_service\_time += event\_completion\_time$ 
7:    $server\_served ++$ 
8: else if  $block\_type == CONSUMAZIONE$  then
9:    $l_i(t) --$ 
10:   $block\_rejected\_jobs ++$ 
11: else
12:    $q_i(t) ++$ 
13: end if
14: return

```

Algorithm 2 Pseudo-codice per la gestione di un evento di completamento al tempo t nel centro i

```

1:  $block\_completed\_jobs ++$ 
2:  $l_i(t) --$ 
3:  $\text{free\_busy\_server}()$ 
4: if  $block\_type == CONSUMAZIONE$  then
5:   return
6: end if
7: if  $q_i(t) > 0$  then
8:   if ( $\{\text{idle servers}\} \in i\} \neq \emptyset$  then
9:      $event \leftarrow \text{create\_completion\_event}()$ 
10:     $\text{insert\_event}(event)$ 
11:     $server\_service\_time += event\_completion\_time$ 
12:     $block\_service\_time += event\_completion\_time$ 
13:     $server\_served ++$ 
14:     $q_i(t) --$ 
15:   end if
16: end if
17:  $\text{schedule\_immediate\_arrival}()$ 
18: return

```

4.3 Statistiche

Le statistiche di output calcolate per ciascun blocco i sono:

- Time-averaged:

- \bar{l}_i , popolazione media;
 - \bar{q}_i , popolazione media in coda;
 - \bar{x}_i , popolazione media in servizio;
- Job-averaged:
 - \bar{r}_i , tempo medio di interarrivo;
 - \bar{w}_i , tempo medio di risposta;
 - \bar{d}_i , tempo medio di attesa;
 - \bar{s}_i , tempo medio di servizio;

Ad ogni processamento di evento le aree \bar{l}_i e \bar{q}_i , rappresentanti gli integrali delle statistiche time-averaged, vengono aggiornate nella procedura `update_area_stats()` come segue.

```
area = canteen->blocks[i]->block_area;
area->node += diff * (double) canteen->blocks[i]->jobs;
area->queue += diff * (double) canteen->blocks[i]->queue_jobs;
```

Per quanto riguarda il calcolo delle statistiche per ogni servente, a ciascuno viene associata la struttura dati `sum` i cui attributi rappresentano il tempo totale di servizio (i.e., `service`) e il numero di job completati (i.e., `served`).

```
typedef struct sum_t {
    long double service;
    long served;
} sum;
```

I valori di `sum` vengono aggiornati alla creazione di un nuovo evento di completamento insieme al valore del tempo totale di servizio per il blocco.

Il calcolo delle statistiche è effettuato nella funzione `get_stats()` del file `helpers.c`.

5 Verifica

L'obiettivo di questa fase è controllare la consistenza dei risultati prodotti, e in particolare confrontare i valori teorici del modello delle specifiche con i valori simulati. Nella discussione che segue, i dati ottenuti fanno riferimento ad una singola simulazione con durata pari al periodo di osservazione (i.e., 3 ore) con seed pari a 123456789.

Vedi Fig. 3 per maggiori dettagli.

```
'Primi' block info:
    people in the block ..... =      1941  people
job averaged statistics:
    average interarrival time =  5.56      s
    average node wait ..... =  66.87      s
    average queue delay ..... =  51.53      s
    average service time .... =  15.34      s

time averaged statistics:
    average # in the node ... =  12.02      people
    average # in the queue .. =  9.26      people
    utilization ..... =  0.9191      -

routing probabilities statistics:
    P(to SECONDO) ..... =  0.5466
    P(to CASSA FAST) .... =  0.2004
    P(to DESSERT) ..... =  0.2530

Multi-server statistics:
    server      utilization      avg service
        0          0.9140          15.05
        1          0.9258          15.75
        2          0.9177          15.25
```

(a) Centro *PRIMI*.

```
'Secondi e Contorni' block info:
    people in the block ..... =      1655  people
job averaged statistics:
    average interarrival time =  6.52      s
    average node wait ..... =  24.49      s
    average queue delay ..... =  9.57      s
    average service time .... =  14.92      s

time averaged statistics:
    average # in the node ... =  3.75      people
    average # in the queue .. =  1.47      people
    utilization ..... =  0.7622      -

routing probabilities statistics:
    P(to CASSA FAST) .... =  0.1438
    P1(to CASSA STD) .... =  0.4091
    P2(to DESSERT) ..... =  0.4471

Multi-server statistics:
    server      utilization      avg service
        0          0.7636          15.56
        1          0.7803          15.46
        2          0.7426          13.83
```

(b) Centro *SECONDI*.

```
'Frutta e Dessert' block info:
    people in the block ..... =      1231  people
job averaged statistics:
    average interarrival time =  8.77      s
    average node wait ..... =  13.39      s
    average queue delay ..... =  3.79      s
    average service time .... =  9.60      s

time averaged statistics:
    average # in the node ... =  1.53      people
    average # in the queue .. =  0.43      people
    utilization ..... =  0.5471      -

routing probabilities statistics:
    P(to CASSA STD) .... =  1.0000      s

Multi-server statistics:
    server      utilization      avg service
        0          0.5516          9.62
        1          0.5426          9.58
```

(c) Centro *DESSERT*.

```
'Casse Fast' block info:
    people in the block ..... =      627  people
job averaged statistics:
    average interarrival time =  17.22      s
    average node wait ..... =  30.77      s
    average queue delay ..... =  19.34      s
    average service time .... =  11.44      s

time averaged statistics:
    average # in the node ... =  1.79      people
    average # in the queue .. =  1.12      people
    utilization ..... =  0.6639      -

Multi-server statistics:
    server      utilization      avg service
        0          0.6639          11.44
```

(d) Centro *CASSA FAST*.

```
'Casse standard' block info:
    people in the block ..... =      1908  people
job averaged statistics:
    average interarrival time =  5.66      s
    average node wait ..... =  33.44      s
    average queue delay ..... =  15.12      s
    average service time .... =  18.32      s

time averaged statistics:
    average # in the node ... =  5.91      people
    average # in the queue .. =  2.67      people
    utilization ..... =  0.8093      -

Multi-server statistics:
    server      utilization      avg service
        0          0.8030          19.10
        1          0.7981          17.52
        2          0.8184          18.77
        3          0.8176          17.98
```

(e) Centro *CASSA STD*.

```
'Locale Mensa' block info:
    people in the block ..... =      2464  people
    rejected people ..... =       71  people
    loss probability ..... =  0.0280079  people

job averaged statistics:
    average interarrival time =  4.38      s
    average node wait ..... =  606.86      s
    average queue delay ..... =  0.00      s
    average service time .... =  606.86      s

time averaged statistics:
    average # in the node ... = 138.45      people
    average # in the queue .. =  0.00      people
    utilization ..... =  0.9230      -

Multi-server statistics:
    server      utilization      avg service
        0          0.8907          506.29
        1          0.8637          621.83
        2          0.8476          416.11
        3          0.8964          605.09
```

(f) centro *CONSUMAZIONE* (solo i primi 4 posti).

Figura 3: Statistiche di output per ogni centro.

5.1 Controlli di Consistenza

La prima verifica riguarda la consistenza dei risultati ottenuti per ogni blocco.

5.1.1 Tempi di risposta nei centri

Per il tempo medio di risposta in ciascun centro i deve valere:

$$E(T_S) = E(T_Q) + E(S)$$

La condizione è verificata per ogni blocco, come mostrato dai tempi simulati riportati in Tab. 5.1.1.

Centro	$E(T_Q)$	$E(S)$	$E(T_S)$
<i>PRIMO</i>	51.53	15.34	66.87
<i>SECONDO</i>	9.57	14.92	24.49
<i>DESSERT</i>	3.79	9.60	13.39
<i>CASSA FAST</i>	19.34	11.44	30.78
<i>CASSA STD</i>	15.12	18.32	33.44
<i>CONSUMAZIONE</i>	0	606.86	606.86

5.1.2 Popolazioni dei centri

Un ulteriore controllo di consistenza riguarda il numero di job nel centro pari a:

$$E(N_S) = E(N_Q) + m \cdot \rho$$

In Tab. 5.1.2 si può verificare che la formula è soddisfatta.

Centro	$E(N_Q)$	m	ρ	$E(N_S)$
<i>PRIMO</i>	9.26	3	0.9191	12.01729
<i>SECONDO</i>	1.47	3	0.7622	3.75662
<i>DESSERT</i>	0.43	2	0.5471	1.5242
<i>CASSA FAST</i>	1.12	1	0.6639	1.78389
<i>CASSA STD</i>	2.67	4	0.8093	5.90718
<i>CONSUMAZIONE</i>	0.0	150	0.9230	138.45062

5.2 Dati di Input

In accordo con quanto definito in Sec. 3.1, si vuole verificare che i dati computati dalla simulazione riprendano quelli dati in input al sistema.

5.2.1 Numero Utenti

Il primo controllo riguarda il numero totale di utenti entrato nel sistema, definito pari a 2500 in media nel modello delle specifiche, mentre per la simulazione vale 2535 ottenuto sommando il numero di arrivi per i centri *CASSA FAST* e *CASSA STD* (vedi Fig. 3d e Fig. 3e): $627 + 1908 = 2535$. Se inoltre si verifica il numero di utenti sommando gli utenti accettati e quelli rifiutati nel blocco *CONSUMAZIONE* si ottiene lo stesso risultato: $2464 + 71 = 2535$.

5.2.2 Tempi e Tassi di Servizio

Anche i tempi di servizio di ciascun servente riprendono quelli definiti nel modello delle specifiche. In particolare per i tre serventi del centro *PRIMO* (vedi Fig. 3a) i valori sono $E(S_{P_1}) = 15.05\text{ s}$, $E(S_{P_2}) = 15.75\text{ s}$, $E(S_{P_3}) = 15.25\text{ s}$, dunque vicini al valore di riferimento $E(S_P) = 15\text{ s}$. Inoltre, la media dei tempi restituisce il corrispettivo valore dell'intero blocco: $\frac{15.05+15.75+15.25}{3} = 15.34\text{ s}$.

In Tab. 5.2.2 sono confrontati i tassi di servizio simulati e quelli passati come input.

Centro	μ Teorico	μ Simulato
<i>PRIMO</i>	0.06667	0.06519
<i>SECONDO</i>	0.06667	0.06702
<i>DESSERT</i>	0.10000	0.10417
<i>CASSA FAST</i>	0.0909	0.08742
<i>CASSA STD</i>	0.05554	0.05458
<i>CONSUMAZIONE</i>	0.00166	0.00165

Come è possibile vedere, i tassi di servizio simulati sono piuttosto vicini ai valori teorici già con una simulazione di 3 ore.

5.2.3 Probabilità di Routing

Sono state verificate anche le probabilità di routing. Ad esempio, considerando il centro *PRIMO* (vedi Fig. 3a) la probabilità di routing dall'esterno al blocco *PRIMO* è facilmente calcolabile dalla simulazione come la frazione di job entranti nel blocco su quelli totali con la seguente formula:

$$p_{0,1} = \frac{\text{job uscenti da PRIMO}}{\text{job entranti da esterno}}$$

Il calcolo delle probabilità di routing è eseguito nella funzione `simulation_routing_probabilities_test()` del file `analytic_test.c`.

In Tab. 5.2.3 sono presentati i risultati. Anche in questo caso i valori simulati sono in linea con i valori teorici.

Sorgente i	Destinazione j	p_{ij} Teorica	p_{ij} Simulata
<i>ESTERNO</i>	<i>PRIMO</i>	0.75	0.7656
<i>ESTERNO</i>	<i>SECONDO</i>	0.25	0.2343
<i>PRIMO</i>	<i>SECONDO</i>	0.55	0.5466
<i>PRIMO</i>	<i>DESSERT</i>	0.25	0.2530
<i>PRIMO</i>	<i>CASSA FAST</i>	0.20	0.2004
<i>SECONDO</i>	<i>DESSERT</i>	0.45	0.4471
<i>SECONDO</i>	<i>CASSA FAST</i>	0.1375	0.1438
<i>SECONDO</i>	<i>CASSA STD</i>	0.4125	0.4091
<i>DESSERT</i>	<i>CASSA STD</i>	1.0	1.0000
<i>CASSA FAST</i>	<i>CONSUMAZIONE</i>	1.0	1.0000
<i>CASSA STD</i>	<i>CONSUMAZIONE</i>	1.0	1.0000

5.2.4 Tassi di Arrivo

I tassi di arrivo per i centri teorici e simulati (dagli inversi dei tempi di interarrivo) sono i visibili in Tab. 5.2.4. I tassi di arrivo simulati sono stati ottenuti dall'inverso del tempo di interarrivo medio ottenuto in output da ciascun blocco. Il tasso di arrivo del blocco *CONSUMAZIONE* considera solamente gli utenti accettati e non quelli rifiutati.

Centro	λ Teorico	λ Simulato
<i>PRIMO</i>	0.1735	0.17969
<i>SECONDO</i>	0.1534	0.15321
<i>DESSERT</i>	0.1124	0.11397
<i>CASSA FAST</i>	0.0558	0.05814
<i>CASSA STD</i>	0.1757	0.17664
<i>CONSUMAZIONE</i>	0.2256	0.22818

5.2.5 Visite medie

Abbiamo verificato le visite medie rispetto a quelle teoriche. Sappiamo che la formula per il calcolo delle visite medie in una rete aperta è la seguente:

$$v_i = \frac{\lambda_i}{\lambda}$$

In Tab. 5.2.5 è possibile osservare come i valori simulati delle visite sono perlopiù vicini a quelli teorici, già con una simulazione di sole 3 ore¹⁰. Per i dettagli sull'implementazione, si rimanda alla funzione `simulation_visit_test()` del file `analytic_test.c`.

Centro	v_i Teorico	v_i Simulato
<i>PRIMO</i>	0.75	0.749801
<i>SECONDO</i>	0.6625	0.666005
<i>DESSERT</i>	0.4856	0.483717
<i>CASSA FAST</i>	0.2411	0.236299
<i>CASSA STD</i>	0.7589	0.763701
<i>CONSUMAZIONE</i>	0.9748	0.976966

5.2.6 Utilizzazioni

In Tab. 5.2.6 è possibile osservare come i valori simulati si avvicinano ai valori teorici.

Centro	ρ Teorica	ρ Simulata
<i>PRIMO</i>	0.86746	0.9191
<i>SECONDO</i>	0.76698	0.7622
<i>DESSERT</i>	0.56197	0.5471
<i>CASSA FAST</i>	0.61388	0.6639
<i>CASSA STD</i>	0.79086	0.8093
<i>CONSUMAZIONE</i>	0.90427	0.9230

¹⁰Le visite (e i tassi di arrivo) per il blocco *CONSUMAZIONE* sono calcolate con usando $\lambda' = \lambda \cdot (1 - p_{loss})$.

Se si aumenta il numero di ore di simulazione, si può verificare che le utilizzazioni simulate tendono alle simulazioni teoriche. Per i dettagli sull'implementazione, si faccia riferimento alla funzione `get_stats()` nel file `helpers.c`.

5.3 QoS

Vengono ora confrontati i QoS della simulazione con quelli calcolati usando le formule teoriche e i dati nel modello delle specifiche.

5.3.1 Tempo di Risposta Globale

E' necessario verificare anche il tempo di risposta globale, che si ricorda essere un QoS.

Sia v_i il numero di visite medie nel centro i nella nostra rete aperta ed $E(T_{r,i})$ il tempo di risposta del centro i . Il tempo di risposta globale teorico della rete è calcolato come segue:

$$E(T_r^{(t)}) = \sum_{i=1}^M E(T_{S,i})^{(t)} \cdot v_i^{(t)} = 673.21152 \text{ s}$$

Il risultato simulato è invece:

$$E(T_r^{(s)}) = \sum_{i=1}^M E(T_{S,i})^{(s)} \cdot v_i^{(s)} = 696.314117 \text{ s}$$

I due risultati non coincidono ma, come vedremo nella simulazione a orizzonte infinito, all'aumentare del tempo di simulazione il tempo di risposta globale simulato tende a quello teorico.

5.3.2 Probabilità di Perdita

Il secondo QoS riguarda il solo centro *CONSUMAZIONE*. Con 150 posti a sedere la probabilità di perdita è la seguente:

$$p_{loss}^{(t)} = \frac{\frac{1}{m!} \cdot \left(\frac{\lambda}{\mu}\right)^m}{\sum_{j=0}^m \frac{1}{j!} \cdot \left(\frac{\lambda}{\mu}\right)^j} = 0.0251$$

La probabilità simulata è:

$$p_{loss}^{(s)} = 0.0280$$

Si ottiene dunque $p_{loss}^{(s)} > p_{loss}^{(t)}$, ma a fronte di numero di ore per la simulazione molto basso. Nel seguito vedremo che questo valore si avvicinerà di più a quello teorico.

6 Validazione

Il sistema in analisi è ipotetico, dunque non prende uno specifico modello reale come riferimento. Non è dunque possibile validarla confrontando i risultati del modello computazionale con i risultati del sistema reale.

Il processo di validazione si riconduce quindi a studiare il comportamento del modello con piccole variazioni al tasso di arrivo.

6.1 Aumento del Tasso di Arrivo

Aumentando il tasso di arrivo a $\lambda = 0.25 \frac{\text{arrivi}}{\text{secondo}}$ si osserva una maggiore probabilità di perdita, ma una diminuzione del tempo di risposta.

I valori sono riportati in Tab. 6.1.1 e 6.1.2¹¹.

6.1.1 Tempo di Risposta Globale

Statistica	Risultato
$E(T_r)^{\text{teorico}}$	695.6529 s
$E(T_r)^{\text{transiente}}$	$695.4047 \text{ s} \pm 3.404221 \text{ s}$
$E(T_r)^{\text{stazionario}}$	$696.260618 \text{ s} \pm 2.274880 \text{ s}$

6.1.2 Probabilità di Perdita

Statistica	Risultato
$p_{\text{loss}}^{\text{teorico}}$	0.062403
$p_{\text{loss}}^{\text{transiente}}$	0.052404 ± 0.001833
$p_{\text{loss}}^{\text{stazionario}}$	0.062450 ± 0.001089

6.2 Diminuzione del Tasso di Arrivo

Con una diminuzione del tasso di arrivo a $\lambda = 0.2 \frac{\text{arrivi}}{\text{secondo}}$ si osservano valori più piccoli sia per la probabilità di perdita che nel tempo di risposta.

I valori simulati sono riportati in Tab. 6.2.1 e 6.2.2.

6.2.1 Tempo di Risposta Globale

Statistica	Risultato
$E(T_r)^{\text{teorico}}$	663.4007 s
$E(T_r)^{\text{transiente}}$	$664.6840 \text{ s} \pm 1.5800 \text{ s}$
$E(T_r)^{\text{stazionario}}$	$662.6915 \text{ s} \pm 0.8893 \text{ s}$

6.2.2 Probabilità di Perdita

Statistica	Risultato
$p_{\text{loss}}^{\text{teorico}}$	0.0010
$p_{\text{loss}}^{\text{transiente}}$	0.0008 ± 0.0002
$p_{\text{loss}}^{\text{stazionario}}$	0.0009 ± 0.0001

7 Processo degli esperimenti

In questo capitolo sarà presentato e discusso lo studio delle statistiche transienti e stazionarie ottenute nelle simulazioni ad orizzonte temporale finito e infinito.

¹¹Nel transiente si fa riferimento a 250 repliche.

7.1 Simulazione Finite-Horizon

Nella simulazione ad orizzonte temporale finito si è studiato l'andamento del tempo globale di risposta e la probabilità di perdita **transienti**, sia al variare del numero di repliche che in funzione del seed.

7.1.1 Variazione del Seed - Simulazione Singola

Prima di vedere l'andamento delle statistiche di interesse con la replicazione, è interessante vedere l'andamento dei risultati prodotti da **singole simulazioni** di diversa durata, facendo variare il seed del sistema, per verificare il suo comportamento.

In Fig. 4 e Fig. 5 è mostrato come per all'**aumentare del periodo di osservazione** i risultati ottenuti hanno un andamento asintotico, ma distante dal valore teorico; inoltre ma c'è una grande varianza tra gli andamenti dei diversi seed. In particolare, in Fig. 4 i tempi tendono ad avvicinarsi al tempo di risposta teorico, mentre in Fig. 5 la differenza con il valore di riferimento è maggiore, ma anche la scala del grafico risulta essere a grana molto fine. E' quindi necessario procedere con la simulazione finite-horizon basata su replicazione, per ridurre la varianza di questi risultati transienti.

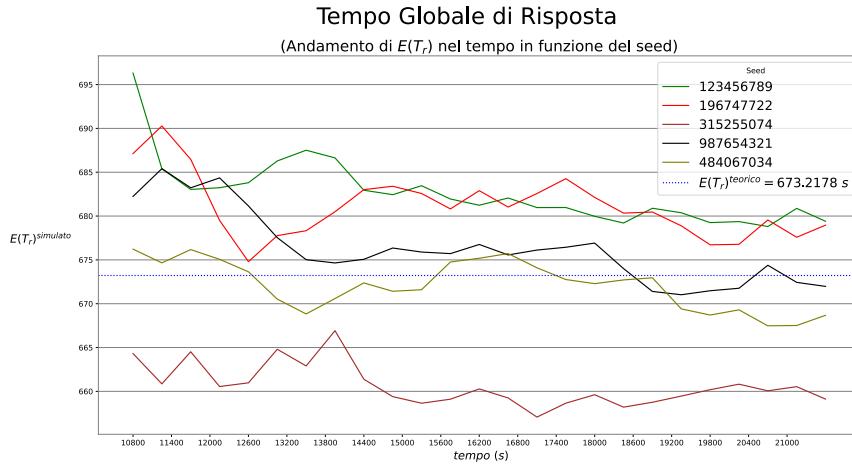


Figura 4: Andamento del tempo globale di risposta al variare del seed di sistema.

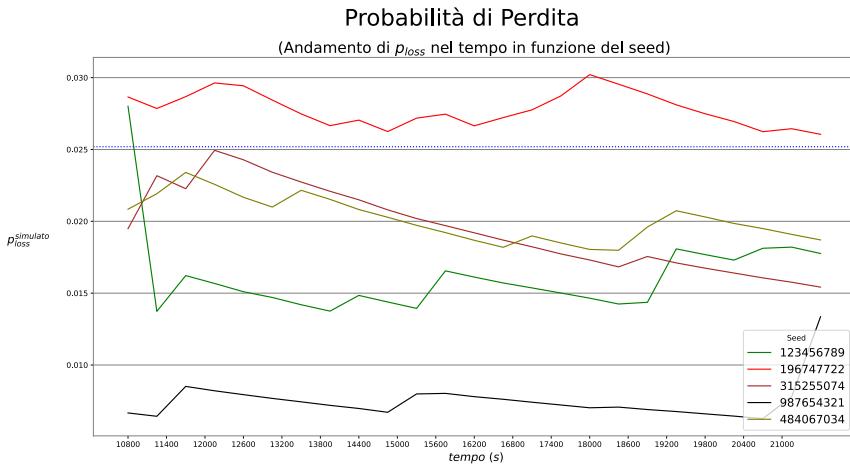


Figura 5: Andamento della probabilità di perdita al variare del seed di sistema.

7.1.2 Replicazione

In questa analisi si è variato il numero di repliche per ciascun **ensemble** partendo da un insieme di 50 fino a 500, con incrementi di 50 repliche ad ogni iterazione.

Gli stream sono inizializzati una sola volta invocando `PlantSeed()` prima del ciclo di replicazione, mentre tra una simulazione e la successiva si riutilizza lo stato finale di ogni stream. In questo modo è possibile generare **stime indipendenti** della stessa statistica **transiente**.

La condizione di terminazione è sempre il tempo *close-the-doors*, ovvero una durata complessiva di 3 ore, e inoltre il numero di job in ogni blocco all'inizio e alla fine è zero.

Per ogni ensemble vengono calcolate, con l'algoritmo **one-pass di Wellford**, **media** e **deviazione standard** e quindi l'intervallo di confidenza al 95 %. Il calcolo degli intervalli è stato implementato sulla base del file `estimate.c`.

Con i risultati ottenuti si vuole dimostrare come all'aumentare del numero di simulazioni, i valori medi tendono a un certo valore diverso da quello teorico, che rappresenta la media della statistica transiente. Inoltre, è possibile notare che all'aumentare delle replicazioni, l'intervallo di confidenza ha un'ampiezza sempre più piccola.

Nelle Fig. 6 e Fig. 7 si può notare come i valori transienti si discostino dai valori teorici stazionari. Poiché il tempo di simulazione di 3 ore è troppo basso per portare a regime il blocco *CONSUMAZIONE*, c'è una forte **dipendenza dalle condizioni iniziali**, che rende il tempo di risposta maggiore (perché molti trovano posto e rimangono a mangiare) e la probabilità di perdita minore (perché la mensa inizialmente è vuota). Questo è di **grande interesse** per l'analisi in quanto questa situazione si presenterebbe ogni giorno in un ipotetico sistema reale organizzato secondo questo modello. Con 10.000 repliche, il tempo di risposta transiente è 674.642288 ± 0.241640 s (superiore di 1.5 s rispetto allo stazionario), mentre la probabilità di perdita è 0.020559 ± 0.000211 (inferiore di 0.5 punti percentuali rispetto alla probabilità di perdita stazionario).

Quindi è di maggiore interesse la simulazione a orizzonte finito rispetto a quella a orizzonte infinito, perché permette di avere una visione del modello molto più vicina a quella reale.

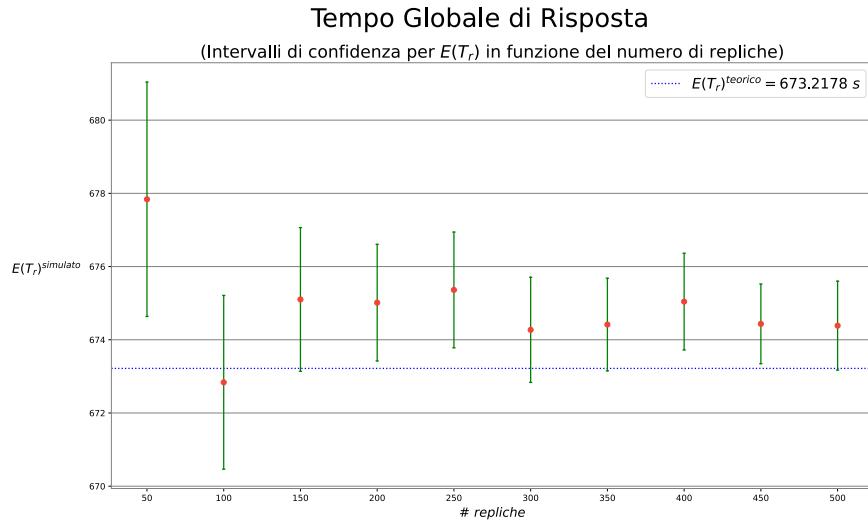


Figura 6: Andamento del tempo globale di risposta all'aumentare del numero di repliche.

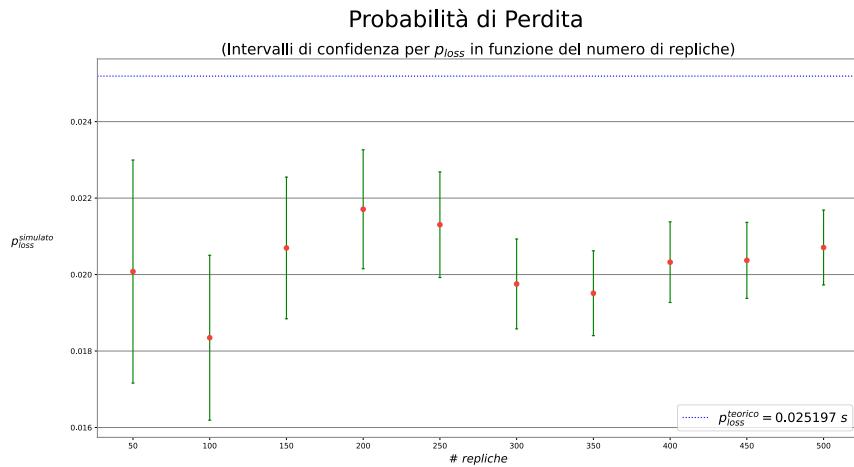


Figura 7: Andamento della probabilità di perdita all'aumentare del numero di repliche.

7.2 Simulazione Infinite-Horizon

L'obiettivo è studiare l'andamento del sistema in un tempo molto più grande di quello operativo (i.e., 3 ore).

7.2.1 Batch Means

Il primo passo nell'utilizzo dell'algoritmo batch means è la scelta dei parametri b (dimensione del batch) ed n (numero di job), per poter così ricavare il numero di job appartenenti ad ogni batch. Per prima cosa è stato fissato il numero di utenti totali a un numero molto grande:

$$n = 2.5 \cdot 10^6$$

Dopodiché, per scegliere il numero di batch tenendo fisso il numero di utenti, sono state seguite le linee guida del libro¹². In particolare il criterio considerato per la scelta del numero di batch è basato sul calcolo dell'**autocorrelazione**¹³, che deve essere inferiore a 0.2 in corrispondenza di un lag pari a 1. Per il calcolo delle autocorrelazioni ci si è basati sul programma **acs.c**, che è stato integrato nel codice della simulazione batch means. Quindi, una volta implementato il codice relativo alla simulazione a orizzonte infinito, sono stati fatti diversi tentativi nella scelta di b , finché l'autocorrelazione per tempo di risposta e probabilità di perdita fosse minore di 0.2. I parametri adottati sono dunque:

$$b = 2000$$

$$k = \left\lfloor \frac{n}{b} \right\rfloor = \left\lfloor \frac{2.5 \cdot 10^6}{2 \cdot 10^3} \right\rfloor = 1250$$

In Fig. 8a e Fig. 8b sono riportati i valori delle autocorrelazioni.

Dopodiché viene eseguita una lunga simulazione di n job, in cui ad ogni b job vengono calcolati i tempi di risposta e le probabilità di perdita del batch. A questo punto lo stato del sistema (numero di job in coda e job nel centro) viene lasciato invariato¹⁴, si azzerano le sole statistiche di output (e.g., tempi di risposta, aree, job completati) e in questo modo è possibile calcolare i tempi di risposta e le probabilità di perdita solo in base ai valori del batch, e non dall'inizio, a differenza di come fatto per la simulazione a orizzonte finito. Grazie a questo accorgimento le medie dei batch sono indipendenti tra loro.

Al termine della simulazione a orizzonte finito, vengono calcolate le medie e le deviazioni standard delle medie dei batch per tempo di risposta e probabilità di perdita, così da costruire un intervallo di confidenza.

¹²Leemis L. M. & Park S. K. (2006). *Discrete-event simulation : a first course*. Pearson Prentice Hall.

¹³Banks, Carson, Nelson, and Nicol (2001, page 483).

¹⁴In quanto nella simulazione a orizzonte infinito non è di interesse che il sistema torni allo stato iniziale.

```

for 1250 data points
the mean is ... 673.28
the stdev is .. 15.70

j (lag) r[j] (autocorrelation)
1 -0.204

```

(a) Tempo di risposta globale.

```

for 1250 data points
the mean is ... 0.02
the stdev is .. 0.01

j (lag) r[j] (autocorrelation)
1 0.012

```

(b) Probabilità di perdita.

Figura 8: Autocorrelazione per il tempo medio di risposta globale e la probabilità di perdita.

In Fig. 9 è possibile osservare come si raggiunga la stazionarietà del tempo di risposta globale già al 300esimo batch, che corrisponde a 600.000 arrivi nella mensa.

Per quanto riguarda la probabilità di perdita (vedi Fig. 10) i valori raggiungono la stazionarietà all'incirca dall'800esimo batch. In questo caso, la maggiore distanza tra valore teorico e simulato è dovuta ad una scala più grande dei valori rappresentati.

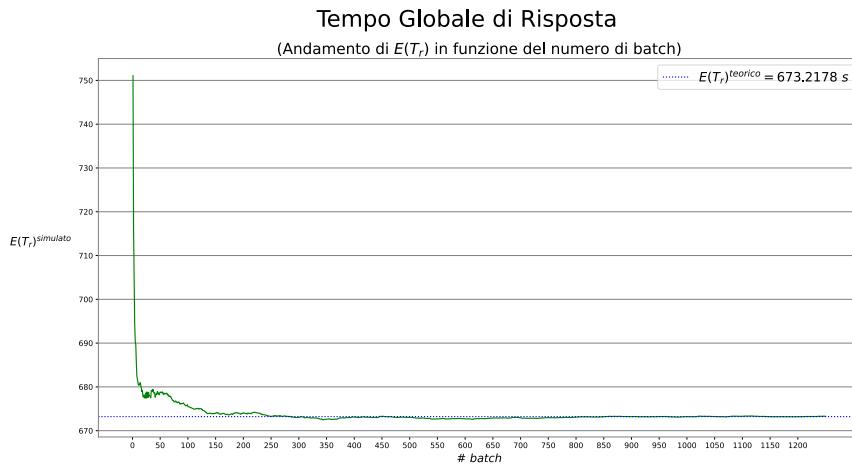


Figura 9: Andamento del tempo globale di risposta medio all'aumentare del numero di batch.

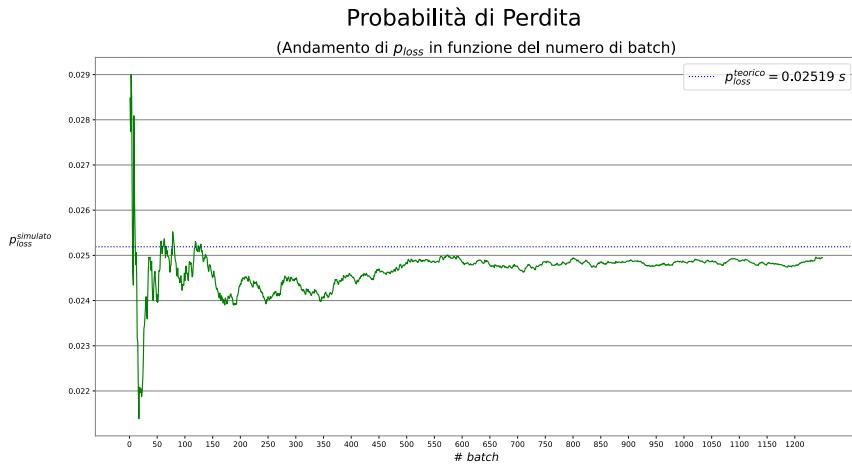


Figura 10: Andamento della probabilità di perdita media all'aumentare del numero di batch.

7.2.2 Consistenza sul Batch Means

Per controllare la correttezza dei risultati ottenuti si è confrontata la media risultante dai batch con quella ottenuta da un'unica simulazione della stessa durata. In Tab. 7.2.2 si può osservare quanto i risultati siano vicini tra loro.

$E(T_r)^{\text{batch}}$	$E(T_r)^{\text{singolo run}}$	p_{loss}^{batch}	$p_{loss}^{\text{singolo run}}$
673.278713 s	673.009476 s	0.024949	0.024958

7.2.3 Variazione del Seed

Come nella simulazione finita, si è voluto studiare un'eventuale variazione dei risultati tra seed diversi nel batch means.

In Fig. 11 e Fig. 12 si nota come le medie siano molto simili tra loro e tutte prossime ai valori teorici sia per il tempo di risposta che per la probabilità di perdita.

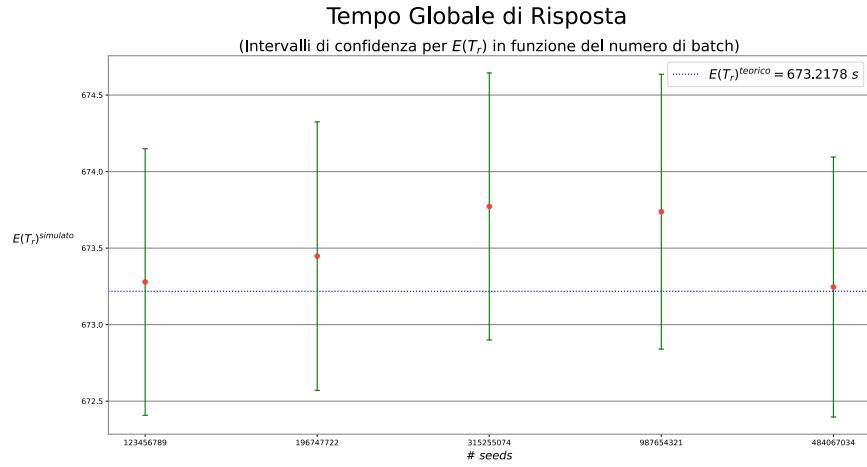


Figura 11: Intervalli di confidenza per il tempo globale di risposta al variare del seed di sistema.

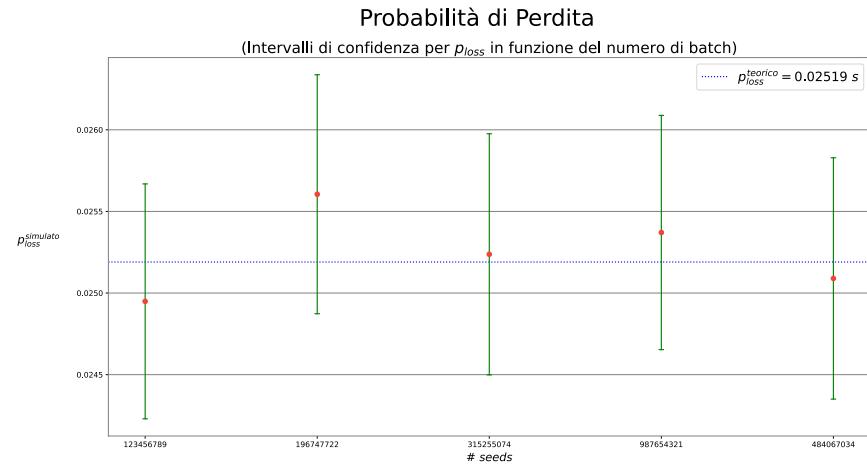


Figura 12: Intervalli di confidenza per la probabilità di perdita al variare del seed di sistema.

8 Modello Migliorativo

Alla luce dei risultati conseguiti nel modello presentato fino ad ora, in questo capitolo vengono proposte tre varianti della zona *CONSUMAZIONE* con l'obiettivo di annullare il numero di dipendenti che non trovano un posto a sedere.

La prima miglioria considerata è di aumentare il numero di posti a sedere passando dai 150 iniziali a 200. In Fig. 13 è riportato il modello concettuale aggiornato¹⁵.

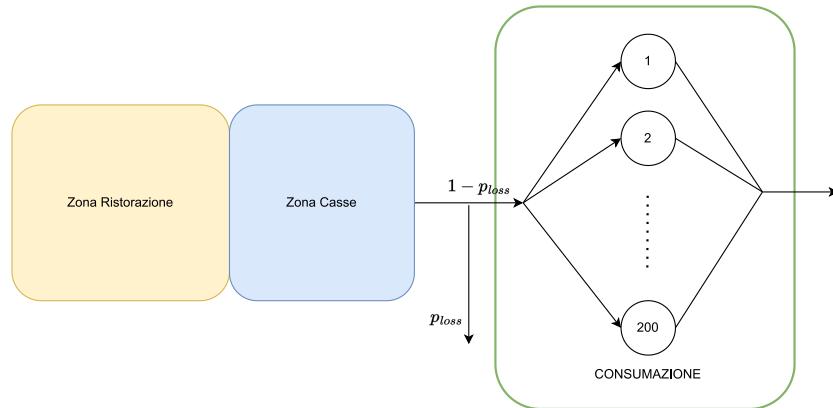


Figura 13: Modello concettuale migliorativo con aumento dei posti a sedere.

Ne segue che il **modello delle specifiche** e il **modello computazionale** è pressocché invariato rispetto a quanto discusso in Sec. 3 e Sec. 4.

8.1 Verifica

I dati ottenuti fanno riferimento ad una singola simulazione con durata pari al periodo di osservazione (i.e., 3 ore) e seed pari a 123456789 (vedi Fig. 14).

¹⁵Le zone *Ristorazione* e *Casse* sono invariate rispetto al modello base.

```
'Primi' block info:
    people in the block ..... =      1913  people

job averaged statistics:
    average interarrival time =  5.64      s
    average node wait ..... =  37.07      s
    average queue delay ..... = 22.15      s
    average service time .... = 14.92      s

time averaged statistics:
    average # in the node ... =   6.57  people
    average # in the queue .. =   3.92  people
    utilization ..... = 0.8812      -

routing probabilities statistics:
P(to SECONDO) ..... = 0.5358
P(to CASSA FAST) .... = 0.1986
P(to DESSERT) ..... = 0.2656

Multi-server statistics:
    server      utilization      avg service
        0          0.8793           15.49
        1          0.8875           15.34
        2          0.8767           14.03
```

(a) Centro PRIMI.

```
'Secondi e Contorni' block info:
    people in the block ..... =      1713  people

job averaged statistics:
    average interarrival time =  6.30      s
    average node wait ..... =  24.08      s
    average queue delay ..... =  9.74      s
    average service time .... = 14.34      s

time averaged statistics:
    average # in the node ... =   3.82  people
    average # in the queue .. =   1.54  people
    utilization ..... = 0.7583      -

routing probabilities statistics:
P(to CASSA FAST) .... = 0.1331
P1(to CASSA STD) .... = 0.4051
P2(to DESSERT) ..... = 0.4618

Multi-server statistics:
    server      utilization      avg service
        0          0.7505           15.35
        1          0.7688           14.44
        2          0.7557           13.38
```

(b) Centro SECONDI.

```
'Frutta e Dessert' block info:
    people in the block ..... =      1299  people

job averaged statistics:
    average interarrival time =  8.31      s
    average node wait ..... =  14.37      s
    average queue delay ..... =  4.79      s
    average service time .... =  9.57      s

time averaged statistics:
    average # in the node ... =   1.73  people
    average # in the queue .. =   0.58  people
    utilization ..... = 0.5758      -

routing probabilities statistics:
P(to CASSA STD) .... = 1.0000  s

Multi-server statistics:
    server      utilization      avg service
        0          0.5632           9.40
        1          0.5885           9.75
```

(c) Centro DESSERT.

```
'Casse Fast' block info:
    people in the block ..... =      608  people

job averaged statistics:
    average interarrival time = 17.75      s
    average node wait ..... = 34.78      s
    average queue delay ..... = 23.11      s
    average service time .... = 11.66      s

time averaged statistics:
    average # in the node ... =   1.96  people
    average # in the queue .. =   1.30  people
    utilization ..... = 0.6567      -

Multi-server statistics:
    server      utilization      avg service
        0          0.6567           11.66
```

(d) Centro CASSA FAST.

```
'Casse standard' block info:
    people in the block ..... =      1993  people

job averaged statistics:
    average interarrival time =  5.42      s
    average node wait ..... = 29.28      s
    average queue delay ..... = 11.63      s
    average service time .... = 17.65      s

time averaged statistics:
    average # in the node ... =   5.40  people
    average # in the queue .. =   2.15  people
    utilization ..... = 0.8142      -

Multi-server statistics:
    server      utilization      avg service
        0          0.7998           18.07
        1          0.8116           16.51
        2          0.8131           17.70
        3          0.8324           18.42
```

(e) Centro CASSA STD.

```
'Locale Mensa' block info:
    people in the block ..... =      2601  people
    rejected people ..... =      0  people
    loss probability ..... =      0  people

job averaged statistics:
    average interarrival time =  4.15      s
    average node wait ..... = 608.62      s
    average queue delay ..... =  0.00      s
    average service time .... = 608.62      s

time averaged statistics:
    average # in the node ... = 146.58      people
    average # in the queue .. =   0.00  people
    utilization ..... = 0.7329      -

Multi-server statistics:
    server      utilization      avg service
        0          0.8245           593.67
        1          0.7333           879.99
        2          0.6995           472.15
        3          0.7896           609.15
```

(f) centro CONSUMAZIONE (solo i primi 4 posti).

Figura 14: Statistiche di output per ogni centro.

8.1.1 Controlli di Consistenza

Per i **tempi di risposta** nei centri le condizioni continuano ad essere verificate per ogni blocco, come mostrato in tabella:

Centro	$E(T_Q)$	$E(S)$	$E(T_r)$
<i>PRIMO</i>	22.15	14.92	37.07
<i>SECONDO</i>	9.74	14.34	24.08
<i>DESSERT</i>	4.79	9.57	14.37
<i>CASSA FAST</i>	23.11	11.66	34.78
<i>CASSA STD</i>	11.63	17.65	29.28
<i>CONSUMAZIONE</i>	0	608.62	608.62

Così come le **popolazioni** dei centri:

Centro	$E(N_Q)$	m	ρ	$E(N_S)$
<i>PRIMO</i>	3.92	3	0.8812	6.57
<i>SECONDO</i>	1.54	3	0.7583	3.82
<i>DESSERT</i>	0.58	2	0.5758	1.73
<i>CASSA FAST</i>	1.30	1	0.6567	1.96
<i>CASSA STD</i>	2.15	4	0.8142	5.40
<i>CONSUMAZIONE</i>	0	200	0.7329	146.58

8.2 Dati di Input

Il **numero di utenti** nel sistema con 200 posti in *CONSUMAZIONE* vale 2601 ottenuto sommando il numero di arrivi nelle casse (vedi Fig. 14d e Fig. 14e) oppure nel blocco consumazione:

$$E(N_F) + E(N_C) = 608 + 1993 = 2601 = E(N_S)$$

Considerando i **tempi di servizio** dei soli tre serventi del centro *PRIMO* (vedi Fig. 14a) si ottiene $E(S_1) = 15.49 \text{ s}$, $E(S_2) = 15.34 \text{ s}$, $E(S_3) = 14.03 \text{ s}$, dunque vicini al valore di riferimento $E(S) = 15 \text{ s}$. Inoltre, la media dei tempi restituisce il corrispettivo valore dell'intero blocco: $\frac{15.49+15.34+14.03}{3} = 14.92 \text{ s}$. Nella tabella che segue sono confrontati i **tassi di servizio** simulati e quelli passati come input.

Centro	μ Teorico	μ Simulato
<i>PRIMO</i>	0.06667	0.06702
<i>SECONDO</i>	0.06667	0.069735
<i>DESSERT</i>	0.10000	0.104493
<i>CASSA FAST</i>	0.0909	0.085763
<i>CASSA STD</i>	0.05554	0.056657
<i>CONSUMAZIONE</i>	0.00166	0.001643

I risultati per le **probabilità di routing** sono:

Sorgente i	Destinazione j	p_{ij} Teorica	p_{ij} Simulata
ESTERNO	PRIMO	0.7500	0.735486
ESTERNO	SECONDO	0.2500	0.264514
PRIMO	SECONDO	0.5500	0.535808
PRIMO	DESSERT	0.2500	0.265551
PRIMO	CASSA FAST	0.2000	0.198641
SECONDO	DESSERT	0.4500	0.461763
SECONDO	CASSA FAST	0.1375	0.1331
SECONDO	CASSA STD	0.4125	0.405137
DESSERT	CASSA STD	1.0000	1.000000
CASSA FAST	CONSUMAZIONE	1.0000	1.0000
CASSA STD	CONSUMAZIONE	1.0000	1.0000

I **tassi di arrivo** simulati e teorici sono calcolati con lo stesso sistema presentato in Sec. 5.2.4 e pari a:

Centro	λ Teorico	λ Simulato
PRIMO	0.1735	0.1770
SECONDO	0.1534	0.1585
DESSERT	0.1124	0.1202
CASSA FAST	0.0558	0.0562
CASSA STD	0.1757	0.1844
CONSUMAZIONE	0.2315	0.2404

Mentre per le **visite medie** si ha:

Centro	v_i Teorico	v_i Simulato
PRIMO	0.7500	0.735486
SECONDO	0.6625	0.658208
DESSERT	0.4856	0.499423
CASSA FAST	0.2410	0.233756
CASSA STD	0.7589	0.766244
CONSUMAZIONE	1.0000	1.0000

Le **utilizzazioni** ottenute sono:

Centro	ρ Teorica	ρ Simulata
PRIMO	0.8677	0.8812
SECONDO	0.7670	0.7583
DESSERT	0.5620	0.5758
CASSA FAST	0.6138	0.6567
CASSA STD	0.7906	0.8142
CONSUMAZIONE	0.6945	0.7329

8.3 QoS

Per il **tempo di risposta globale** si ha:

$$E(T_r^{(t)}) = \sum_{i=1}^M E(T_{S,i}) \cdot v_i$$

$$\begin{aligned}
&= E(T_{S,1}) \cdot v_1 + E(T_{S,2}) \cdot v_2 + E(T_{S,3}) \cdot v_3 + E(T_{S,F}) \cdot v_F + E(T_{S,C}) \cdot v_C + E(T_{S,S}) \cdot v_S \\
&= 43.8488 \cdot 0.75 + 27.7345 \cdot 0.6625 + 14.6181 \cdot 0.4856 + 28.4897 \cdot 0.2410 + 30.4488 \cdot 0.7589 + 600 \cdot 1 \\
&= 688.3362 \text{ s}
\end{aligned}$$

Per il simulato:

$$E(T_r^{(s)}) = 689.4784 \text{ s}$$

Infine, con 200 posti a sedere la **probabilità di perdita** è:

$$p_{loss}^{(t)} \simeq p_{loss}^{(s)} = 0$$

9 Modello Migliorativo 2 con Politica Random

La seconda proposta per il blocco *CONSUMAZIONE* considera la possibilità di avere due zone separate dove poter pranzare, ma di dimensione ridotta rispetto al caso precedente. Quest'idea nasce come alternativa all'ampliamento dello spazio già presente. Dunque si usa una zona aggiuntiva, separata da quella iniziale. In Fig. 15 è presentato il modello concettuale con 100 posti per ciascuna zona.

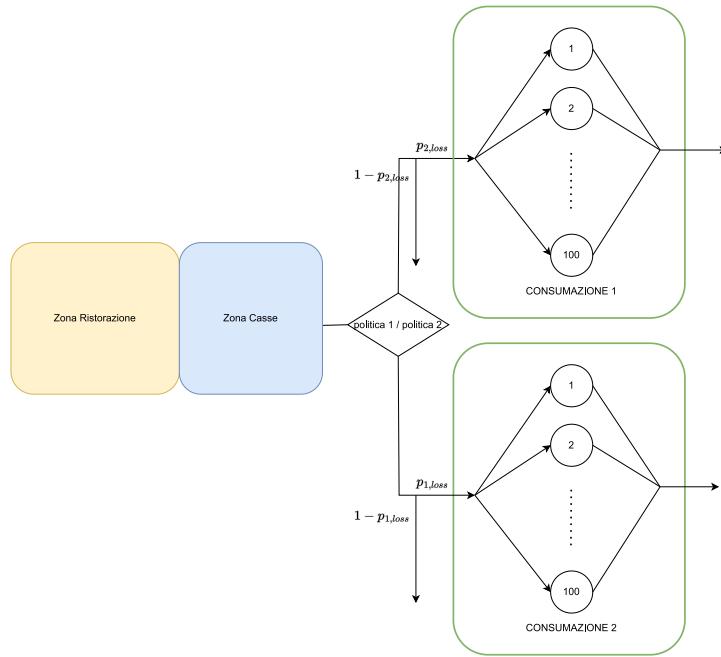


Figura 15: Modello concettuale migliorativo con aggiunta del blocco *CONSUMAZIONE 2*.

Le politiche studiate per la scelta del blocco dove mangiare sono due:

1. *Random*, la scelta è casuale;
2. *Least busy*, la scelta ricade sul blocco meno utilizzato;

9.1 Modello delle specifiche

La definizione del **modello delle specifiche** è pressoché invariata rispetto a quanto discusso nelle Sec. 3, con le seguenti differenze nei blocchi *CONSUMAZIONE 1* e *CONSUMAZIONE 2*:

- La probabilità di routing verso i blocchi consumazione è 0.5 invece di 1.0, sia per la politica *random* che per la politica *least busy*.
- Il tasso di arrivo teorico per ciascuno di questi blocchi è dimezzato e pari a $0.115741 \frac{\text{arrivi}}{\text{secondo}}$.
- L'utilizzazione teorica è ridotta a 0.694349.
- Come per il modello migliorativo precedente, la probabilità di perdita è ridotta a quasi 0, grazie al numero aumentato di posti a sedere.
- Il tempo di risposta teorico aumenta a 688.271262 s, in quanto si ha perdita quasi nulla.

9.2 Modello Computazionale

Oltre alle strutture dati descritte in Sec. 4 è stato necessario definire due macro per l'aggiunta del blocco *CONSUMAZIONE 2* e della politica di scelta.

```
#define EXTENDED
#define CHOOSE_LEAST_BUSY
```

Definendo queste due macro è stato possibile passare dal modello base al modello migliorativo semplicemente andando a modificare il codice nei punti in cui è coinvolto il blocco *CONSUMAZIONE*. Se la macro **EXTENDED** non viene definita, le simulazioni tornano ad essere quelle relative al modello base. Con la macro **CHOOSE_LEAST_BUSY** è possibile modificare la politica di scelta tra i due blocchi *CONSUMAZIONE*. Se definita, si considera l'utilizzazione come parametro di scelta, altrimenti si sceglie casualmente. Naturalmente questa macro non ha alcun effetto se viene definita per il modello base.

9.3 Verifica

I dati ottenuti fanno riferimento ad una singola simulazione con durata pari al periodo di osservazione (i.e., 3 ore) con seed pari a 123456789 (vedi Fig. 16 e Fig. 17).

```

'Primi' block info:
    people in the block ..... =      1944  people

job averaged statistics:
    average interarrival time =   5.55      s
    average node wait ..... =   41.20      s
    average queue delay ..... =   26.47      s
    average service time .... =   14.73      s

time averaged statistics:
    average # in the node ... =   7.42      people
    average # in the queue .. =   4.76      people
    utilization ..... = 0.8838      -

routing probabilities statistics:
    P(to SECONDO) ..... = 0.5468
    P(to CASSA FAST) .... = 0.2155
    P(to DESSERT) ..... = 0.2377

Multi-server statistics:
    server      utilization      avg service
      0          0.8878           14.51
      1          0.8858           14.41
      2          0.8779           15.32

```

(a) Centro PRIMI.

```

'Secondi e Contorni' block info:
    people in the block ..... =      1665  people

job averaged statistics:
    average interarrival time =   6.49      s
    average node wait ..... =   26.38      s
    average queue delay ..... =   11.39      s
    average service time .... =   14.99      s

time averaged statistics:
    average # in the node ... =   4.07      people
    average # in the queue .. =   1.76      people
    utilization ..... = 0.7706      -

routing probabilities statistics:
    P(to CASSA FAST) .... = 0.1351
    P1(to CASSA STD) .... = 0.4048
    P2(to DESSERT) ..... = 0.4601

Multi-server statistics:
    server      utilization      avg service
      0          0.7703           15.38
      1          0.7549           14.51
      2          0.7865           15.11

```

(b) Centro SECONDI.

```

'Frutta e Dessert' block info:
    people in the block ..... =      1228  people

job averaged statistics:
    average interarrival time =   8.79      s
    average node wait ..... =   13.62      s
    average queue delay ..... =   4.16      s
    average service time .... =   9.46      s

time averaged statistics:
    average # in the node ... =   1.55      people
    average # in the queue .. =   0.47      people
    utilization ..... = 0.5379      -

routing probabilities statistics:
    P(to CASSA STD) .... = 1.0000      s

Multi-server statistics:
    server      utilization      avg service
      0          0.5350           9.94
      1          0.5408           9.03

```

(c) Centro DESSERT.

```

'Casse Fast' block info:
    people in the block ..... =      644  people

job averaged statistics:
    average interarrival time =  16.77      s
    average node wait ..... =  33.97      s
    average queue delay ..... =  22.66      s
    average service time .... =  11.30      s

time averaged statistics:
    average # in the node ... =   2.03      people
    average # in the queue .. =   1.35      people
    utilization ..... = 0.6739      -

routing probabilities statistics:
    P(to CONSUMAZIONE 1) ..... = 0.5124
    P(to CONSUMAZIONE 2) ..... = 0.4876

Multi-server statistics:
    server      utilization      avg service
      0          0.6739           11.30

```

(d) Centro CASSA FAST.

```

'Casse standard' block info:
    people in the block ..... =      1902  people

job averaged statistics:
    average interarrival time =   5.68      s
    average node wait ..... =   28.75      s
    average queue delay ..... =   10.47      s
    average service time .... =   18.28      s

time averaged statistics:
    average # in the node ... =   5.06      people
    average # in the queue .. =   1.84      people
    utilization ..... = 0.8049      -

routing probabilities statistics:
    P(to CONSUMAZIONE 1) ..... = 0.5110
    P(to CONSUMAZIONE 2) ..... = 0.4890

Multi-server statistics:
    server      utilization      avg service
      0          0.8038           18.32
      1          0.8220           18.50
      2          0.7865           18.47
      3          0.8074           17.87

```

(e) Centro CASSA STD.

Figura 16: Statistiche di output per ogni centro.

```

'Locale Mensa' block info:
    people in the block ..... =      1302  people
    rejected people ..... =          0  people
    loss probability ..... =          0  people

job averaged statistics:
    average interarrival time =   8.29      s
    average node wait ..... = 598.99      s
    average queue delay ..... =   0.00      s
    average service time .... = 598.99      s

time averaged statistics:
    average # in the node ... =  72.21      people
    average # in the queue ... =   0.00      people
    utilization ..... = 0.7221      -

Multi-server statistics:
    server      utilization      avg service
        0          0.8242       989.01
        1          0.7067       587.07
        2          0.6538       470.74
        3          0.9114       757.15

'Locale Mensa 2' block info:
    people in the block ..... =      1244  people
    rejected people ..... =          0  people
    loss probability ..... =          0  people

job averaged statistics:
    average interarrival time =   8.68      s
    average node wait ..... = 598.03      s
    average queue delay ..... =   0.00      s
    average service time .... = 598.03      s

time averaged statistics:
    average # in the node ... =  68.88      people
    average # in the queue ... =   0.00      people
    utilization ..... = 0.6888      -

Multi-server statistics:
    server      utilization      avg service
        0          0.7288       605.48
        1          0.6599       548.24
        2          0.8204       590.66
        3          0.7090       478.58

```

(a) centro CONSUMAZIONE 1 (solo i primi 4 posti). (b) centro CONSUMAZIONE 2 (solo i primi 4 posti).

Figura 17: Statistiche di output per ogni centro.

9.3.1 Controlli di Consistenza

I risultati per i **tempi di risposta** sono:

Centro	$E(T_Q)$	$E(S)$	$E(T_S)$
<i>PRIMO</i>	26.47	14.73	41.20
<i>SECONDO</i>	11.39	14.99	26.38
<i>DESSERT</i>	4.16	9.46	13.62
<i>CASSA FAST</i>	22.66	11.30	33.97
<i>CASSA STD</i>	10.47	18.28	28.75
<i>CONSUMAZIONE 1</i>	0	598.99	598.99
<i>CONSUMAZIONE 2</i>	0	598.03	598.03

Mentre per le **popolazioni** dei centri:

Centro	$E(N_Q)$	m	ρ	$E(N_S)$
<i>PRIMO</i>	4.76	3	0.8838	7.41
<i>SECONDO</i>	1.76	3	0.7706	4.07
<i>DESSERT</i>	0.47	2	0.5379	1.55
<i>CASSA FAST</i>	1.35	1	0.6739	2.03
<i>CASSA STD</i>	1.84	4	0.8049	5.06
<i>CONSUMAZIONE 1</i>	0	100	0.7221	72.21
<i>CONSUMAZIONE 2</i>	0	100	0.6888	68.88

9.3.2 Dati di Input

Il **numero di utenti** nel sistema con due blocchi *CONSUMAZIONE* e politica di scelta *random* vale 2546 ottenuto sommando il numero di arrivi nelle casse (vedi Fig. 16d e Fig. 16e) oppure nei blocchi consumazione:

$$E(N_F) + E(N_C) = 644 + 1902 = 2546$$

$$E(N_{S_1}) + E(N_{S_2}) = 1302 + 1244 = 2546$$

Considerando i **tempi di servizio** dei soli tre serventi del centro *PRIMO* (vedi Fig. 14a) si ottiene $E(S_1) = 14.51 \text{ s}$, $E(S_2) = 14.41 \text{ s}$, $E(S_3) = 15.32 \text{ s}$, dunque vicini al valore di riferimento $E(S) = 15 \text{ s}$. Inoltre, la media dei tempi restituisce il corrispettivo valore dell'intero blocco: $\frac{14.51+14.41+15.32}{3} = 14.73 \text{ s}$. Nella tabella che segue sono confrontati i **tassi di servizio** simulati e quelli passati come input.

Centro	μ Teorico	μ Simulato
<i>PRIMO</i>	0.06667	0.0678
<i>SECONDO</i>	0.06667	0.0667
<i>DESSERT</i>	0.10000	0.1057
<i>CASSA FAST</i>	0.0909	0.0885
<i>CASSA STD</i>	0.05554	0.0547
<i>CONSUMAZIONE 1</i>	0.001666	0.001669
<i>CONSUMAZIONE 2</i>	0.001666	0.001672

Nella tabella che segue sono presentati i risultati per le **probabilità di routing**. Anche in questo caso i valori simulati sono in linea con i valori teorici.

Sorgente i	Destinazione j	p_{ij} Teorica	p_{ij} Simulata
<i>ESTERNO</i>	<i>PRIMO</i>	0.7500	0.7635
<i>ESTERNO</i>	<i>SECONDO</i>	0.2500	0.2364
<i>PRIMO</i>	<i>SECONDO</i>	0.5500	0.5468
<i>PRIMO</i>	<i>DESSERT</i>	0.2500	0.2377
<i>PRIMO</i>	<i>CASSA FAST</i>	0.2000	0.2155
<i>SECONDO</i>	<i>DESSERT</i>	0.4500	0.4601
<i>SECONDO</i>	<i>CASSA FAST</i>	0.1375	0.1351
<i>SECONDO</i>	<i>CASSA STD</i>	0.4125	0.4048
<i>DESSERT</i>	<i>CASSA STD</i>	1.0000	1.0000
<i>CASSA FAST</i>	<i>CONSUMAZIONE 1</i>	0.5000	0.5124
<i>CASSA FAST</i>	<i>CONSUMAZIONE 2</i>	0.5000	0.4876
<i>CASSA STD</i>	<i>CONSUMAZIONE 1</i>	0.5000	0.5110
<i>CASSA STD</i>	<i>CONSUMAZIONE 2</i>	0.5000	0.4890

Per quanto riguarda i **tassi di arrivo** si ha che:

Centro	λ Teorico	λ Simulato
<i>PRIMO</i>	0.1735	0.1799
<i>SECONDO</i>	0.1534	0.1541
<i>DESSERT</i>	0.1124	0.1136
<i>CASSA FAST</i>	0.0558	0.0595
<i>CASSA STD</i>	0.1757	0.1760
<i>CONSUMAZIONE 1</i>	0.1157	0.1204
<i>CONSUMAZIONE 2</i>	0.1157	0.1151

I valori delle **visite medie** simulate e teoriche sono riportati di seguito:

Centro	v_i Teorico	v_i Simulato
<i>PRIMO</i>	0.7500	0.7635
<i>SECONDO</i>	0.6625	0.6538
<i>DESSERT</i>	0.4856	0.4822
<i>CASSA FAST</i>	0.2410	0.2528
<i>CASSA STD</i>	0.7589	0.7468
<i>CONSUMAZIONE 1</i>	0.5000	0.5111
<i>CONSUMAZIONE 2</i>	0.5000	0.4884

Nella tabella sottostante è possibile osservare come i valori simulati delle **utilizzazioni** si avvicinano ai valori teorici.

Centro	ρ Teorica	ρ Simulata
<i>PRIMO</i>	0.8677	0.8838
<i>SECONDO</i>	0.7670	0.7706
<i>DESSERT</i>	0.5620	0.5379
<i>CASSA FAST</i>	0.6138	0.6739
<i>CASSA STD</i>	0.7906	0.8049
<i>CONSUMAZIONE 1</i>	0.6945	0.7211
<i>CONSUMAZIONE 2</i>	0.6945	0.6883

9.4 QoS

Il **tempo di risposta globale** è pari a:

$$\begin{aligned}
 E(T_r^{(t)}) &= \sum_{i=1}^M E(T_i) \cdot v_i \\
 &= E(T_1) \cdot v_1 + E(T_2) \cdot v_2 + E(T_3) \cdot v_3 + E(T_F) \cdot v_F + E(T_C) \cdot v_C + E(T_{S1}) \cdot v_{S1} + E(T_{S2}) \cdot v_{S2} \\
 &= 43.8488 \cdot 0.75 + 27.7345 \cdot 0.6625 + 14.6181 \cdot 0.4856 + 28.4897 \cdot 0.2410 + 30.4488 \cdot 0.7589 + 600 \cdot 0.5000 + 600 \cdot 0.5000 \\
 &= 688.2712 \text{ s}
 \end{aligned}$$

Il risultato simulato è invece:

$$E(T_r^{(s)}) = 683.8556 \text{ s}$$

Anche in questa configurazione la **probabilità di perdita** è:

$$p_{loss,S1}^{(t)} = p_{loss,S2}^{(t)} \simeq p_{loss,S1}^{(s)} = p_{loss,S2}^{(s)} = 0$$

10 Modello Migliorativo 2 con Politica Least Busy

Rispetto a quanto discusso in Sec. 9, la scelta dell'area dove consumare il pasto dipende dalle utilizzazioni. Infatti intuitivamente, un utente preferisce il locale meno occupato, in modo tale da aumentare le sue probabilità di trovare un posto a sedere. Sia per il **modello concettuale** (vedi Fig. 15) che per il **modello delle specifiche** e il **modello computazionale** vale quanto già discusso in Sec. 9.

10.1 Verifica

I dati ottenuti fanno riferimento ad una singola simulazione con durata pari al periodo di osservazione (i.e., 3 ore) con seed pari a 123456789 (vedi Fig. 18 e Fig. 19).

```

'Primi' block info:
    people in the block .... =     1895  people

job averaged statistics:
    average interarrival time =  5.70      s
    average node wait ..... = 49.46      s
    average queue delay ..... = 34.11      s
    average service time .... = 15.35      s

time averaged statistics:
    average # in the node ... =  8.68      people
    average # in the queue .. =  5.99      people
    utilization ..... = 0.8978      -

routing probabilities statistics:
    P(to SECONDO) ..... = 0.5404
    P(to CASSA FAST) .... = 0.2000
    P(to DESSERT) ..... = 0.2596

Multi-server statistics:
    server      utilization      avg service
    0          0.8962            15.39
    1          0.8944            15.71
    2          0.9029            14.98

```

(a) Centro PRIMI.

```

'Secondi e Contorni' block info:
    people in the block .... =     1614  people

job averaged statistics:
    average interarrival time =  6.69      s
    average node wait ..... = 25.68      s
    average queue delay ..... = 10.94      s
    average service time .... = 14.74      s

time averaged statistics:
    average # in the node ... =  3.84      people
    average # in the queue .. =  1.64      people
    utilization ..... = 0.7341      -

routing probabilities statistics:
    P(to CASSA FAST) .... = 0.1252
    P1(to CASSA STD) .... = 0.4157
    P2(to DESSERT) ..... = 0.4591

Multi-server statistics:
    server      utilization      avg service
    0          0.7233            15.29
    1          0.7343            14.42
    2          0.7448            14.55

```

(b) Centro SECONDI.

```

'Frutta e Dessert' block info:
    people in the block .... =     1233  people

job averaged statistics:
    average interarrival time =  8.75      s
    average node wait ..... = 12.42      s
    average queue delay ..... = 2.95      s
    average service time .... = 9.47      s

time averaged statistics:
    average # in the node ... =  1.42      people
    average # in the queue .. =  0.34      people
    utilization ..... = 0.5406      -

routing probabilities statistics:
    P(to CASSA STD) .... = 1.0000      s

Multi-server statistics:
    server      utilization      avg service
    0          0.5317            9.20
    1          0.5495            9.75

```

(c) Centro DESSERT.

```

'Casse Fast' block info:
    people in the block .... =     581  people

job averaged statistics:
    average interarrival time = 18.58      s
    average node wait ..... = 23.09      s
    average queue delay ..... = 12.32      s
    average service time .... = 10.77      s

time averaged statistics:
    average # in the node ... =  1.24      people
    average # in the queue .. =  0.66      people
    utilization ..... = 0.5796      -

routing probabilities statistics:
    P(to CONSUMAZIONE 1) ..... = 0.4716
    P(to CONSUMAZIONE 2) ..... = 0.5284

Multi-server statistics:
    server      utilization      avg service
    0          0.5796            10.77

```

(d) Centro CASSA FAST.

```

'Casse standard' block info:
    people in the block .... =     1904  people

job averaged statistics:
    average interarrival time =  5.67      s
    average node wait ..... = 27.47      s
    average queue delay ..... = 9.18      s
    average service time .... = 18.29      s

time averaged statistics:
    average # in the node ... =  4.84      people
    average # in the queue .. =  1.62      people
    utilization ..... = 0.8061      -

routing probabilities statistics:
    P(to CONSUMAZIONE 1) ..... = 0.4942
    P(to CONSUMAZIONE 2) ..... = 0.5058

Multi-server statistics:
    server      utilization      avg service
    0          0.7994            18.21
    1          0.7974            17.65
    2          0.8019            17.93
    3          0.8258            19.43

```

(e) Centro CASSA STD.

Figura 18: Statistiche di output per ogni centro.

```
'Locale Mensa' block info:
    people in the block ..... =      1215  people
    rejected people ..... =          0  people
    loss probability ..... =          0  people

job averaged statistics:
    average interarrival time =   8.88      s
    average node wait ..... = 615.23      s
    average queue delay ..... =   0.00      s
    average service time ..... = 615.23      s

time averaged statistics:
    average # in the node ... =  69.21     people
    average # in the queue ... =   0.00     people
    utilization ..... = 0.6921      -
Multi-server statistics:
    server      utilization      avg service
      0          0.8545           615.22
      1          0.8794           949.72
      2          0.8760           860.09
      3          0.6695           516.45

'Locale Mensa 2' block info:
    people in the block ..... =      1270  people
    rejected people ..... =          0  people
    loss probability ..... =          0  people

job averaged statistics:
    average interarrival time =   8.50      s
    average node wait ..... = 585.93      s
    average queue delay ..... =   0.00      s
    average service time ..... = 585.93      s

time averaged statistics:
    average # in the node ... =  68.90     people
    average # in the queue ... =   0.00     people
    utilization ..... = 0.6890      -
Multi-server statistics:
    server      utilization      avg service
      0          0.6185           513.85
      1          0.7486           621.91
      2          0.7000           629.99
      3          0.5740           387.43
```

(a) centro CONSUMAZIONE 1 (solo i primi 4 posti). (b) centro CONSUMAZIONE 2 (solo i primi 4 posti).

Figura 19: Statistiche di output per ogni centro.

10.1.1 Controlli di Consistenza

La condizione per i **tempi di risposta** è verificata per ogni blocco, come mostrato dai tempi simulati riportati di seguito:

Centro	$E(T_Q)$	$E(S)$	$E(T_S)$
<i>PRIMO</i>	34.11	15.35	49.46
<i>SECONDO</i>	10.94	14.74	25.68
<i>DESSERT</i>	2.95	9.47	12.42
<i>CASSA FAST</i>	12.32	10.77	23.09
<i>CASSA STD</i>	9.18	18.29	27.47
<i>CONSUMAZIONE 1</i>	0	615.23	615.23
<i>CONSUMAZIONE 2</i>	0	585.93	585.93

Come anche per le **popolazioni** dei centri:

Centro	$E(N_Q)$	m	ρ	$E(N_S)$
<i>PRIMO</i>	5.99	3	0.8978	8.68
<i>SECONDO</i>	1.64	3	0.7341	3.84
<i>DESSERT</i>	0.34	2	0.5406	1.42
<i>CASSA FAST</i>	0.66	1	0.5796	1.24
<i>CASSA STD</i>	1.62	4	0.8061	4.84
<i>CONSUMAZIONE 1</i>	0	100	0.6921	69.21
<i>CONSUMAZIONE 2</i>	0	100	0.6890	68.90

10.1.2 Dati di Input

Il **numero di utenti** entrato nel sistema è 2485 ottenuto sommando il numero di arrivi per i centri *CASSA FAST* e *CASSA STD* (vedi Fig. 18d e Fig. 18e): $581 + 1904 = 2485$.

Con i **tempi di servizio** del centro *PRIMO* (vedi Fig. 18a) si ottiene $E(S_1) = 15.39\text{ s}$, $E(S_2) = 15.71\text{ s}$, $E(S_3) = 14.98\text{ s}$, dunque vicini al valore di riferimento $E(S) = 15\text{ s}$. Inoltre, la media dei tempi restituisce il corrispettivo valore dell'intero blocco: $\frac{15.39+15.71+14.98}{3} = 15.35\text{ s}$. Di seguito sono confrontati i **tassi di servizio** simulati e quelli passati in input.

Centro	μ Teorico	μ Simulato
<i>PRIMO</i>	0.06667	0.0651
<i>SECONDO</i>	0.06667	0.0678
<i>DESSERT</i>	0.10000	0.1056
<i>CASSA FAST</i>	0.0909	0.0928
<i>CASSA STD</i>	0.05554	0.05467
<i>CONSUMAZIONE 1</i>	0.001666	0.001625
<i>CONSUMAZIONE 2</i>	0.001666	0.001706

Nella tabella che segue sono presentati i risultati per le **probabilità di routing**. Anche in questo caso i valori simulati sono in linea con i valori teorici.

Sorgente i	Destinazione j	p_{ij} Teorica	p_{ij} Simulata
<i>ESTERNO</i>	<i>PRIMO</i>	0.7500	0.7625
<i>ESTERNO</i>	<i>SECONDO</i>	0.2500	0.2374
<i>PRIMO</i>	<i>SECONDO</i>	0.5500	0.5404
<i>PRIMO</i>	<i>DESSERT</i>	0.2500	0.2596
<i>PRIMO</i>	<i>CASSA FAST</i>	0.2000	0.2000
<i>SECONDO</i>	<i>DESSERT</i>	0.4500	0.4591
<i>SECONDO</i>	<i>CASSA FAST</i>	0.1375	0.1252
<i>SECONDO</i>	<i>CASSA STD</i>	0.4125	0.4157
<i>DESSERT</i>	<i>CASSA STD</i>	1.0000	1.0000
<i>CASSA FAST</i>	<i>CONSUMAZIONE 1</i>	0.5000	0.4716
<i>CASSA FAST</i>	<i>CONSUMAZIONE 2</i>	0.5000	0.5284
<i>CASSA STD</i>	<i>CONSUMAZIONE 1</i>	0.5000	0.4942
<i>CASSA STD</i>	<i>CONSUMAZIONE 2</i>	0.5000	0.5058

Per quanto riguarda i **tassi di arrivo** si ha che:

Centro	λ Teorico	λ Simulato
<i>PRIMO</i>	0.1735	0.1754
<i>SECONDO</i>	0.1534	0.1493
<i>DESSERT</i>	0.1124	0.1140
<i>CASSA FAST</i>	0.0558	0.0537
<i>CASSA STD</i>	0.1757	0.1761
<i>CONSUMAZIONE 1</i>	0.1157	0.1124
<i>CONSUMAZIONE 2</i>	0.1157	0.1175

E' possibile osservare i valori delle **visite medie** simulate e teoriche:

Centro	v_i Teorico	v_i Simulato
<i>PRIMO</i>	0.7500	0.7625
<i>SECONDO</i>	0.6625	0.6494
<i>DESSERT</i>	0.4856	0.4960
<i>CASSA FAST</i>	0.2410	0.2338
<i>CASSA STD</i>	0.7589	0.7659
<i>CONSUMAZIONE 1</i>	0.5000	0.4887
<i>CONSUMAZIONE 2</i>	0.5000	0.5109

Nella tabella sottostante è possibile osservare come i valori simulati delle **utilizzazioni** si avvicinano ai valori teorici.

Centro	ρ Teorica	ρ Simulata
<i>PRIMO</i>	0.8677	0.8978
<i>SECONDO</i>	0.7670	0.7341
<i>DESSERT</i>	0.5620	0.5406
<i>CASSA FAST</i>	0.6138	0.5796
<i>CASSA STD</i>	0.7906	0.8061
<i>CONSUMAZIONE 1</i>	0.6945	0.6921
<i>CONSUMAZIONE 2</i>	0.6945	0.6890

10.2 QoS

Per il **tempo di risposta globale** si ha:

$$\begin{aligned}
 E(T_r^{(t)}) &= \sum_{i=1}^M E(T_i) \cdot v_i \\
 &= E(T_1) \cdot v_1 + E(T_2) \cdot v_2 + E(T_3) \cdot v_3 + E(T_F) \cdot v_F + E(T_C) \cdot v_C + E(T_{S1}) \cdot v_{S1} + E(T_{S2}) \cdot v_{S2} \\
 &= 43.8488 \cdot 0.75 + 27.7345 \cdot 0.6625 + 14.6181 \cdot 0.4856 + 28.4897 \cdot 0.2410 + 30.4488 \cdot 0.7589 + 600 \cdot 0.5000 + 600 \cdot 0.5000 \\
 &\quad = 688.2712 \text{ s}
 \end{aligned}$$

Il risultato simulato è invece:

$$E(T_r^{(s)}) = 687.2456 \text{ s}$$

Anche in questa configurazione la **probabilità di perdita** è:

$$p_{loss,S1}^{(t)} = p_{loss,S2}^{(t)} \simeq p_{loss,S1}^{(s)} = p_{loss,S2}^{(s)} = 0$$

11 Risultati

Sono ora analizzati i risultati ottenuti nei modelli migliorativi, anche confrontandoli con il modello base da cui si è partiti.

11.1 Simulazione finite-horizon

Nelle Fig. 20, 21 e 22 sono mostrati i valori degli intervalli di confidenza all'aumentare del numero di **repliche**. Per i tempi di risposta vale quanto già detto riguardo al modello base e in particolare questi si discostano dal tempo di risposta stazionario teorico in quanto il tempo di simulazione di tre ore è troppo breve e quindi è ancora presente un bias rispetto allo stato iniziale della rete.

Per un ensemble con 10000 repliche, il tempo di risposta **transiente** è 686.869893 ± 0.314576 .

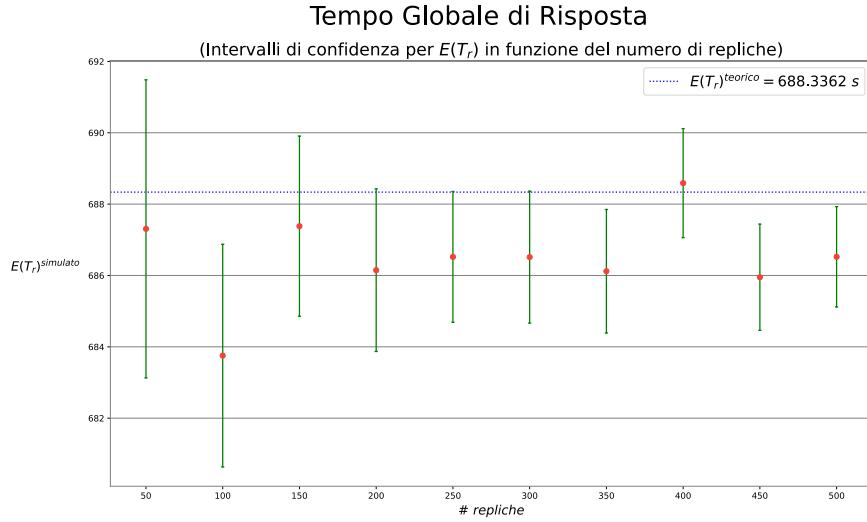


Figura 20: Intervalli di confidenza del tempo globale di risposta della rete all'aumentare del numero di repliche per il modello migliorativo con 200 posti nel blocco *CONSUMAZIONE*.

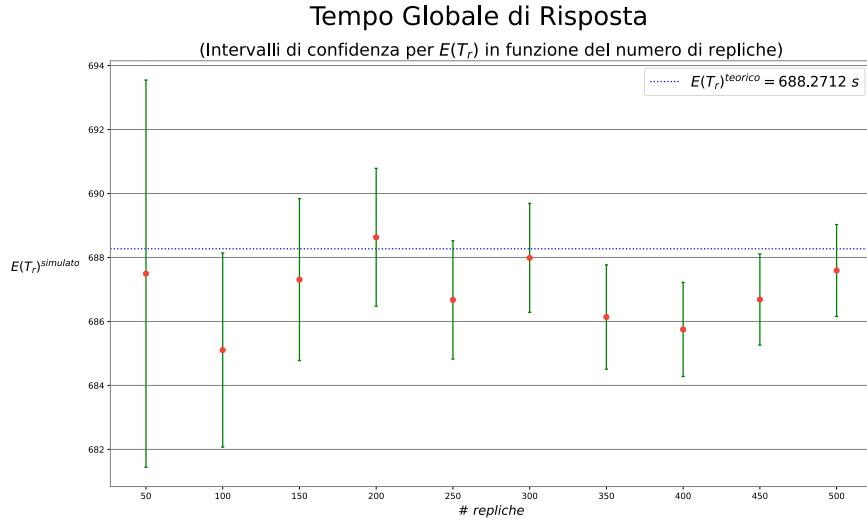


Figura 21: Intervalli di confidenza del tempo globale di risposta della rete all'aumentare del numero di repliche per il modello migliorativo 2 con politica random.

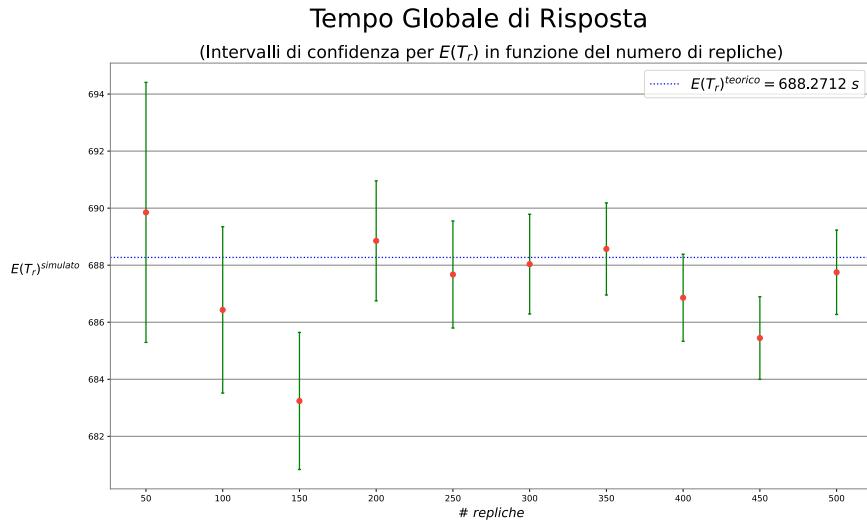


Figura 22: Intervalli di confidenza del tempo globale di risposta della rete all'aumentare del numero di repliche per il modello migliorativo 2 con politica least busy.

Nelle Fig. 23, 24 e 25 vengono presentati gli andamenti dei valori per i **seed** al passare del tempo. Per il modello migliorativo 1 i tempi per una singola simulazione hanno un andamento che tende al valore teorico, a differenza degli altri.

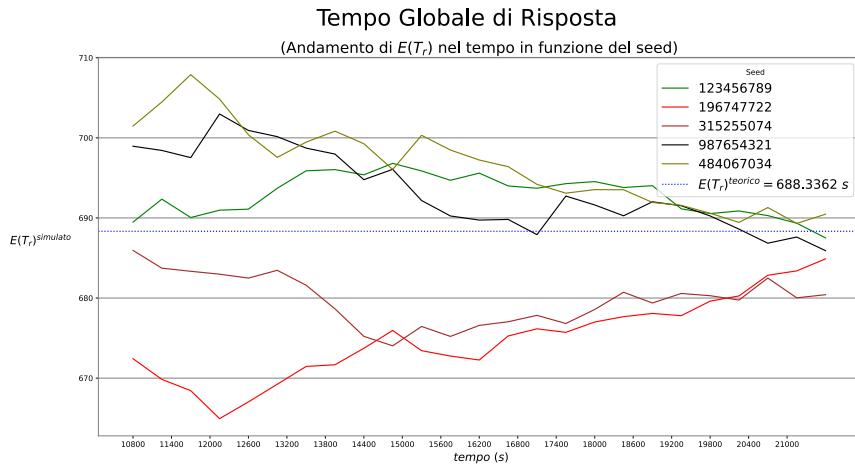


Figura 23: Andamento del tempo globale di risposta al variare del seed di sistema per il modello migliorativo con 200 posti a sedere.

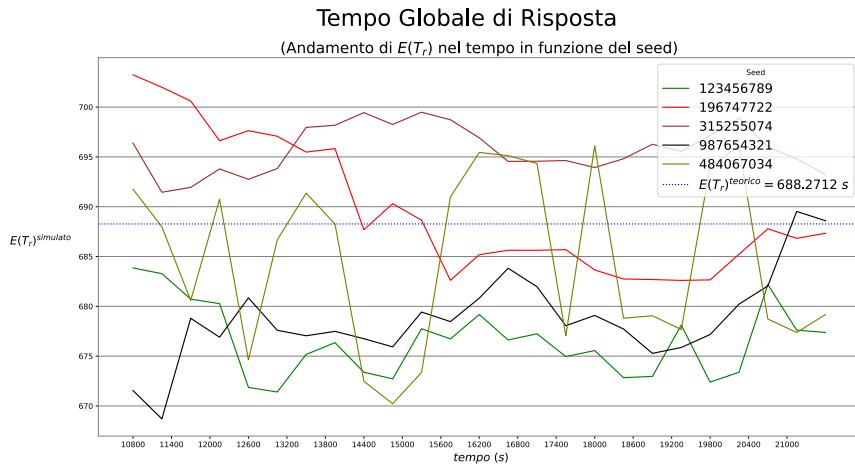


Figura 24: Andamento del tempo globale di risposta al variare del seed di sistema per il modello migliorativo 2 con politica random.

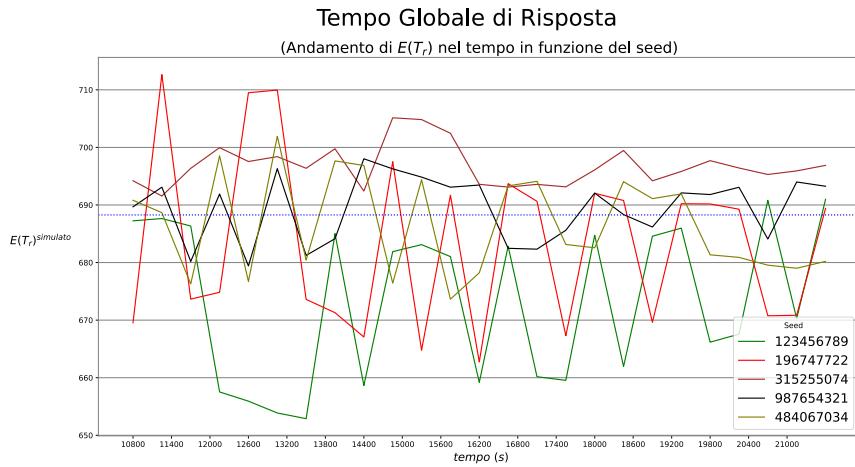


Figura 25: Andamento del tempo globale di risposta al variare del seed di sistema per il modello migliorativo 2 con politica least busy.

11.2 Simulazione Infinite-Horizon

Infine sono confrontati i tempi dei quattro modello studiati (vedi Fig. 26) nella **simulazione infinita**. Il grafico mostra una convergenza per i modelli migliorativi al valore teorico, con quest'ultimo più elevato rispetto al modello iniziale. Questo incremento è dovuto al maggior numero di dipendenti che consuma il pasto e dunque a tempi di servizio aggiuntivi rispetto al modello base.

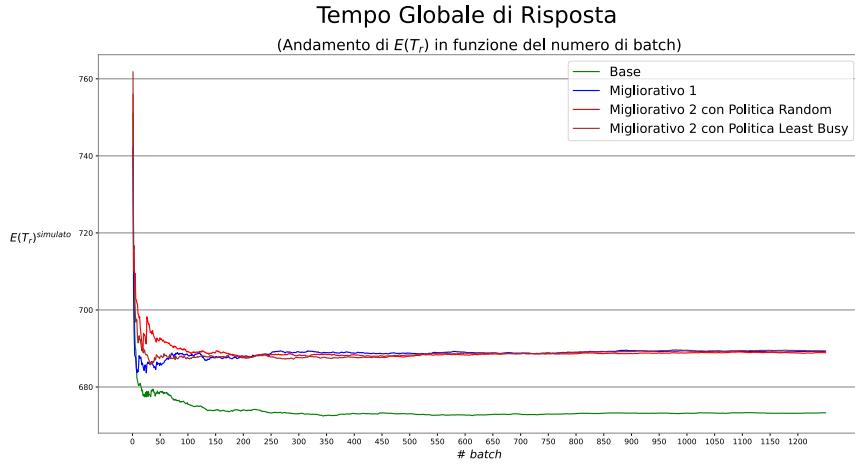


Figura 26: Andamento del tempo globale di risposta medio all'aumentare del numero di batch per i quattro modelli.

Gli intervalli di confidenza al variare del seed nella simulazione infinite sono riportati in Fig. 27, 28 e 29.

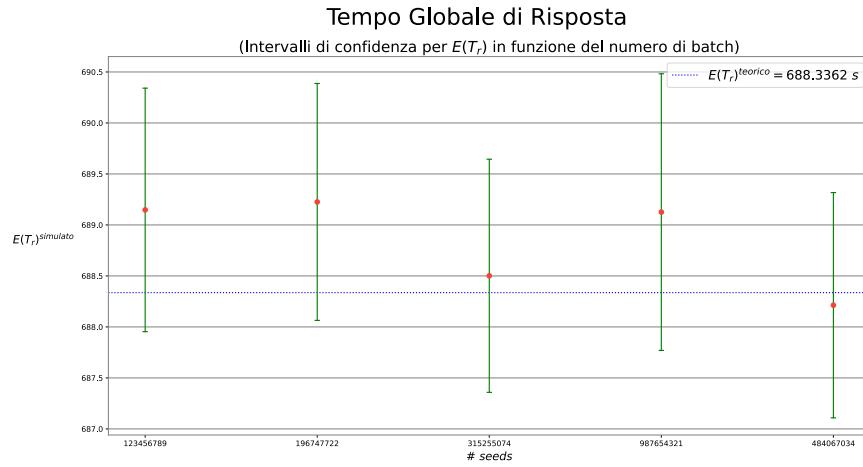


Figura 27: Intervalli di confidenza per il tempo globale di risposta medio al variare del seed per il modello migliorativo 1.

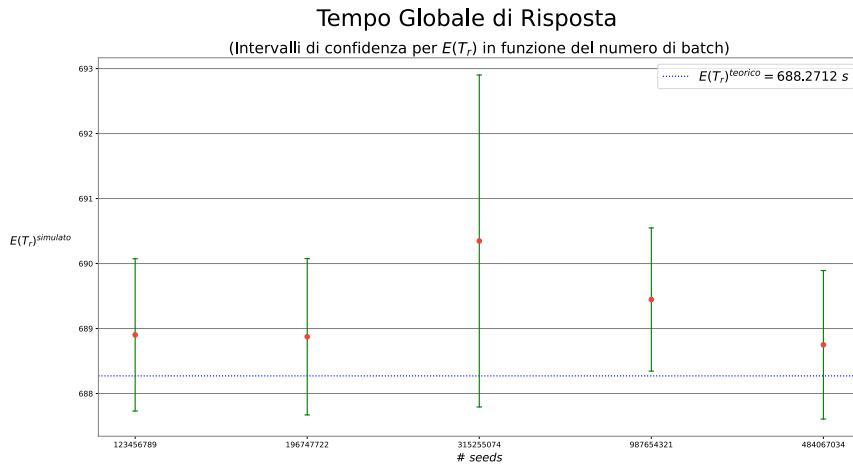


Figura 28: Intervalli di confidenza per il tempo globale di risposta medio al variare del seed per il modello migliorativo 2 con politica random.

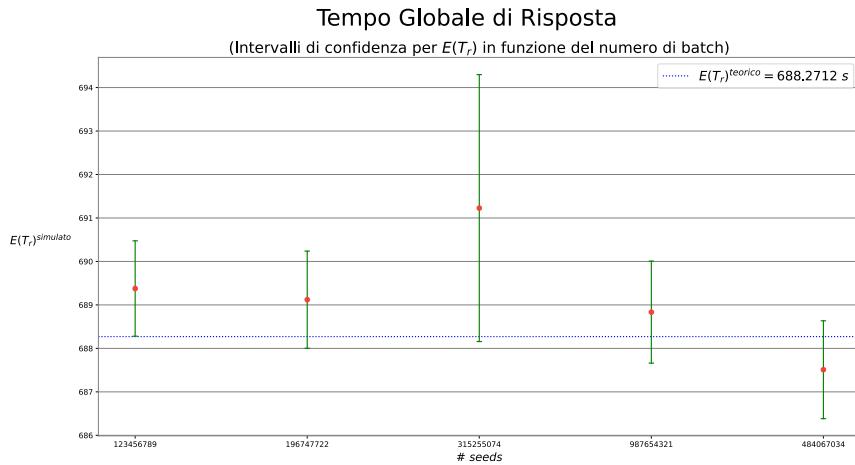


Figura 29: Intervalli di confidenza per il tempo globale di risposta medio al variare del seed per il modello migliorativo 2 con politica least busy.

11.3 Probabilità di Perdita

Dall’analisi dei risultati conseguiti nei modelli migliorativi, si è visto come la probabilità di perdita sia nulla. Tutti i dipendenti dunque trovano un posto a sedere dove poter consumare il proprio pasto. Non sono così riportati grafici di alcun genere.

12 Conclusioni

Riassumendo il lavoro svolto, si è partiti da un modello base dal quale si sono estrapolati i dati necessari per analizzarlo, trovando così delle limitazioni. In seguito, sono stati definiti tre modelli migliorativi, ognuno con una sua implementazione specifica, che annullano la probabilità di perdita a scapito di un aumento del tempo di risposta globale.

Un’ultima osservazione riguarda la progettazione del sistema, in particolare la mancata considerazione dei **costi** di mantenimento e manutenzione, che sicuramente incidono sulle scelte da fare.