

REPORT PROGETTO PMCSN – Luca Capotombolo e Matteo Fanfarillo

Sommario.

Descrizione del sistema.	3
Problematiche del sistema.	3
Obiettivi.	4
Modello concettuale.	4
Politiche di scheduling dei job all'interno delle code.	5
Stato di code e server.	5
Eventi.	5
Descrizione degli eventi.	6
Profitto del sistema.	9
Modello delle specifiche.	10
Modellazione dei centri.	10
Matrice di routing.	12
Fasce orarie.	12
Proporzione tra numero di automobili e numero di famiglie.	12
Algoritmi per la Next-Event simulation.	13
Modello computazionale.	18
Stato del sistema.	18
Eventi del sistema.	18
Politica di selezione del server <i>idle</i>	20
PRNG utilizzato.	20
Verifica.	20
Caso 1.	20
Caso 2.	21
Controlli di consistenza.	21
Validazione.	22
Progettazione degli esperimenti.	23
Analisi del collo di bottiglia.	24
Servizio medio dei centri del sistema.	24
Calcolo del numero medio di visite ai centri del sistema.	24
Calcolo della domanda media per i centri del sistema.	25
Esecuzione delle simulazioni.	25
Simulazione a orizzonte infinito.	25
Fascia 1 (09:00 → 11:00).	25

Fascia 2 (11:00 → 12:00).....	26
Fascia 3 (12:00 → 15:00).....	26
Fascia 4 (15:00 → 18:00).....	27
Fascia 5 (18:00 → 22:00).....	27
Fascia 6 (22:00 → 23:00).....	28
Conclusioni.	28
Simulazione a orizzonte finito.	28
Fascia 1 (09:00 → 11:00).....	29
Fascia 2 (11:00 → 12:00).....	30
Fascia 3 (12:00 → 15:00).....	30
Fascia 4 (15:00 → 18:00).....	30
Fascia 5 (18:00 → 22:00).....	30
Fascia 6 (22:00 → 23:00).....	30
Conclusioni.	31
Modello migliorativo.	32
Descrizione del modello migliorativo.	32
Obiettivi del modello migliorativo.....	32
Modello concettuale.	33
Politiche di scheduling dei job all'interno delle code.....	33
Stato di code e serventi.	34
Eventi.	34
Modello delle specifiche.....	34
Modellazione dei centri.....	34
Algoritmi per la Next-Event simulation.	35
Calcolo delle probabilità di perdita del centro electronic ordering and payment.....	37
Modello computazionale.....	37
Verifica.....	37
Controlli di consistenza.....	38
Validazione.	38
Progettazione degli esperimenti.	39
Analisi del collo di bottiglia.....	40
Esecuzione delle simulazioni.	41
Simulazione a orizzonte finito.	41
Simulazione a orizzonte infinito.	43

Descrizione del sistema.

Il sistema che è stato analizzato è un ristorante di una catena di fast food statunitense molto nota e frequentata: il McDonald's. Generalmente, questo tipo di ristoranti rappresenta un punto di incontro per giovani e famiglie e si trova all'interno di centri commerciali oppure in prossimità dei luoghi più frequentati delle città. L'ordinazione può essere effettuata direttamente all'interno del ristorante oppure vi è la possibilità di accodarsi all'interno del McDrive ordinando e pagando senza dover scendere dalla propria vettura. Nel caso di utilizzo del McDrive, le vetture si dispongono all'interno di una fila e la vettura in testa alla coda potrà ordinare da mangiare e pagare. Le famiglie che entrano all'interno del ristorante possono effettuare l'ordinazione e il pagamento usufruendo delle casse fisiche gestite dai dipendenti oppure utilizzando le casse automatiche posizionate all'ingresso del ristorante. Solitamente, il ristorante dispone di una struttura adibita per l'area giochi, che può essere acceduta da alcune famiglie con bambini dopo la consumazione del pasto. Tale servizio nei McDonald's è gratuito ma dispone di un numero limitato di posti. Di conseguenza, nel momento in cui un bambino vuole usufruire dell'area giochi, potrebbe non trovare posto; in tale evenienza, il bambino e la sua famiglia tipicamente non hanno il tempo o la pazienza di attendere, per cui abbandonano il ristorante, causando l'insoddisfazione del bambino. È per questo motivo che la disponibilità dell'area giochi rappresenta un ulteriore fattore che determina la qualità del servizio offerto dal ristorante.

Problematiche del sistema.

Essendo il McDonald's un ristorante molto frequentato, negli orari di punta è possibile trovare alcune decine o centinaia di persone al suo interno, a seconda della sua dimensione. Per questo motivo, spesso l'attesa che il cliente sperimenta prima di poter ordinare da mangiare è così elevata da spingerlo a uscire dal ristorante. Inoltre, una volta effettuata l'ordinazione, qualora l'organizzazione del ristorante non fosse ottimale, il tempo di attesa relativo alla consegna del pasto ordinato potrebbe essere lungo e il cliente potrebbe ricevere del cibo freddo. Basandosi su queste problematiche, sono stati presi in considerazione gli aspetti critici riportati in seguito.

- Aspetti critici riguardanti i clienti (che tipicamente sono raggruppati in famiglie) che vogliono consumare il pasto all'interno del ristorante:

1) L'attesa per ordinare il pasto e pagare alla cassa fisica o, in alternativa, per utilizzare un dispositivo elettronico (cassa automatica) che permette di effettuare l'ordine e il pagamento.

2) L'attesa per ritirare il pasto.

3) L'attesa per trovare un tavolo libero per poter consumare il pasto, da cui dipende anche la qualità del cibo (e.g. se l'attesa è eccessiva, potrebbe essere consumato un pasto freddo).

4) La disponibilità dell'area giochi.

- Aspetti critici riguardanti i clienti che vogliono portare il pasto a casa utilizzando il McDrive:

1) L'attesa per ordinare il pasto e per effettuare il pagamento al McDrive.

Se non venissero curati questi aspetti, si potrebbe andare incontro a una perdita di fiducia di alcuni clienti nei confronti del ristorante, per cui potrebbero non tornare più nel futuro.

Obiettivi.

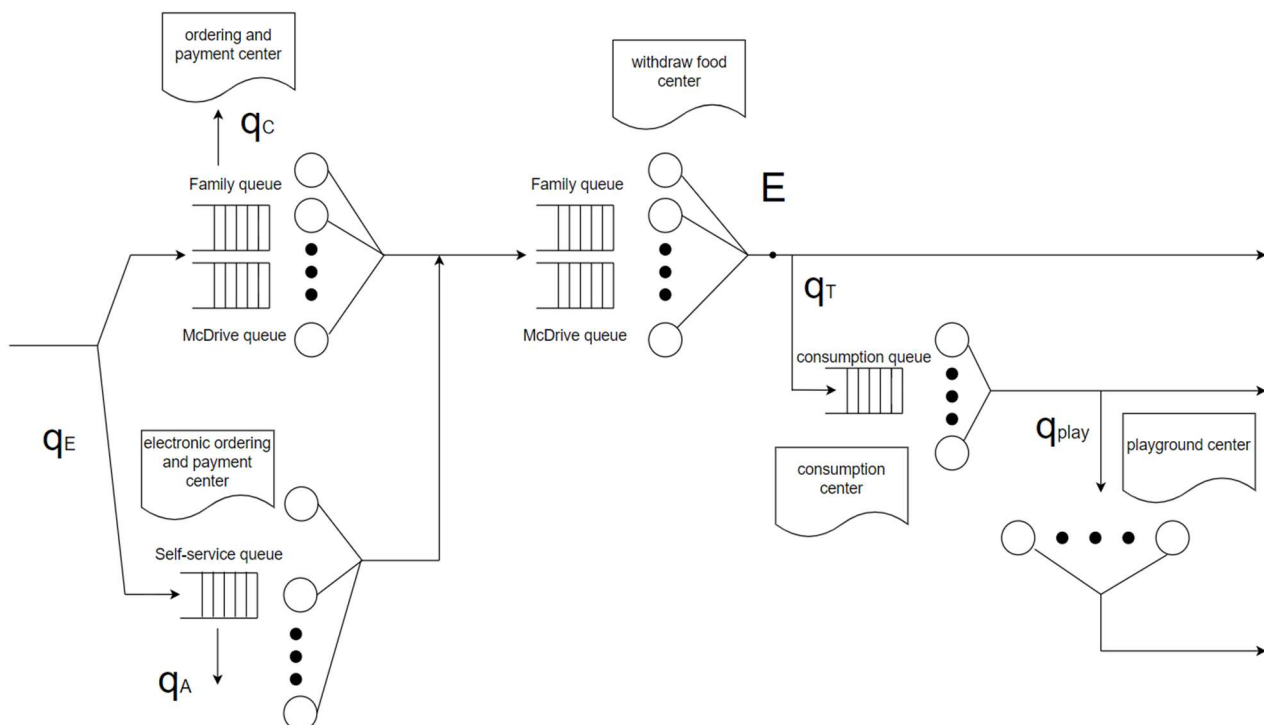
Lo studio che è stato condotto sul sistema è finalizzato al raggiungimento dei seguenti obiettivi:

- Si vuole determinare la configurazione migliore del sistema per massimizzare il profitto del ristorante, tenendo conto che il servizio è più efficiente all'aumentare del numero dei dipendenti, dei dispositivi elettronici per l'ordinazione, dei tavoli e dei posti disponibili nell'area giochi; tuttavia, è necessario trovare un compromesso tra la qualità del servizio e il costo necessario per garantirla (i.e. stipendi per i dipendenti e costi legati all'affitto e/o alla manutenzione).
- Si vuole garantire i seguenti QoS (Quality of Service):
 - 1) Il tempo medio giornaliero per un cliente per ritirare il cibo deve essere inferiore ai 15 minuti (attesa in coda inclusa).
 - 2) Il tempo medio giornaliero in cui un cliente, dopo aver pagato e ricevuto il pasto, rimane in attesa che si liberi un tavolo per poter consumare il pasto deve essere minore di 30 secondi.
 - 3) Almeno l'80% dei bambini che richiedono nell'arco della giornata l'uso dell'area giochi deve trovare un posto libero.

Modello concettuale.

Gli utenti del sistema possono essere delle famiglie, che accedono all'interno del ristorante, oppure delle automobili, che fanno uso del McDrive limitandosi a ordinare e pagare il pasto. Ciascuna di queste due tipologie di utenti è stata modellata mediante un singolo job.

Il system diagram del sistema in esame è illustrato nella seguente figura:



q_E : job che utilizzano le casse automatiche per ordinare e pagare
 q_E^C : job che utilizzano le casse fisiche per ordinare e pagare
 q_A : job che abbandonano il centro relativo alle casse automatiche
 q_C : job che abbandonano il centro relativo alle casse fisiche
 q_T : job che vanno a consumare il pasto al tavolo
 q_T^C : job che, dopo aver ricevuto il pasto, escono dal sistema
 q_{play} : job che vanno verso l'area giochi
 q_{play}^C : job che, dopo aver consumato il pasto, escono dal sistema
 E: punto del sistema a cui giungono tutti e soli i job che apportano un ricavo

Politiche di scheduling dei job all'interno delle code.

Il centro corrispondente a ordine e pagamento (*ordering and payment*) e quello corrispondente al ritiro del cibo (*withdraw food*) presentano due code distinte poiché sono relativi ai due servizi di cui si può usufruire anche attraverso il McDrive. Infatti, una coda (detta *McDrive queue* o *coda del McDrive*) è riservata alle automobili, mentre l'altra (detta *family queue* o *coda delle famiglie*) è riservata alle famiglie che entrano all'interno del fast food. Ciascuna di queste due code, se presa singolarmente, è caratterizzata da uno scheduling FIFO (è naturale pensare che una famiglia che accede al ristorante dopo un'altra famiglia riceverà il servizio successivamente e, analogamente, un'automobile che accede al McDrive dopo un'altra automobile riceverà il servizio successivamente). D'altra parte, i job appartenenti alla coda del McDrive (ovvero le automobili) hanno priorità sui job appartenenti alla coda delle famiglie (ovvero le famiglie); tuttavia, per come è fatta fisicamente la struttura del McDrive, è impossibile servire più automobili per volta all'interno dello stesso centro. In altre parole, se un'automobile è già in servizio, gli altri serventi possono occuparsi solo delle famiglie, lasciando eventuali altre automobili in attesa.

I centri relativi alle casse automatiche (*electronic ordering and payment*) e alla consumazione del pasto al tavolo (*consumption*) presentano un'unica coda caratterizzata da uno scheduling FIFO.

Infine, il centro corrispondente all'area giochi (*playground*) non ha una coda per effetto del fatto che, come specificato in fase di descrizione del sistema, le famiglie in generale non sono disposte ad aspettare che si liberi un posto per mandare il proprio bambino a giocare.

Stato di code e serventi.

In qualunque istante, lo stato di ciascuna coda può essere *empty* (vuota) o *not empty* (non vuota), lo stato di ciascun servente dei centri con una singola coda può essere *idle* (libero) o *busy* (occupato), e lo stato di ciascun servente dei centri con due code (coda del McDrive e coda delle famiglie) può essere *idle* (libero), *busy_a* (occupato con un'automobile) oppure *busy_f* (occupato con una famiglia).

Eventi.

Gli eventi considerati in tutti i centri del sistema in esame sono:

- Completamento del servizio di un job.
- Cambio di fascia oraria.

Altri eventi considerati all'interno del centro *ordering and payment* sono:

- Arrivo di un nuovo job nella coda del McDrive.
- Arrivo di un nuovo job nella coda delle famiglie.
- Perdita di un job dalla coda delle famiglie.

Altri eventi considerati all'interno del centro *withdraw food* sono:

- Arrivo di un nuovo job nella coda del McDrive.
- Arrivo di un nuovo job nella coda delle famiglie.

Altri eventi considerati all'interno del centro *electronic ordering and payment* sono:

- Arrivo di un nuovo job.
- Perdita di un job.

Infine, un altro evento considerato all'interno del centro *consumption* e all'interno del centro *playground* è l'arrivo di un nuovo job.

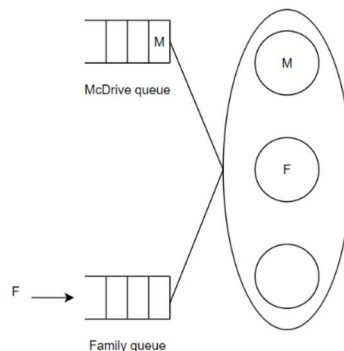
Descrizione degli eventi.

Se arriva una famiglia, può ordinare e pagare alla cassa oppure in modalità self-service tramite una cassa automatica. Se arriva un'automobile, può ordinare e pagare solo alla cassa, impegnando dunque un servente che lavora in cassa.

Per motivi logistici, può essere servita al più un'automobile per volta da un qualsiasi servente. Quando un'automobile arriva nel centro *ordering and payment*, verrà servita senza attesa se la coda del McDrive è vuota e almeno un servente è *idle* e non c'è alcuna altra automobile in servizio; in caso contrario, si mette in attesa nell'apposita coda del McDrive. Quando un'automobile termina ordine e pagamento, il servente controlla lo stato della coda del McDrive e, se è *not empty*, serve la prossima automobile; altrimenti, se lo stato della coda delle famiglie è *not empty*, serve la prossima famiglia; altrimenti, lo stato del servente passa da *busy_a* a *idle*.

Quando una famiglia arriva nel centro *ordering and payment*, verrà servita senza attesa se la coda del McDrive è *empty* oppure in quell'istante un'automobile è in servizio, e se la coda delle famiglie è *empty* e almeno un servente è *idle*; in caso contrario, si mette in attesa nell'apposita coda delle famiglie. Di fatto, se esiste un servente nello stato *busy_a*, non è necessario che la coda del McDrive sia *not empty* affinché la famiglia possa ordinare e pagare senza attesa: in tal caso, ricordando che non possono essere servite due automobili contemporaneamente, eventuali serventi *idle* si mettono a disposizione delle famiglie finché l'automobile correntemente in servizio non avrà terminato ordine e pagamento.

La seguente figura mostra un esempio di tale situazione:

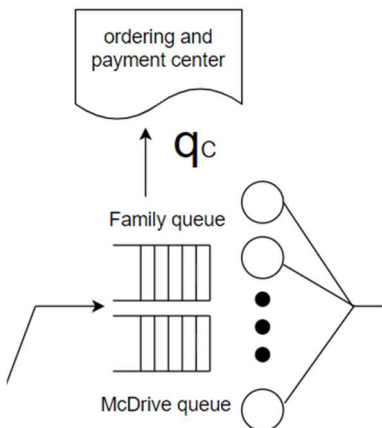


Espressione errata: (McDrive queue empty) AND (Family queue empty) AND (exists servente idle)

Espressione corretta: (McDrive queue empty OR exists servente busy_a) AND (Family queue empty) AND (exists servente idle)

Quando una famiglia termina ordine e pagamento, il servente passa allo stato *idle* se la coda del McDrive e quella delle famiglie sono entrambe *empty* oppure se la coda delle famiglie è *empty* e c'è un altro servente nello stato *busy_a*; invece, serve una macchina se la coda del McDrive è *not empty* e non ci sono altri serventi nello stato *busy_a*; infine, serve una famiglia se la coda del McDrive è *empty* e quella della famiglia è *not empty* oppure se la coda della famiglia è *not empty* e un altro servente è nello stato *busy_a*.

Se una famiglia rimane troppo tempo in attesa nella coda relativa all'ordine e al pagamento può abbandonare il ristorante.

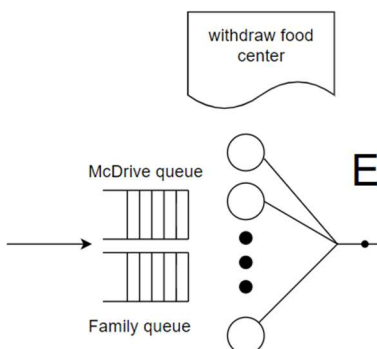


Ordering and payment center è, alternativamente a electronic ordering and payment center, il primo centro visitato dai job.

I job, una volta usciti da questo centro, si dirigono verso il centro relativo al ritiro del cibo (withdraw food center).

Quando un'automobile termina ordine e pagamento, si accoda per ricevere il cibo sempre nell'apposita coda del McDrive e verrà servita senza attesa se la coda del McDrive è vuota e almeno un servente è *idle* e non c'è alcuna altra automobile che in quell'istante sta ricevendo il pasto; in caso contrario, si mette in attesa nell'apposita coda del McDrive. Dopo aver ricevuto il pasto, le automobili escono dal sistema. Quando un'automobile ha ricevuto il pasto, il servente controlla lo stato della coda del McDrive e, se è *not empty*, serve la prossima automobile; altrimenti, se lo stato della coda delle famiglie è *not empty*, serve la prossima famiglia; altrimenti, lo stato del servente passa da *busy_a* a *idle*.

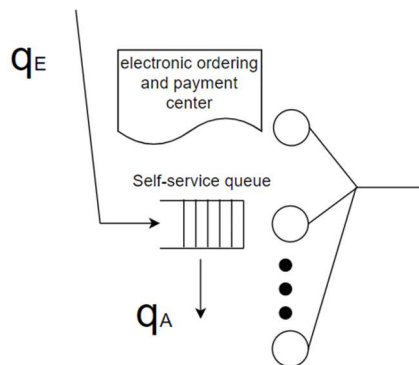
Quando una famiglia termina ordine e pagamento, si accoda per ricevere il cibo nell'apposita coda delle famiglie e verrà servita senza attesa se la coda del McDrive è *empty* oppure in quell'istante un'automobile sta ricevendo il pasto, e se la coda delle famiglie è *empty* e almeno un servente è *idle*; in caso contrario, si mette in attesa nell'apposita coda delle famiglie. Inoltre, quando una famiglia riceve il pasto, il servente passa allo stato *idle* se la coda del McDrive e quella delle famiglie sono entrambe *empty* oppure se la coda delle famiglie è vuota e c'è un altro servente nello stato *busy_a*; invece, serve un'automobile se la coda del McDrive è *not empty* e non ci sono altri serventi nello stato *busy_a*; infine, serve una famiglia se la coda del McDrive è *empty* e quella della famiglia è *not empty* oppure se la coda della famiglia è *not empty* e un altro servente è nello stato *busy_a*.



I job, una volta usciti da questo centro, possono abbandonare il sistema oppure dirigersi verso il centro relativo alla consumazione (consumption center).

Quando arriva una famiglia alle casse automatiche, se c'è almeno un servente (i.e. una cassa automatica) *idle*, essa prende subito servizio, altrimenti attende in coda. Quando una famiglia termina ordine e pagamento alle casse automatiche, si accoda per ricevere il pasto nell'apposita coda per le famiglie, mentre la cassa automatica serve la famiglia in testa alla coda se la coda è *not empty*, diventa *idle* altrimenti. Anche

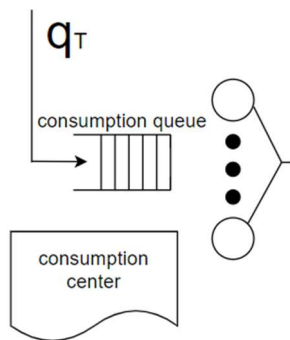
qui, se una famiglia rimane troppo tempo in attesa nella coda relativa alle casse automatiche, può abbandonare il ristorante.



Electronic ordering and payment center è, alternativamente a ordering and payment center, il primo centro visitato dai job.

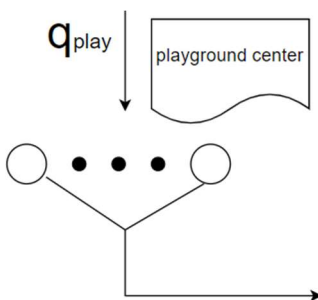
I job, una volta usciti da questo centro, si dirigono verso il centro relativo al ritiro del cibo (withdraw food center), immettendosi nella coda delle famiglie.

Dopo aver ricevuto il pasto, la famiglia può decidere se consumare a casa (i.e. uscire dal sistema) oppure al tavolo. Quando una famiglia arriva all'area consumazione, può sedersi senza attesa se esiste almeno un tavolo libero (i.e. servente *idle*); in caso contrario, dovrà mettersi in attesa all'interno della coda. Quando una famiglia termina il consumo, il tavolo diventa libero (*idle*) se la coda è *empty*; altrimenti il tavolo verrà occupato dalla famiglia in testa alla coda.



I job, una volta usciti da questo centro, possono abbandonare il sistema oppure dirigersi verso il centro relativo all'area giochi (playground center).

Quando una famiglia termina il consumo, può dirigersi verso l'area giochi oppure abbandonare il sistema. Quando una famiglia (con un bambino) arriva all'area giochi, se c'è almeno un posto disponibile (i.e. servente *idle*), il bambino può entrare a giocare, altrimenti la famiglia esce dal sistema senza attendere. Quando il bambino ha finito di giocare, il posto che aveva occupato passa dallo stato *busy* allo stato *idle* e la famiglia esce dal sistema. Per semplicità, l'assunzione che è stata fatta nella modellazione del sistema è che una famiglia può occupare al più un posto anche nell'area giochi.



I job, una volta usciti da questo centro, abbandonano il sistema.

Profitto del sistema.

In questa sezione vengono descritti i ricavi e i costi del modello che concorrono nel determinare il profitto del ristorante.

Per motivi logistici, un'automobile che entra all'interno del McDrive non può uscire dal sistema senza essere servita. Di conseguenza, tutte le automobili che entrano nel sistema riceveranno un servizio e apporteranno un ricavo. Invece, le famiglie che entrano nel sistema potrebbero abbandonare mentre sono in coda per effettuare l'ordine e il pagamento se l'attesa è eccessivamente lunga: queste famiglie non apporteranno alcun ricavo. Le famiglie che invece completano l'ordine e il pagamento apporteranno un ricavo. Nel modello si è assunto che la dimensione media di una famiglia nel sistema sia pari a tre persone con un profitto pari a 20€ per ogni famiglia (i.e. per ogni pasto).

$$\begin{aligned}
 T &= \text{tempo di riferimento (in mesi)} \\
 N &= \text{numero di job serviti con successo nel tempo di riferimento} \\
 R &= \text{ricavo per ogni job} \\
 C_{OP} &= \text{costo mensile per i dipendenti che lavorano per ordinazioni e pagamenti} \\
 C_{food} &= \text{costo mensile per i dipendenti che servono il cibo} \\
 C_E &= \text{costo operativo mensile delle casse automatiche (self-service)} \\
 C_T &= \text{costo mensile per la manutenzione dei tavoli} \\
 C_{play} &= \text{costo mensile per l'affitto dell'area giochi} \\
 P &= \text{profitto nel tempo di riferimento}
 \end{aligned}$$

$$P = N * R - (C_{OP} + C_{food} + C_E + C_T + C_{play}) * T$$

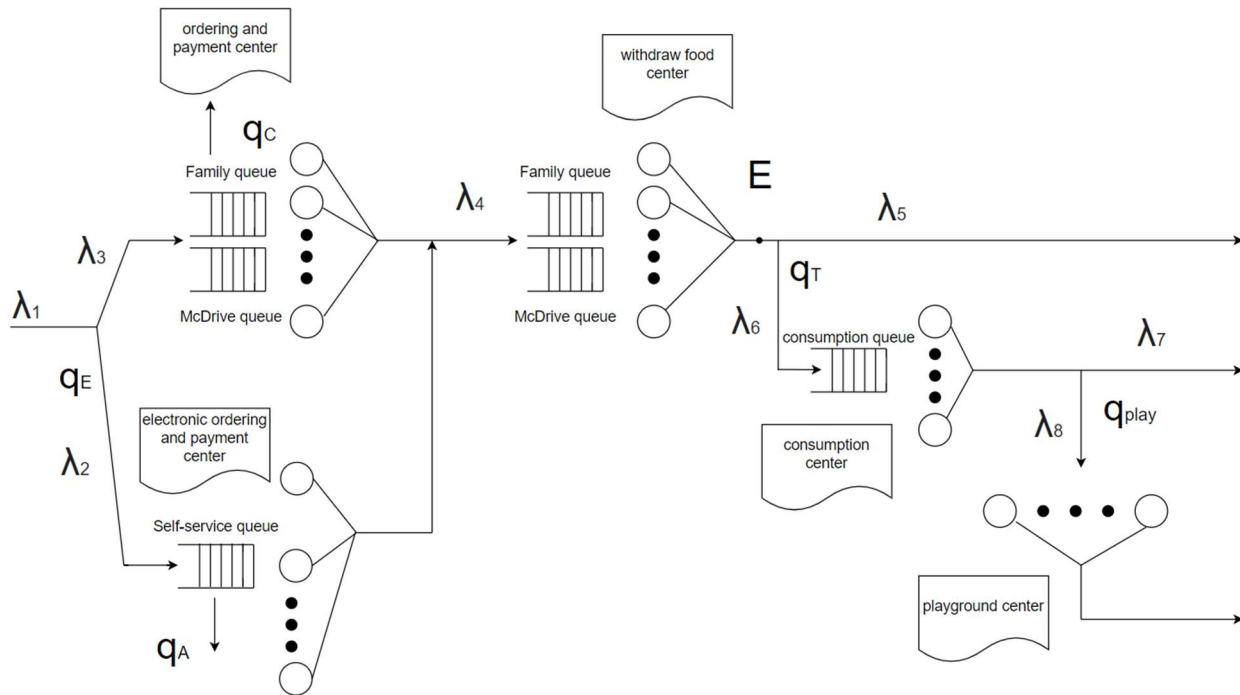
I serventi del centro *ordering and payment* sono dei dipendenti che vengono pagati 8€ l'ora. Anche i serventi del centro *withdraw food* sono dei dipendenti che vengono pagati 8€ l'ora. I serventi del centro *electronic ordering and payment* sono dispositivi elettronici che hanno un costo operativo pari a 1€ all'ora cadauno. I serventi del centro *consumption* sono tavoli che hanno un costo di manutenzione pari a 5€ al mese per ogni tavolo. I serventi del centro *playground* sono i posti disponibili per giocare, che hanno un costo di affitto di 10€/mq al mese. Nel modello si è assunto di assegnare un posto libero (i.e. un servente) ogni 3 mq.

$$\begin{aligned}
 A &= \text{superficie dell'area giochi (in metri quadrati)} \\
 S &= \text{numero di serventi} \\
 Q &= \text{superficie richiesta per ogni servente} = 3 \text{ mq} \\
 K &= \text{costo mensile di affitto per l'area giochi al metro quadrato} = 10€ \\
 C_{play} &= \text{costo mensile di affitto per l'area giochi}
 \end{aligned}$$

$$\begin{aligned}
 S &= \text{floor}(A/Q) \\
 C_{play} &= A * K
 \end{aligned}$$

Il profitto del ristorante è dato dalla differenza tra la somma dei ricavi e la somma dei costi, che verranno calcolati su un intervallo di tempo prefissato.

Modello delle specifiche.



$$\lambda_1 = \lambda_{MD} + \lambda_{FF} + \lambda_{FE}$$

$$\lambda_2 = \lambda_{FE}$$

$$\lambda_3 = \lambda_{MD} + \lambda_{FF}$$

$$\lambda_4 = \lambda_{MD} + \lambda_{FF} * (1 - q_c) + \lambda_{FE} * (1 - q_A)$$

$$\lambda_5 = \lambda_{MD} + [\lambda_{FF} * (1 - q_c) + \lambda_{FE} * (1 - q_A)] * (1 - q_T)$$

$$\lambda_6 = [\lambda_{FF} * (1 - q_c) + \lambda_{FE} * (1 - q_A)] * (q_T)$$

$$\lambda_7 = [\lambda_{FF} * (1 - q_c) + \lambda_{FE} * (1 - q_A)] * (q_T) * (1 - q_{play})$$

$$\lambda_8 = [\lambda_{FF} * (1 - q_c) + \lambda_{FE} * (1 - q_A)] * (q_T) * (q_{play})$$

λ_{MD} = tasso di arrivo alla coda del McDrive del centro ordering and payment

λ_{FF} = tasso di arrivo alla coda delle famiglie del centro ordering and payment

λ_{FE} = tasso di arrivo nel centro electronic ordering and payment

Modellazione dei centri.

Disclaimer

Purtroppo non sono disponibili in rete dei dati o delle statistiche riguardanti il numero di clienti che accedono al ristorante a pranzo, a cena o comunque nell'arco di una giornata, né tantomeno è possibile reperire informazioni riguardanti il tempo impiegato da un dipendente per servire una famiglia, il tempo medio trascorso da una famiglia nell'area consumazione e dati simili.

D'altra parte, il McDonald's è un locale molto frequentato anche da noi, per cui abbiamo avuto la possibilità di osservare quante automobili arrivano al McDrive, quante persone si trovano all'interno del ristorante al variare dell'orario, quanto tempo è richiesto a un dipendente per prendere un ordine e poi per servire il cibo, quante famiglie portano il cibo a casa piuttosto che consumarlo all'interno del ristorante, quanto tempo mediamente una famiglia rimane seduta al tavolo, quante famiglie portano con sé dei bambini che

vogliono accedere all'area giochi e così via.

Per queste ragioni, i valori numerici relativi alle informazioni sopra citate sono dati dalle nostre esperienze personali e sono stati pensati per risultare i più realistici possibile.

Centro ordering and payment (centro M/M/m con due classi di priorità)

- Arrivi Markoviani nella coda del McDrive (tasso λ_{MD})
- Arrivi Markoviani nella coda delle famiglie (tasso λ_{FF}) $\rightarrow \lambda_{MD} + \lambda_{FF} = \lambda_3$
- La possibilità di perdita di un job J dalla coda delle famiglie viene introdotta nel momento in cui davanti a lui in coda ci sono più di N job; in tal caso, la perdita viene modellata con un tempo esponenziale di parametro λ_{loss} ma, nel momento in cui i job davanti a J in coda diventano al più N, la probabilità di perdita si azzera.
- Tempi di servizio esponenziali (media $E[S_i] = 1,5$ minuti)

Centro electronic ordering and payment (centro M/M/m)

- Arrivi Markoviani (tasso $\lambda_{FE} = \lambda_2$)
- La possibilità di perdita di un job J viene introdotta nel momento in cui davanti a lui in coda ci sono più di N job; in tal caso, la perdita viene modellata con un tempo esponenziale di parametro λ_{loss} ma, nel momento in cui i job davanti a J in coda diventano al più N, la probabilità di perdita si azzera.
- Tempi di servizio esponenziali (media $E[S_i] = 3$ minuti)

Centro withdraw food (centro M/M/m con due classi di priorità)

- Arrivi Markoviani nella coda del McDrive (tasso λ_{4A})
- Arrivi Markoviani nella coda delle famiglie (tasso λ_{4F}) $\rightarrow \lambda_{4A} + \lambda_{4F} = \lambda_4$
- Tempi di servizio esponenziali (media $E[S_i] = 4$ minuti)

Centro consupion (centro M/M/m)

- Arrivi Markoviani (tasso λ_6)
- Tempi di servizio esponenziali (media $E[S_i] = 30$ minuti)

Centro playground (centro M/M/m/m)

- Arrivi Markoviani (tasso λ_8)
- Tempi di servizio esponenziali (media $E[S_i] = 30$ minuti)

Per quanto riguarda le perdite che possono avvenire nei primi due centri (*ordering and payment*, *electronic ordering and payment*), sono stati selezionati i seguenti valori per i parametri λ_{loss} e N:

$$\rightarrow \lambda_{loss} = 1/360 \text{ sec}^{-1}$$

$$\rightarrow N = 3$$

Di fatto, quando una famiglia entra all'interno del ristorante, guarda quante altre famiglie ha davanti in coda per ordinare il pasto. Se sono numerose (dove, per numerose, è possibile intendere più di 3), allora valuta se la coda viene smaltita in fretta o meno: se viene smaltita in fretta, allora è lecito assumere che la famiglia non si spazientisca e rimanga in coda finché non prenderà servizio; in caso contrario, è possibile che la famiglia abbandoni il centro senza ordinare il pasto, inficiando così sul guadagno del ristorante. Per questo motivo, se la famiglia vede più di N altre famiglie davanti a lei in coda, potrebbe abbandonare il ristorante dopo un tempo medio pari a $1/\lambda_{loss} = 360$ secondi (i.e. 6 minuti), a meno che prima di questo tempo le famiglie davanti in coda non siano diventate sufficientemente poche (ovvero al più 3).

Matrice di routing.

	EST	1	2	3	4	5
EST	0	$1-q_E$	q_E	0	0	0
1	q_C	0	0	$1-q_C$	0	0
2	q_A	0	0	$1-q_A$	0	0
3	$1-q_T$	0	0	0	q_T	0
4	$1-q_{play}$	0	0	0	0	q_{play}
5	1	0	0	0	0	0

EST = esterno
 1 = centro ordering and payment
 2 = centro electronic ordering and payment
 3 = centro withdraw food
 4 = centro consumption
 5 = centro playground

p_{ij} = probabilità di andare al centro j a partire dal centro i

q_E = probabilità di entrare nel sistema accedendo al centro electronic ordering+payment

$1-q_E$ = probabilità di entrare nel sistema accedendo al centro ordering+payment

q_C = probabilità di perdita di una famiglia nel centro ordering and payment

q_A = probabilità di perdita nel centro electronic ordering and payment

q_T = probabilità per una famiglia di accedere al centro consumption a partire dal centro withdraw food

$1-q_T$ = probabilità per una famiglia di uscire dal sistema a partire dal centro withdraw food

q_{play} = probabilità di accedere al centro playground a partire dal centro consumption

$1-q_{play}$ = probabilità di uscire dal sistema a partire dal centro consumption

In base alle nostre osservazioni e alle nostre esperienze personali, i componenti della matrice di routing sono stati scelti nel seguente modo:

-> $q_E = 0.6$

-> $q_T = 0.7$

-> $q_{play} = 0.3$

Fasce orarie.

Poiché un McDonald's di norma è aperto tutti i giorni dalle 9:00 alle 23:00, sono state identificate le seguenti fasce orarie:

1° slot – 09:00 → 11:00 – tasso di arrivo: $\lambda_1 = 1/3$ job/min

2° slot – 11:00 → 12:00 – tasso di arrivo: $\lambda_1 = 1/8$ job/min

3° slot – 12:00 → 15:00 – tasso di arrivo: $\lambda_1 = 1$ job/min

4° slot – 15:00 → 18:00 – tasso di arrivo: $\lambda_1 = 1/5$ job/min

5° slot – 18:00 → 22:00 – tasso di arrivo: $\lambda_1 = 2$ job/min

6° slot – 22:00 → 23:00 – tasso di arrivo: $\lambda_1 = 1/3$ job/min

Proporzione tra numero di automobili e numero di famiglie.

Una volta identificato il valore di λ_1 dipendentemente dalla fascia oraria, è necessario definire anche P_{car} , ovvero la percentuale di automobili su tutti i job che entrano nel sistema. In base alle osservazioni che abbiamo potuto fare visitando il McDonald's, è possibile constatare che le automobili costituiscano circa il 10% dei clienti del ristorante; per questo motivo, viene posto $P_{car} = 0,1$.

Poiché q_E era stato definito come la percentuale di job che entra nel sistema accedendo al centro *electronic ordering and payment*, abbiamo che:

$$q_E + P_{car} + P_{fam_ph} = 1 \rightarrow P_{fam_ph} = 1 - (q_E + P_{car})$$

dove P_{fam_ph} è uguale alla percentuale di job che costituiscono una famiglia che entra nel sistema accedendo alle casse fisiche (centro *ordering and payment*).

In conclusione:

$$\rightarrow P(\text{essere una famiglia}) = 1 - P_{car} = q_E + P_{fam_ph}$$

$$\rightarrow P(\text{accedere al centro } ordering \text{ and payment}) = 1 - q_E = P_{car} + P_{fam_ph}$$

Algoritmi per la Next-Event simulation.

CENTRO ORDERING AND PAYMENT

Variabili di stato

- q_A = popolazione nella coda McDrive
- q_F = popolazione nella coda famiglie
- $x_i \in \{idle, busy_a, busy_f\} \forall i$

Evento 1 (arrivo di un'automobile)

```

 $t_{aA\_OP} := GetArrival()$ 
if ( $t_{aA\_OP} > \tau$ ) then
     $t_{aA\_OP} := \infty$ 
endif
if ( $\nexists x_k = busy\_a \wedge q_A = 0 \wedge \exists x_i = idle$ ) then
     $x_i := busy\_a$ 
     $t_{ci} := t + GetService()$ 
else
     $q_A++$ 
endif

```

Evento 2 (arrivo di una famiglia)

```

 $t_{aF\_OP} := GetArrival()$ 
if ( $t_{aF\_OP} > \tau$ ) then
     $t_{aF\_OP} := \infty$ 
endif
if ( $(q_A = 0 \vee \exists x_k = busy\_a) \wedge q_F = 0 \wedge \exists x_i = idle$ ) then
     $x_i := busy\_f$ 
     $t_{ci} := t + GetService()$ 
else
    if ( $q_F > N$ ) then
         $t_{p\_tail} := t + GetAbandon()$ 
         $abandonsList.append(t_{p\_tail})$ 
    endif
     $q_F++$ 
endif

```

Evento 3 (completamento di un'automobile)

```

if ( $q_A \neq 0$ ) then
   $t_{ci} := t + \text{GetService}()$ 
   $q_A--$ 
else if ( $q_F \neq 0$ ) then
   $t_{ci} := t + \text{GetService}()$ 
   $q_F--$ 
   $x_i := \text{busy}_f$ 
  //Lasciamo la possibilità ai job di abbandonare solo se hanno almeno N job avanti.
  if ( $t_{p\_head} \neq \infty \wedge t_{p\_head} \neq \text{null}$ ) then
     $t_{p\_head} := \infty$ 
     $\text{abandonsList.remove}(t_{p\_head})$ 
  endif
else
   $t_{ci} := \infty$ 
   $x_i := \text{idle}$ 
endif
 $\text{arrivalsMcDriveWithdrawFoodList.append}(t)$ 

```

Evento 4 (completamento di una famiglia)

```

if ( $\nexists x_k = \text{busy}_a \wedge q_A \neq 0$ ) then
   $t_{ci} := t + \text{GetService}()$ 
   $q_A--$ 
   $x_i := \text{busy}_a$ 
else if ( $(q_A = 0 \wedge q_F \neq 0) \vee (q_F \neq 0 \wedge \exists x_k = \text{busy}_a)$ ) then
   $t_{ci} := t + \text{GetService}()$ 
   $q_F--$ 
  if ( $t_{p\_head} \neq \infty \wedge t_{p\_head} \neq \text{null}$ ) then
     $t_{p\_head} := \infty$ 
     $\text{abandonsList.remove}(t_{p\_head})$ 
  endif
else
   $t_{ci} := \infty$ 
   $x_i := \text{idle}$ 
endif
 $\text{arrivalsFamilyWithdrawFoodList.append}(t)$ 

```

Evento 5 (abbandono di una famiglia)

```

 $\text{abandonsList.remove}(tPx)$ 
 $q_F--$ 

```

CENTRO ELECTRONIC ORDERING AND PAYMENT**Variabili di stato**

- l = popolazione nel centro
- $x_i \in \{idle, busy\} \forall i$

Evento 1 (arrivo)

```

l++
tae := GetArrival()
if (tae >  $\tau$ ) then
    tae :=  $\infty$ 
endif
if ( $\exists x_i = idle$ ) then
    xi := busy
    tci := t + GetService()
else if (l > N+m) then
    tp_tail := t + GetAbandon()
    abandonsList.append(tp_tail)
endif

```

Evento 2 (completamento)

```

l--
if (tp_head !=  $\infty$   $\wedge$  tp_head != null) then
    tp_head :=  $\infty$ 
    abandonsList.remove(tp_head)
endif
if (l >= num_serventi) then
    tci := t + GetService()
else
    tci :=  $\infty$ 
    xi := idle
endif
arrivalsFamilyWithdrawFoodList.append(t)

```

Evento 3 (abbandono)

```

abandonsList.remove(tpx)
l--

```

CENTRO WITHDRAW FOOD**Variabili di stato**

- q_A = popolazione nella coda McDrive
- q_F = popolazione nella coda famiglie
- $x_i \in \{idle, busy_a, busy_f\} \forall i$

Evento 1 (arrivo di un'automobile)

```

 $t_{aA\_food} := arrivalsMcDriveWithdrawFoodList.pop()$ 
if ( $\nexists x_k = busy\_a \wedge q_A = 0 \wedge \exists x_i = idle$ ) then
   $x_i := busy\_a$ 
   $t_{ci} := t + GetService()$ 
else
   $q_A++$ 
endif

```

Evento 2 (arrivo di una famiglia)

```

 $t_{aF\_food} := arrivalsFamilyWithdrawFoodList.pop()$ 
if ( $(q_A = 0 \vee \exists x_k = busy\_a) \wedge q_F = 0 \wedge \exists x_i = idle$ ) then
   $x_i := busy\_f$ 
   $t_{ci} := t + GetService()$ 
else
   $q_F++$ 
endif

```

Evento 3 (completamento di un'automobile)

```

if ( $q_A \neq 0$ ) then
   $t_{ci} := t + GetService()$ 
   $q_A--$ 
else if ( $q_F \neq 0$ ) then
   $t_{ci} := t + GetService()$ 
   $q_F--$ 
   $x_i := busy\_f$ 
else
   $t_{ci} := \infty$ 
   $x_i := idle$ 
endif

```

Evento 4 (completamento di una famiglia)

```

if ( $\nexists x_k = busy\_a \wedge q_A \neq 0$ ) then
   $t_{ci} := t + GetService()$ 
   $q_A--$ 
   $x_i := busy\_a$ 
else if ( $(q_A = 0 \wedge q_F \neq 0) \vee (q_F \neq 0 \wedge \exists x_k = busy\_a)$ ) then
   $t_{ci} := t + GetService()$ 
   $q_F--$ 
else
   $t_{ci} := \infty$ 
   $x_i := idle$ 
endif
if ( $Random() < q_\tau$ ) then
   $arrivalsConsumptionList.append(t)$ 
endif

```


CENTRO CONSUPTION**Variabili di stato**

- l = popolazione nel centro
- $x_i \in \{idle, busy\} \forall i$

Evento 1 (arrivo)

```

l++
tQT := arrivalsConsumptionList.pop()
if (( $\exists x_i = idle$ ) then
  xi := busy
  tci := t + GetService()
endif

```

Evento 2 (completamento)

```

l--
if (l >= num_serventi) then
  tci := t + GetService()
else
  tci :=  $\infty$ 
  xi := idle
endif
if (Random() < qplay) then
  arrivalsPlaygroundList.append(t)
endif

```

CENTRO PLAYGROUND**Variabili di stato**

- $x_i \in \{idle, busy\} \forall i$

Evento 1 (arrivo)

```

ta_play := arrivalsPlaygroundList.pop()
if (( $\exists x_i = idle$ ) then
  xi := busy
  tci := t + GetService()
endif

```

Evento 2 (completamento)

```

tci :=  $\infty$ 
xi := idle

```

Modello computazionale.

Stato del sistema.

Nel modello computazionale, lo stato del sistema viene modellato sfruttando le strutture dati offerte dal linguaggio C. I centri del sistema sono stati classificati in due diversi gruppi, ognuno dei quali ha la propria struttura dati che modella lo stato. Le due strutture dati utilizzate sono:

```
struct state_variables1 {
    int qA;
    int qF;
    int *x;
};
```

```
struct state_variables2 {
    int l;
    int *x;
};
```

La struttura *struct state_variables1* è relativa ai centri *ordering and payment* e *withdraw food*. Di fatto, sono i centri che hanno due code distinte (per le automobili e per le famiglie): per caratterizzare in modo esaustivo il loro stato è possibile tenere traccia della popolazione all'interno della coda del McDrive (campo *qA* della struttura), la popolazione all'interno della coda delle famiglie (campo *qF* della struttura) e lo stato di ciascun server (campo *x* della struttura); in particolare, *x* è un array di *m* valori (dove *m* è il numero dei server nel centro di riferimento) ciascuno dei quali indica se il server corrispondente è nello stato *idle* (valore 0), *busy_f* (valore 1) oppure *busy_a* (valore 2).

La struttura *struct state_variables2*, invece, è relativa ai centri *electronic ordering and payment*, *consumption* e *playground*. Questi sono i tre centri con un'unica coda, per cui, per rappresentare in modo esaustivo il loro stato, è sufficiente tenere traccia della popolazione totale all'interno del centro (campo *l* della struttura) e lo stato di ciascun server (campo *x* della struttura); qui *x* è lo stesso array del caso di *struct state_variables1*, con l'unica differenza che le *m* entry di *x* possono assumere esclusivamente i valori 0 (server nello stato *idle*) e 1 (server nello stato *busy_f*).

Una precisazione da fare è che il centro *playground*, benché utilizzi la struttura *struct state_variables2*, non fa uso del campo *l*: essendo un centro M/M/m/m, non ammette coda, per cui il suo stato può essere caratterizzato in modo esaustivo solo tramite lo stato dei singoli server.

L'utilizzo delle fasce orarie ha reso l'implementazione più complessa. Nella transizione da una fascia oraria alla successiva si deve tener conto dello stato attuale dei server che verranno disabilitati. Nel momento in cui il server *k* sta offrendo servizio a un job e nella nuova fascia oraria non sarà più attivo, esso dovrà terminare il servizio del job corrente prima di disattivarsi. Di conseguenza, per poter eseguire la simulazione a orizzonte finito (di cui si discuterà successivamente), è stato necessario introdurre tre ulteriori valori possibili per l'array *x* che tengano conto di questi possibili scenari:

- 1) Nel cambio di fascia, il server sta processando un job appartenente alla classe delle automobili: in questo caso, a livello implementativo ciò è stato modellato assegnando come valore della corrispettiva entry di *x* il valore -2.
- 2) Nel cambio di fascia, il server sta processando un job appartenente alla classe delle famiglie: in questo caso, a livello implementativo ciò è stato modellato assegnando come valore della corrispettiva entry di *x* il valore -1.
- 3) Nel cambio di fascia, se il server è *idle*, allora non dovrà terminare alcun servizio. A livello implementativo, ciò è stato modellato assegnando come valore della corrispettiva entry di *x* il valore -3.

Se in un certo istante di tempo esiste una entry *x[k]* di *x* che assume un valore tra -1 e -2, appena il server corrispondente avrà terminato il servizio, a *x[k]* verrà assegnato il valore -3.

Eventi del sistema.

Per effettuare la simulazione del sistema, è stato utilizzato l'approccio della *Next-Event Simulation*. La struttura dati che mantiene gli eventi del sistema è riportata qui di seguito:

```

struct event_list {
    struct car_arrival carArr1;
    struct family_arrival familyArr1;
    double *completionTimes1;
    struct job *head1;
    struct job *tail1;
    struct family_arrival familyArr2;
    double *completionTimes2;
    struct job *head2;
    struct job *tail2;
    struct car_arrival carArr3;
    struct family_arrival familyArr3;
    double *completionTimes3;
    struct family_arrival familyArr4;
    double *completionTimes4;
    struct family_arrival familyArr5;
    double *completionTimes5;
    double changeInterval;
    double sampling;
};

```

Gli eventi rappresentati da tale struttura dati sono i seguenti:

- Arrivo della prossima automobile all'interno del centro *ordering and payment* con il relativo tempo di arrivo; viene descritto all'interno della struttura *struct car_arrival carArr1*, che verrà riportata di seguito.
- Arrivo della prossima automobile all'interno del centro *withdraw food* con il relativo tempo di arrivo; viene descritto all'interno della struttura *struct car_arrival carArr3*.
- Arrivo della prossima famiglia all'interno di un determinato centro K con il relativo tempo di arrivo; viene descritto all'interno della struttura *struct family_arrival familyArrK*, che verrà riportata di seguito.
- Completamento di ogni servente di ciascuno dei cinque centri del sistema, dove il centro K ha associato l'array di double *completionTimesK*: ogni valore di tale array indica il tempo di completamento del relativo servente.
- Abbandono di ogni famiglia dai centri *ordering and payment* ed *electronic ordering and payment*, che viene rappresentato mediante due liste doppiamente collegate (una per ogni centro). Ciascuna lista doppiamente collegata contiene i nodi (di tipo *struct job*) relativi alle famiglie che hanno la possibilità di abbandonare il sistema (ovvero le famiglie con più di N=3 altre famiglie davanti in coda); questi nodi tengono traccia di un identificatore della famiglia e del tempo di abbandono che è stato schedato. L'ordine della lista è dato dall'ordine di arrivo delle famiglie all'interno del centro: per questo motivo, nel momento in cui una nuova famiglia prende servizio, l'elemento in testa alla lista è ora relativo alla famiglia che ha esattamente 3 famiglie davanti (mentre prima erano 4), ovvero alla famiglia che esclude l'idea di abbandonare il centro; di conseguenza, per ogni nuova famiglia che prende servizio, è sufficiente effettuare una rimozione dalla testa della lista, mentre gli inserimenti all'interno della lista avvengono esclusivamente in coda. Ovviamente, se la rimozione di un nodo dalla lista doppiamente collegata non avviene prima dell'istante di abbandono, vuol dire che la famiglia associata a quel nodo abbandona il sistema. Per facilitare le operazioni di inserimento e di rimozione dalla lista doppiamente collegata, all'interno di *struct event_list* sono stati riportati sia il nodo coda (*struct job *tailK*) che il nodo testa (*struct job *headK*).
- Altri due eventi (identificati dai campi *changeInterval* e *sampling* della struttura) che verranno discussi nella descrizione della simulazione a orizzonte finito.

```

struct family_arrival {
    double familyArrivalTime;
    bool isFamilyArrivalActive;
};

```

```

struct car_arrival {
    double carArrivalTime;
    bool isCarArrivalActive;
};

```

```

struct job {
    int id;
    double abandonTime;
    struct job *prev;
    struct job *next;
};

```

Da queste ultime figure è possibile notare che, per quanto concerne gli arrivi delle famiglie e delle automobili nei centri *ordering and payment* ed *electronic ordering and payment*, viene specificato anche se tali arrivi sono abilitati o meno: in particolare, risultano disabilitati nel momento in cui il clock supera l'istante di STOP (che indica la fine della simulazione dell'orizzonte finito).

Politica di selezione del servente *idle*.

Nel momento in cui un job deve prendere servizio, se sono presenti più serventi nello stato *idle*, viene scelto sempre quello con ID più piccolo, ovvero quello con offset minore all'interno dell'array di serventi.

PRNG utilizzato.

Per eseguire le simulazioni, è stato sfruttato lo Pseudo-Random Number Generator della libreria *rngs.c*.

Verifica.

Nel processo di verifica, si è utilizzata la configurazione (1, 3, 1, 3, 3) e la seconda fascia oraria. In questo modo, grazie a una bassa utilizzazione dei centri, abbiamo evitato che all'interno dei centri *ordering and payment* ed *electronic ordering and payment* si verificassero delle perdite. Di conseguenza, la simulazione è risultata fedele al modello analitico che non tiene conto delle perdite.

Siccome nei centri *ordering and payment* e *withdraw food* il numero di serventi che possono essere impegnati nel servizio delle automobili è al più 1 nonostante la coda delle automobili possa non essere vuota, non è possibile utilizzare direttamente i modelli visti durante il corso. Per poter utilizzare tali modelli, per il processo di verifica abbiamo deciso di affrontare i seguenti due casi:

- 1) Il numero di serventi per i due centri è stato posto a 1 in modo da ricondurci a un modello di coda M/M/1 con due classi di priorità astratta.
- 2) Il valore di P_{car} è stato forzato a 0 in modo da eliminare la classe dei job *automobile* che portavano all'inconveniente per i calcoli analitici. In questo caso, l'unica classe dei job all'interno del sistema è quella delle famiglie.

Caso 1.

I risultati ottenuti utilizzando i modelli analitici per i cinque centri per quanto riguarda il tempo di attesa, il tempo di risposta, l'utilizzazione e la probabilità di perdita sono rappresentati all'interno della seguente tabella:

CENTRI	Tempo di attesa	Tempo di risposta	Utilizzazione	Prob. perdita
Ordering and payment	7,297320 s	97,297320 s	0,075	0
Electronic ordering and payment	0,106300 s	180,106300 s	0,075	0
Withdraw food	240 s	480 s	0,5	0
Consumption	1770,269760 s	3570,269760 s	0,7875	0
Playground	0 s	1800 s	0,229308	0,029386

Gli intervalli di confidenza, con livello di confidenza al $1-\alpha = 0,99$, prodotti dalle simulazioni sono riportati nella tabella di seguito:

CENTRI	Tempo di attesa	Tempo di risposta	Utilizzazione	Prob. perdita
Ordering and payment	7,160734 +/- 0,315252 s	96,666150 +/- 0,843778 s	0,074763 +/- 0,000863	0,0 +/- 0,0
Electronic ordering and payment	0,092438 +/- 0,025974 s	179,543591 +/- 1,347238 s	0,074836 +/- 0,000706	0,0 +/- 0,0

Withdraw food	239,364705 +/- 9,033057 s	479,401297 +/- 9,994784 s	0,499037 +/- 0,004589	0,0 +/- 0,0
Consumption	1696,582651 +/- 131,604903 s	3490,831469 +/- 137,585469 s	0,784275 +/- 0,007611	0,0 +/- 0,0
Playground	0,0 +/- 0,0 s	1797,899522 +/- 13,262890 s	0,229158 +/- 0,002261	0,028473 +/- 0,001274

Caso 2.

I risultati ottenuti utilizzando i modelli analitici per i cinque centri per quanto riguarda il tempo di attesa, il tempo di risposta, l'utilizzazione e la probabilità di perdita sono rappresentati all'interno della seguente tabella:

CENTRI	Tempo di attesa	Tempo di risposta	Utilizzazione	Prob. perdita
Ordering and payment	7,297320 s	97,297320 s	0,075	0
Electronic ordering and payment	0,106300 s	180,106300 s	0,075	0
Withdraw food	240 s	480 s	0,5	0
Consumption	3711,823680 s	5511,823680 s	0,875	0
Playground	0 s	1800 s	0,252694	0,037357

Gli intervalli di confidenza, con livello di confidenza al $1-\alpha = 0,99$, prodotti dalle simulazioni sono riportati nella tabella riportata di seguito:

CENTRI	Tempo di attesa	Tempo di risposta	Utilizzazione	Prob. perdita
Ordering and payment	7,250621 +/- 0,360189 s	96,756127 +/- 0,857716 s	0,074681 +/- 0,000802	0,0 +/- 0,0
Electronic ordering and payment	0,092438 +/- 0,025974 s	179,543591 +/- 1,347238 s	0,074836 +/- 0,000706	0,0 +/- 0,0
Withdraw food	238,820804 +/- 8,920195 s	478,856149 +/- 9,791838 s	0,499856 +/- 0,004854	0,0 +/- 0,0
Consumption	3349,336686 +/- 393,309827 s	5143,622981 +/- 398,153757 s	0,871087 +/- 0,007953	0,0 +/- 0,0
Playground	0,0 +/- 0,0 s	1797,884177 +/- 11,632935 s	0,252290 +/- 0,002369	0,036972 +/- 0,001690

Controlli di consistenza.

Nel processo di verifica sono stati effettuati i seguenti controlli di consistenza:

$$\rightarrow E[T_S] = E[T_Q] + E[S_i]$$

$$\rightarrow E[N_S] = E[N_Q] + m \cdot \rho$$

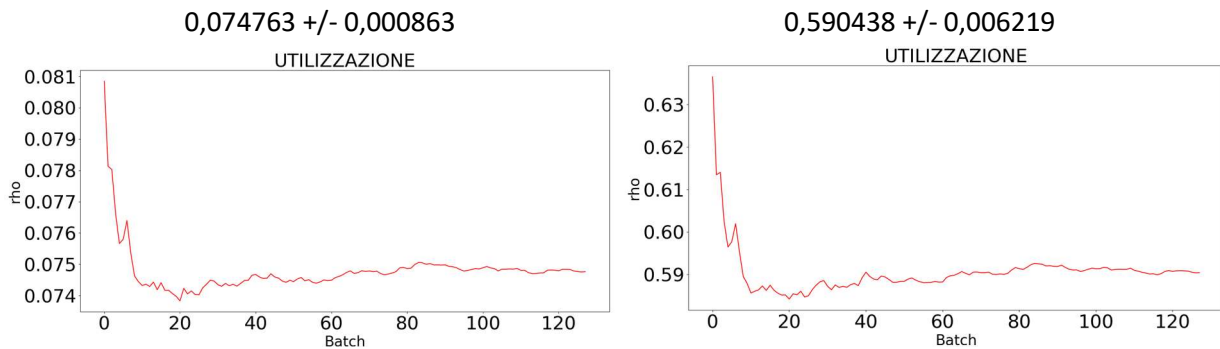
$$\rightarrow \rho \geq 0 \ \& \ \rho \leq 1 \ \& \ \text{prob. perdita} \geq 0 \ \& \ \text{prob. perdita} \leq 1$$

Per evitare problemi nell'approssimazione, non è stata verificata un'uguaglianza precisa, bensì è stato controllato se la differenza tra $E[T_S]$ e $E[T_Q] + E[S_i]$ e la differenza tra $E[N_S]$ e $E[N_Q] + m \cdot \rho$ fossero entrambe minori di 0,00001.

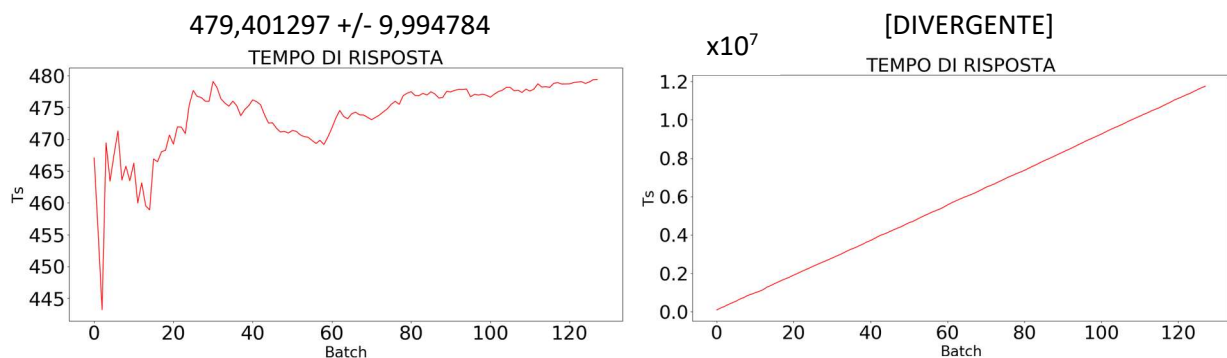
Validazione.

Poiché non è stato possibile recuperare dei dati relativi al nostro sistema con cui poter fare la validazione, abbiamo deciso di aggirare il problema controllando se il valore delle metriche varia al variare delle fasce orarie nel modo in cui ci aspettiamo. A parità di configurazione, nelle fasce orarie più affollate, le prestazioni del sistema dovrebbero peggiorare e viceversa. In particolare, abbiamo studiato il valore delle metriche di prestazione durante la seconda e la terza fascia oraria per i centri del sistema. In seguito, sono mostrati gli intervalli di confidenza e i grafici che dimostrano come le prestazioni dei centri peggiorino mantenendo sempre la configurazione (1, 3, 1, 3, 3).

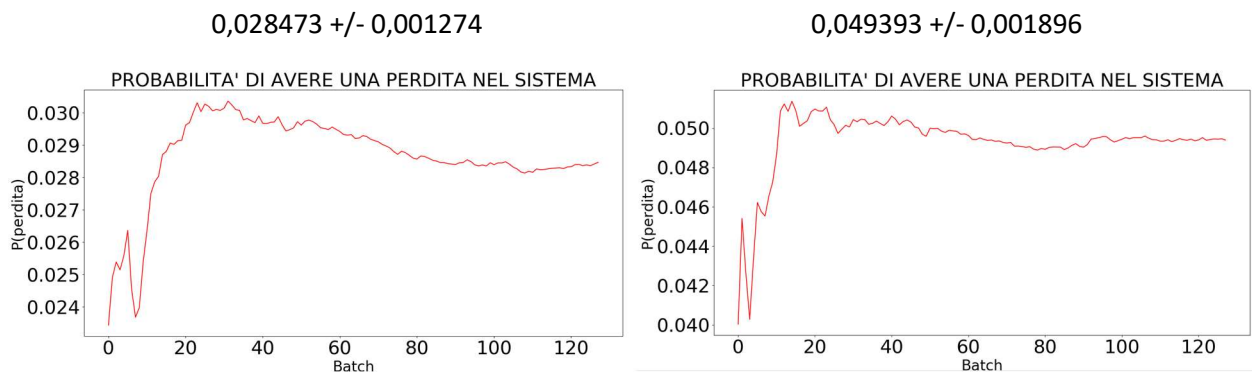
Utilizzazione del centro ordering and payment nella seconda (sx) e nella terza (dx) fascia oraria:



Tempo di risposta del centro withdraw food nella seconda (sx) e nella terza (dx) fascia oraria:



Probabilità di perdita nel centro playground nella seconda (sx) e nella terza (dx) fascia oraria:



Progettazione degli esperimenti.

La progettazione degli esperimenti si articola in tre fasi principali:

- 1) Analisi del collo di bottiglia.
- 2) Esecuzione delle simulazioni a orizzonte infinito.
- 3) Esecuzione delle simulazioni a orizzonte finito.

L'analisi del collo di bottiglia consiste nel calcolare con l'ausilio dell'analisi operativa la domanda media per ciascun centro del sistema, in modo tale da identificare i centri con la domanda più alta che, quindi, rappresentano il collo di bottiglia.

La simulazione a orizzonte infinito prevede che il sistema venga simulato per un tempo molto superiore a quello reale, ed è stata eseguita considerando le singole fasce orarie (e non l'intera giornata di lavoro), con tasso di arrivo e configurazione costanti. Inoltre, questo tipo di simulazione viene eseguito col metodo delle Batch Means, dove la run è stata suddivisa in $k=128$ batch di dimensione $b=1024$ job.

Nella simulazione a orizzonte infinito, per le singole fasce orarie, abbiamo determinato la configurazione ottimale concentrandoci sui QoS. Tuttavia, questa configurazione potrebbe non determinare il massimo profitto del ristorante. Infatti:

-> Non essendoci dei QoS relativi ai centri *ordering and payment* ed *electronic ordering and payment* che tengano conto della perdita dei clienti (fattore fondamentale per determinare il profitto totale del ristorante), è possibile che la configurazione finale di questi centri ottenuta a seguito dell'esecuzione delle simulazioni a orizzonte infinito non ottimizzi il profitto.

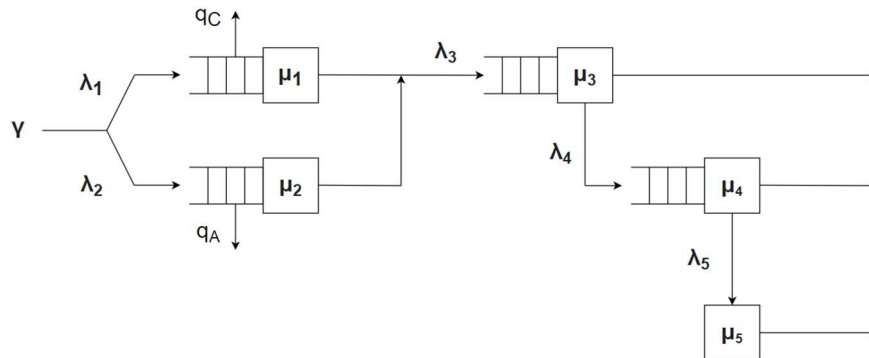
-> I risultati dati dalle simulazioni a orizzonte infinito non considerano l'esecuzione del sistema per l'intera giornata. Non sarebbe strano se fosse possibile diminuire il numero di server in qualche centro (non rispettando più i vincoli prestazionali per le fasce orarie più affollate) mantenendo comunque realizzati i requisiti QoS relativi alla giornata nel suo complesso.

La simulazione a orizzonte finito, infine, prevede che il sistema venga simulato sulle 14 ore lavorative. Di conseguenza, qui sono state considerate, in ciascun cambio di fascia oraria, la variazione della frequenza di arrivo e la riorganizzazione del numero di server attivi per ogni centro. Questo tipo di simulazione viene eseguito col metodo delle *Replicazioni*, dove la run è stata eseguita 128 volte in modo tale da poterne derivare delle statistiche mediate sulle 128 repliche. Inoltre, per le simulazioni è stato scelto uno stato iniziale corrispondente al sistema vuoto.

Dal punto di vista implementativo, sono stati introdotti due nuovi campi all'interno della struttura dati relativa agli eventi: *changeInterval* e *sampling*. Il primo corrisponde all'evento "cambio di fascia oraria", mentre il secondo corrisponde alla misurazione delle statistiche di output, che avviene ogni 300 secondi e consente dunque di studiare l'andamento del sistema nel transiente mediato sulle 128 repliche.

Nella simulazione a orizzonte finito ci siamo posti l'obiettivo di massimizzare il profitto del ristorante considerando come punto di partenza le configurazioni ottenute dalle simulazioni a orizzonte infinito. Poiché i centri soggetti a potenziali perdite sono *ordering and payment* ed *electronic ordering and payment*, ci aspettiamo che il numero di server possa dover essere incrementato per massimizzare il profitto. A questo punto, è possibile che si verifichi un incremento del tasso di arrivo al terzo centro con conseguente potenziale aumento del tasso di arrivo al quarto e al quinto centro. Di conseguenza, il numero di server per il quarto e il quinto centro potrebbe dover essere aumentato per rispettare i requisiti di QoS.

Analisi del collo di bottiglia.



λ_i = tasso di ingresso nel centro i

γ = tasso di ingresso nel sistema

Servizio medio dei centri del sistema.

Per l'analisi del collo di bottiglia abbiamo impostato un unico servente per tutti e cinque i centri, per cui:

$$S_1 = 1/\mu_1 = 90 \text{ sec}$$

$$S_2 = 1/\mu_2 = 180 \text{ sec}$$

$$S_3 = 1/\mu_3 = 240 \text{ sec}$$

$$S_4 = 1/\mu_4 = 1800 \text{ sec}$$

$$S_5 = 1/\mu_5 = 1800 \text{ sec}$$

Calcolo del numero medio di visite ai centri del sistema.

$$\lambda_1 = \gamma(1-q_E)$$

$$\lambda_2 = \gamma q_E$$

$$\lambda_3 = \lambda_1[(1-q_C)(1-P_{car})+P_{car}]+\lambda_2(1-q_A)$$

$$\lambda_4 = [\lambda_1(1-q_C)(1-P_{car})+\lambda_2(1-q_A)]q_T$$

$$\lambda_5 = \lambda_4 q_{play}$$

Da queste equazioni, in base ai valori dei parametri fissati nel modello delle specifiche, si ottiene che:

$$\lambda_1 = 0,4\gamma$$

$$\lambda_2 = 0,6\gamma$$

$$\lambda_3 = [0,04+0,36(1-q_C)+0,6(1-q_A)] \gamma$$

$$\lambda_4 = [0,252(1-q_C)+0,42(1-q_A)] \gamma$$

$$\lambda_5 = [0,0756(1-q_C)+0,126(1-q_A)] \gamma$$

Da cui:

$$v_1 = \lambda_1/\gamma = 0,4$$

$$v_2 = \lambda_2/\gamma = 0,6$$

$$v_3 = \lambda_3/\gamma = [0,04+0,36(1-q_C)+0,6(1-q_A)]$$

$$v_4 = \lambda_4/\gamma = [0,252(1-q_C)+0,42(1-q_A)]$$

$$v_5 = \lambda_5/\gamma = [0,0756(1-q_C)+0,126(1-q_A)]$$

Calcolo della domanda media per i centri del sistema.

$$D_1 = v_1 S_1 = 36 \text{ sec}$$

$$D_2 = v_2 S_2 = 108 \text{ sec}$$

$$D_3 = v_3 S_3 = [9,6 + 86,4(1 - q_c) + 144(1 - q_A)] \text{ sec}$$

$$D_4 = v_4 S_4 = [453,6(1 - q_c) + 756(1 - q_A)] \text{ sec}$$

$$D_5 = v_5 S_5 = [136,08(1 - q_c) + 226,8(1 - q_A)] \text{ sec}$$

Eseguendo la simulazione a orizzonte infinito per le singole fasce orarie abbiamo ottenuto i valori per la probabilità (media) di abbandono dal centro 1 (*ordering and payment*), per la probabilità (media) di abbandono dal centro 2 (*electronic ordering and payment*) e per le domande medie relative ai cinque centri. I valori sono mostrati nella seguente tabella:

	09:00 → 11:00	11:00 → 12:00	12:00 → 15:00	15:00 → 18:00	18:00 → 22:00	22:00 → 23:00
q_c	0,00005	0	0,016909	0	0,280541	0,00005
q_A	0,034454	0,000732	0,452461	0,004951	0,723259	0,034454
D₁ (sec)	36	36	36	36	36	36
D₂ (sec)	108	108	108	108	108	108
D₃ (sec)	235,034304	239,894592	173,3846784	239,287056	111,6119616	235,034304
D₄ (sec)	1183,530096	1209,046608	859,8695616	1205,857044	535,5627984	1183,530096
D₅ (sec)	355,0590288	362,7139824	257,9608685	361,7571132	160,6688395	355,0590288

È possibile osservare che, secondo l'analisi appena effettuata, in tutte le fasce orarie il centro 4 (*consumption*) rappresenta il collo di bottiglia, mentre il centro 5 (*playground*) è il secondo centro con domanda media più elevata. Di conseguenza, aumentando il numero di serventi del centro 4, il centro 5 è destinato a diventare il nuovo collo di bottiglia.

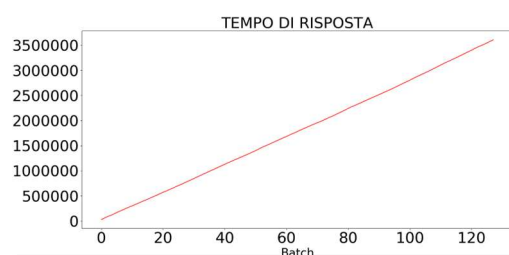
Esecuzione delle simulazioni.

Simulazione a orizzonte infinito.

Gli esperimenti che sono stati progettati in questa simulazione ci hanno consentito di identificare, per ogni fascia oraria e per ogni centro, la configurazione ottimale garantendo il soddisfacimento dei QoS definiti all'inizio dello studio. La configurazione così ottenuta può essere presa come punto di partenza per le simulazioni a orizzonte finito, che sono relative alla giornata complessiva.

Fascia 1 (09:00 → 11:00).

Il tasso degli arrivi al sistema è fortemente influenzato dalla colazione e, di conseguenza, è una delle fasce orarie più affollate durante la giornata. A seguito dei risultati conseguiti nel calcolo del collo di bottiglia, abbiamo iniziato la simulazione utilizzando una configurazione che preveda più serventi nel centro *playground* e soprattutto nel centro *consumption*, ovvero la configurazione (1, 1, 1, 15, 10). Dai risultati della simulazione osserviamo che per il centro *withdraw food* il tempo di risposta diverge:



Di conseguenza, questa configurazione porta il centro *withdraw food* a essere il collo di bottiglia del sistema e non ci permette di soddisfare il primo QoS. Effettivamente, gli altri centri non presentano delle metriche di prestazione negative come quelle del centro *withdraw food*. Per risolvere il problema, abbiamo cercato la prima configurazione con la quale è possibile soddisfare il QoS. Per prima cosa, abbiamo aumentato il numero dei serventi del centro *withdraw food* senza alterare la configurazione per gli altri centri.

Aggiungendo un solo servente, otteniamo un tempo medio di risposta nell'intervallo di confidenza $411,770131 \pm 10,319925$ sec, che è abbondantemente al di sotto della soglia massima dei 15 minuti (900 sec). Inoltre, il tempo medio di attesa che si liberi un tavolo che un utente sperimenta per poter consumare il pasto e la percentuale di bambini che non trovano un posto libero nell'area giochi specificati dai QoS sono entrambi soddisfatti da questa configurazione. Infatti, gli intervalli di confidenza che ricaviamo per il tempo medio di attesa e la probabilità sono rispettivamente $0,288185 \pm 0,159059$ sec (abbondantemente al di sotto della soglia massima di 30 secondi) e $0,000046 \pm 0,000056$ (abbondantemente al di sotto della soglia massima di 0,2).

Data la configurazione (1, 1, 2, 15, 10) non è possibile diminuire il numero di serventi dei primi tre centri. Tuttavia, abbiamo trovato il valore minimo di serventi nei centri *consumption* e *playground* in modo da minimizzare i costi e da rispettare i QoS. La configurazione finale per la fascia oraria è (1, 1, 2, 11, 3), dove l'intervallo di confidenza del tempo medio di attesa nel centro *consumption* è dato da $19,390623 \pm 3,552156$ e l'intervallo di confidenza della probabilità di perdita nel centro *playground* è dato da $0,185143 \pm 0,003456$.

Fascia 2 (11:00 → 12:00).

Nella fascia 1 sono richiesti 11 serventi per il centro *consumption* e 3 serventi per il centro *playground*.

Poiché, per loro natura, questi due centri hanno lo stesso numero di serventi nell'arco di tutta la giornata, abbiamo considerato la configurazione (1, 1, 1, 11, 3) come base per eseguire la simulazione a orizzonte infinito relativa alla seconda fascia oraria. Di fatto, una diminuzione del numero di serventi negli ultimi due centri porterebbe un costo minore (e quindi un profitto maggiore) per il ristorante, ma non consentirebbe di rispettare i QoS nella prima fascia oraria.

Per questa fascia, con questa configurazione otteniamo i seguenti intervalli di confidenza:

- Tempo di risposta nel centro *withdraw food*: $480,702329 \pm 10,904113$ sec.
- Tempo di attesa nel centro *consumption*: $0,005436 \pm 0,006836$ sec.
- Probabilità di perdita nel centro *playground*: $0,029106 \pm 0,001479$.

È possibile osservare che tutti e tre i QoS sono già soddisfatti, per cui (1, 1, 1, 11, 3) è una configurazione accettabile.

Fascia 3 (12:00 → 15:00).

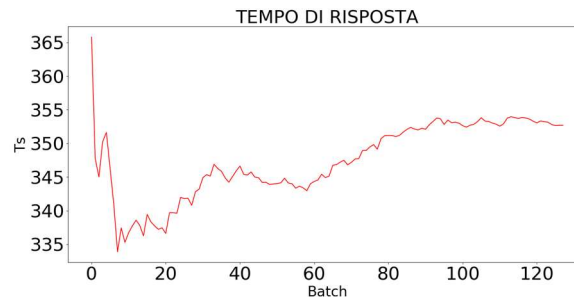
Essendo questa la seconda fascia più affollata della giornata, abbiamo iniziato con una configurazione che preveda un numero di serventi per il centro *withdraw food* almeno pari a quello utilizzato nella fascia 1, nella quale il centro era collo di bottiglia. La configurazione dunque è (1, 1, 2, 11, 3). Tuttavia, un numero di serventi pari a 2 non è sufficiente per far convergere il tempo di risposta relativo al centro *withdraw food*. Incrementando il numero di serventi in tale centro si osserva che il primo QoS viene soddisfatto con l'utilizzo di 4 serventi. Infatti, con quest'ultima configurazione, l'intervallo di confidenza del tempo medio di risposta di *withdraw food* risulta essere di $352,687144 \pm 8,178942$ sec, che è al di sotto dei 15 minuti massimi previsti.

Tuttavia, il numero di tavoli e di posti all'interno dell'area giochi non sono sufficienti per soddisfare gli altri due QoS. Incrementando il numero di serventi anche nei centri *consumption* e *playground*, si arriva alla configurazione finale (1, 1, 4, 19, 5), in cui si ottengono i seguenti intervalli di confidenza:

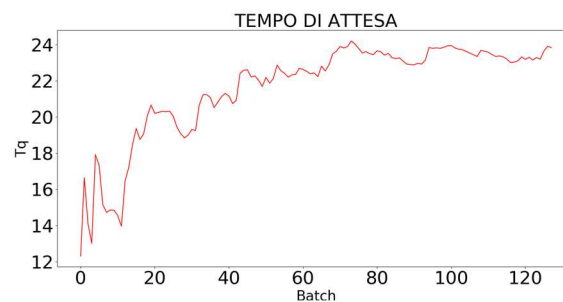
- Tempo di attesa nel centro *consumption*: $23,835521 \pm 4,024639$ sec (al di sotto dei 30 secondi massimi previsti).

- Probabilità di perdita nel centro *playground*: 0,189461 +/- 0,004092 (al di sotto del 20% massimo previsto).

Tempo di risposta del centro withdraw food nella terza fascia oraria (configurazione (1, 1, 4, 19, 5)):



Tempo di attesa del centro consumption nella terza fascia oraria (configurazione (1, 1, 4, 19, 5)):



Fascia 4 (15:00 → 18:00).

Dovendo come prima mantenere i tavoli e i posti nell'area giochi ottenuti nelle fasce orarie precedenti, per la simulazione a orizzonte infinito della fascia 4 siamo partiti dalla configurazione (1, 1, 1, 19, 5). Tuttavia, tale configurazione non rispetta il QoS relativo al tempo medio di risposta del centro *withdraw food*, il cui intervallo di confidenza risulta essere 1152,469797 +/- 68,221994 sec, che supera i 15 minuti massimi.

Aggiungendo un servente al centro *withdraw food*, si ottiene la configurazione (1, 1, 2, 19, 5) e un intervallo di confidenza sul tempo di risposta di tale centro pari a 285,572421 +/- 3,157248 sec, che è accettabile.

Fascia 5 (18:00 → 22:00).

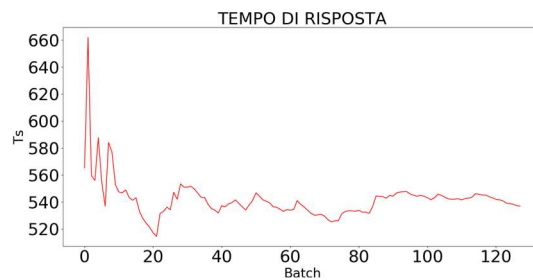
Essendo la fascia oraria più popolata di tutte, abbiamo deciso di partire dalla configurazione ottenuta nella fascia oraria relativa all'ora di pranzo (la terza), che è (1, 1, 4, 19, 5). Utilizzando già questa configurazione, nessuno dei QoS è stato soddisfatto. Infatti, sono stati ottenuti i seguenti intervalli di confidenza:

- Tempo di risposta nel centro *withdraw food*: 1923,344517 +/- 386,035053 sec.
- Tempo di attesa nel centro *consumption*: 258,332669 +/- 38,187410 sec.
- Probabilità di perdita nel centro *playground*: 0,268646 +/- 0,005225.

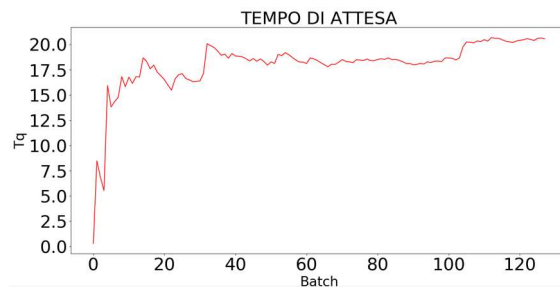
La configurazione minima che invece rispetta tutti i QoS è (1, 1, 5, 23, 6), da cui si ottengono i seguenti intervalli di confidenza:

- Tempo di risposta nel centro *withdraw food*: 536,987959 +/- 35,245249 sec.
- Tempo di attesa nel centro *consumption*: 20,560713 +/- 4,599469 sec.
- Probabilità di perdita nel centro *playground*: 0,178116 +/- 0,004405.

Tempo di risposta del centro withdraw food nella quinta fascia oraria (configurazione (1, 1, 5, 23, 6)):



Tempo di attesa del centro consumption nella quinta fascia oraria (configurazione (1, 1, 5, 23, 6)):



Fascia 6 (22:00 → 23:00).

Poiché il tasso degli arrivi al ristorante nella fascia 6 è uguale al tasso degli arrivi nella fascia 1, in base alla simulazione a orizzonte infinito, la configurazione minima che rispetta tutti i QoS coincide con quella ottenuta analizzando la fascia 1. Comunque sia, anche in questo caso è stato necessario mantenere il numero di serventi nei centri *consumption* e *playground* pari al massimo che si è trovato per le fasce orarie finora analizzate (ovvero rispettivamente 23 e 6), arrivando così alla configurazione finale (1, 1, 2, 23, 6).

Conclusioni.

In definitiva, in base ai risultati ottenuti con la simulazione a orizzonte infinito, in base ai soli requisiti QoS e tenendo conto che il numero di tavoli e di posti nell'area giochi è fisso nell'arco di tutta la giornata, abbiamo ottenuto le seguenti configurazioni minime per le singole fasce orarie:

- > Fascia 1: (1, 1, 2, 23, 6)
- > Fascia 2: (1, 1, 1, 23, 6)
- > Fascia 3: (1, 1, 4, 23, 6)
- > Fascia 4: (1, 1, 2, 23, 6)
- > Fascia 5: (1, 1, 5, 23, 6)
- > Fascia 6: (1, 1, 2, 23, 6)

Come ipotizzato nella fase di progettazione, vedendo le configurazioni risultanti nelle fasce orarie, che si basano esclusivamente sui QoS, i centri *ordering and payment* ed *electronic ordering and payment* hanno il numero minimo di serventi possibile, che verosimilmente porterà a una gran quantità di perdite all'interno del sistema e, di conseguenza, una perdita nel profitto.

Simulazione a orizzonte finito.

Utilizzando le configurazioni ricavate con la simulazione a orizzonte infinito, gli intervalli di confidenza che sono stati ottenuti nella simulazione a orizzonte finito sono i seguenti:

- Tempo di risposta medio giornaliero nel centro *withdraw food*: 444,828219 +/- 22,826368 sec.
- Tempo di attesa medio giornaliero nel centro *consumption*: 7,339941 +/- 5,472249 sec.
- Probabilità di perdita media giornaliera nel centro *playground*: 0,088595 +/- 0,010996.

Come si può notare, i QoS sono rispettati utilizzando le configurazioni ottenute con le simulazioni a orizzonte infinito. Il profitto mensile che si riesce a ottenere con queste configurazioni è pari a 253.592,19€.

A questo punto, abbiamo provato ad aumentare il numero di serventi nei centri *ordering and payment* ed *electronic ordering and payment* con l'intento di diminuire le perdite ma mantenendo sempre soddisfatti i QoS richiesti.

Come prima cosa, ci siamo concentrati sulla fascia oraria relativa alla cena, essendo quella più popolata. Inoltre, poiché la maggior parte del traffico si riversa verso il centro 2, abbiamo per prima aumentato il numero dei serventi di tale centro. Nel momento in cui il profitto ha iniziato a ridursi, ci siamo concentrati sull'aumento del numero di serventi del centro 1, cercando di aumentare ulteriormente il guadagno. Quando non è stato più possibile aumentare il profitto, siamo passati alla fascia oraria di pranzo (i.e. la seconda più popolata). Anche in questo caso, siamo partiti con l'aumentare il numero di serventi prima per il centro 2 e successivamente per il centro 1. Effettivamente, aumentando il numero dei serventi, siamo riusciti a incrementare il profitto totale. A questo punto, ci siamo occupati del numero di serventi per i due centri nelle fasce orarie meno popolate. Partendo dalla fascia oraria relativa alla colazione, abbiamo incrementato il numero di serventi del centro 2, essendo quello con un tasso di arrivo maggiore. Tuttavia, l'incremento del numero dei serventi per il centro 1 ha portato solamente a una diminuzione del profitto. Effettivamente, il numero di serventi per il primo centro nella fascia oraria di pranzo risulta essere pari a 1 e si ritrova ad avere un tasso di arrivo maggiore. Per quanto riguarda la seconda fascia oraria, non c'è stato alcun incremento del numero di serventi nei due centri in quanto con un servente si riesce a massimizzare il profitto con queste configurazioni. Infine, nell'ultima fascia oraria, c'è stata la necessità di incrementare il numero di serventi di entrambi i centri: probabilmente, ciò è dovuto all'alto tasso di arrivi che si ha durante l'ora di cena che determina una dimensione della coda non trascurabile al termine della fascia oraria di cena. Questo implica che i serventi dell'ultima fascia devono gestire anche le richieste rimanenti della fascia precedente.

Fino a questo punto, siamo stati in grado di trovare le configurazioni ottimali che tengono conto del profitto e dei QoS per i centri 1 e 2. Tali configurazioni sono:

- > Fascia 1: (1, 3, 2, 23, 6)
- > Fascia 2: (1, 1, 1, 23, 6)
- > Fascia 3: (1, 4, 4, 23, 6)
- > Fascia 4: (1, 1, 2, 23, 6)
- > Fascia 5: (2, 7, 5, 23, 6)
- > Fascia 6: (2, 3, 2, 23, 6)

Il profitto mensile che si riesce a ottenere con queste configurazioni è pari a 416.967,50€.

A seguito della diminuzione del numero di perdite nei centri 1 e 2, si ha un flusso di arrivi maggiore in tutti i centri successivi. Di conseguenza, le configurazioni finora ottenute potrebbero non essere più sufficienti per rispettare i QoS definiti. Abbiamo dunque deciso di tornare a eseguire delle simulazioni a orizzonte infinito per determinare il numero di serventi per i centri 3, 4 e 5 nelle fasce orarie che hanno subito delle modifiche, tenendo conto dell'aumento del flusso dovuto all'incremento del numero di serventi nei centri 1 e 2.

Fascia 1 (09:00 → 11:00).

A seguito dell'incremento del numero di serventi nei primi due centri, per la prima fascia oraria otteniamo ora i seguenti intervalli di confidenza:

- Tempo di risposta nel centro *withdraw food*: 431,558693 +/- 12,388369 sec.
- Tempo di attesa nel centro *consumption*: 0,000000 +/- 0,000000 sec.
- Probabilità di perdita nel centro *playground*: 0,008766 +/- 0,000864.

In realtà, qui i QoS continuano a essere rispettati, per cui non è stato necessario alcun incremento del numero di serventi per questi tre centri.

Fascia 2 (11:00 → 12:00).

In questa fascia oraria non c'è stata alcuna modifica per quanto riguarda i primi due centri, per cui il numero degli arrivi nei tre centri successivi non è aumentato rispetto a prima. Per questo motivo, anche qui non è necessario incrementare il numero di serventi per i centri *withdraw food*, *consumption* e *playground*.

Fascia 3 (12:00 → 15:00).

A seguito dell'incremento del numero di serventi nei primi due centri, per la terza fascia oraria otteniamo ora i seguenti intervalli di confidenza:

- Tempo di risposta nel centro *withdraw food*: 5882,680633 +/- 1271,514578 sec.
- Tempo di attesa nel centro *consumption*: 99,365344 +/- 16,039886 sec.
- Probabilità di perdita nel centro *playground*: 0,237900 +/- 0,004705.

È possibile osservare che, per l'ora di pranzo presa singolarmente, i QoS non sono più rispettati. Perciò abbiamo deciso di aumentare il numero di serventi al centro 3 che ha comportato la necessità di aumentare anche il numero di serventi nei centri 4 e 5. Per rispettare i vincoli imposti nella sola fascia oraria relativa al pranzo, siamo giunti alla configurazione (1, 4, 5, 26, 7), che ha comportato i seguenti intervalli di confidenza:

- Tempo di risposta nel centro *withdraw food*: 372,355406 +/- 12,205325 sec.
- Tempo di attesa nel centro *consumption*: 16,827864 +/- 3,673941 sec.
- Probabilità di perdita nel centro *playground*: 0,159576 +/- 0,004571.

Fascia 4 (15:00 → 18:00).

Come nella fascia 2, non essendoci state modifiche nei primi due centri, non è necessario incrementare il numero di serventi per i centri *withdraw food*, *consumption* e *playground*.

Fascia 5 (18:00 → 22:00).

A seguito dell'incremento del numero di serventi nei primi due centri, nella quinta fascia oraria il tempo di risposta nel centro *withdraw food* e il tempo di attesa nel centro *consumption* divergono; la probabilità di perdita nel centro *playground* ora cade nell'intervallo di confidenza 0,461929 +/- 0,005429. Dunque, è stato necessario incrementare il numero di serventi in questi tre centri fino a raggiungere la configurazione (2, 7, 9, 47, 12), che dà luogo ai seguenti intervalli di confidenza:

- Tempo di risposta nel centro *withdraw food*: 459,511787 +/- 31,260546 sec.
- Tempo di attesa nel centro *consumption*: 16,601332 +/- 4,400772 sec.
- Probabilità di perdita nel centro *playground*: 0,170029 +/- 0,005551.

Fascia 6 (22:00 → 23:00).

A seguito dell'incremento del numero di serventi nei primi due centri, per la sesta fascia oraria otteniamo i seguenti intervalli di confidenza:

- Tempo di risposta nel centro *withdraw food*: 430,983810 +/- 12,291534 sec.
- Tempo di attesa nel centro *consumption*: 0,000000 +/- 0,000000 sec.
- Probabilità di perdita nel centro *playground*: 0,000000 +/- 0,000000.

In realtà, qui i QoS continuano a essere rispettati, per cui non è stato necessario alcun incremento del numero di serventi per questi tre centri.

In definitiva, abbiamo ottenuto le seguenti configurazioni per le sei fasce orarie:

- > Fascia 1: (1, 3, 2, 47, 12)
- > Fascia 2: (1, 1, 1, 47, 12)
- > Fascia 3: (1, 4, 5, 47, 12)
- > Fascia 4: (1, 1, 2, 47, 12)

-> Fascia 5: (2, 7, 9, 47, 12)

-> Fascia 6: (2, 3, 2, 47, 12)

Tornando all'esecuzione delle simulazioni a orizzonte finito, il profitto mensile che si riesce a ottenere con queste configurazioni è pari a 412.107,50€. Siccome viene processato lo stesso numero di job (e quindi abbiamo lo stesso ricavo) ma ci sono più serventi (e quindi abbiamo una spesa maggiore), il profitto diminuisce.

A questo punto, siamo partiti da queste nuove configurazioni per cercare di identificare il valore dei serventi per i centri 3, 4 e 5 che ci consenta di massimizzare il profitto mantenendo però i QoS.

Gli intervalli di confidenza per le statistiche mediate nell'intera giornata che otteniamo da queste nuove configurazioni sono i seguenti:

- Tempo di risposta nel centro *withdraw food*: 429,988167 +/- 25,796215 sec.

- Tempo di attesa nel centro *consumption*: 4,189602 +/- 2,472621 sec.

- Probabilità di perdita nel centro *playground*: 0,070180 +/- 0,009736.

Come si può osservare, queste configurazioni ci permettono di rispettare i QoS definiti. Di conseguenza, abbiamo provato a diminuire il numero di serventi nei centri 3, 4 e 5 cercando di aumentare il profitto.

Abbiamo iniziato col diminuire il numero di serventi nel centro 3. La diminuzione ha comportato un aumento del tempo di interarrivo medio tra i job nei centri successivi che ci ha consentito di diminuire il numero di serventi anche in tali centri successivi.

Conclusioni.

Le configurazioni finali che massimizzano il profitto e, al tempo stesso, rispettano i QoS obiettivi di studio sono le seguenti:

-> Fascia 1: (1, 3, 1, 43, 9)

-> Fascia 2: (1, 1, 1, 43, 9)

-> Fascia 3: (1, 4, 5, 43, 9)

-> Fascia 4: (1, 1, 1, 43, 9)

-> Fascia 5: (2, 7, 8, 43, 9)

-> Fascia 6: (2, 3, 1, 43, 9)

Con queste configurazioni, siamo riusciti ad aumentare notevolmente il profitto portandolo a 414.617,50€.

Gli intervalli di confidenza ottenuti per l'intera giornata sono i seguenti:

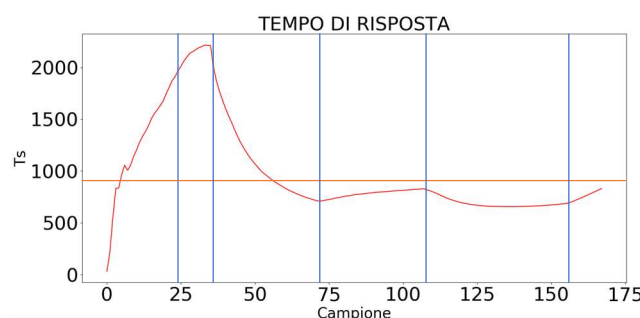
- Tempo di risposta nel centro *withdraw food*: 830,326723 +/- 58,470013 sec.

- Tempo di attesa nel centro *consumption*: 15,877398 +/- 6,944076 sec.

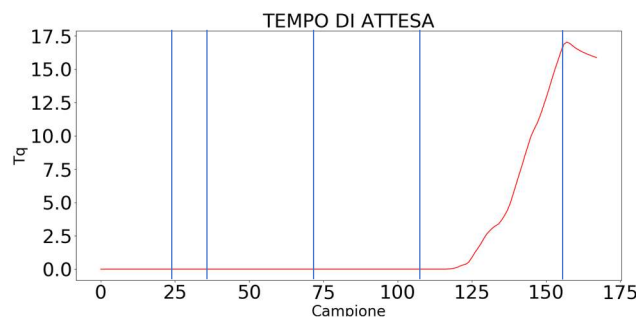
- Probabilità di perdita nel centro *playground*: 0,157930 +/- 0,012733.

Raggiunte le configurazioni finali, il tempo medio di risposta nel centro *withdraw food*, il tempo di attesa nel centro *consumption* e la probabilità di perdita nel centro *playground* sono rappresentati dai seguenti grafici.

Tempo medio di risposta giornaliero del centro *withdraw food*:



Tempo medio di attesa giornaliero del centro consumption:



Probabilità media di perdita giornaliera nel centro playground:



Modello migliorativo.

Descrizione del modello migliorativo.

Supponiamo che per la sicurezza dei clienti venga introdotta una nuova legge che preveda una capienza massima per il ristorante. In particolare, si ha un limite massimo di famiglie che possono attendere in coda nei primi centri che vengono visitati dai clienti, che sono *ordering and payment* ed *electronic ordering and payment*. Le automobili, invece, non subiscono alcuna limitazione.

In queste condizioni, le famiglie sono scoraggiate ad abbandonare il sistema prima di effettuare l'ordine del loro pasto. Di fatto, ora non è più scontato che una famiglia riesca a entrare all'interno del sistema, per cui è più invogliata ad attendere in coda; inoltre, il numero di job in coda è sempre limitato, per cui un'eventuale attesa risulterebbe comunque sufficientemente breve.

Obiettivi del modello migliorativo.

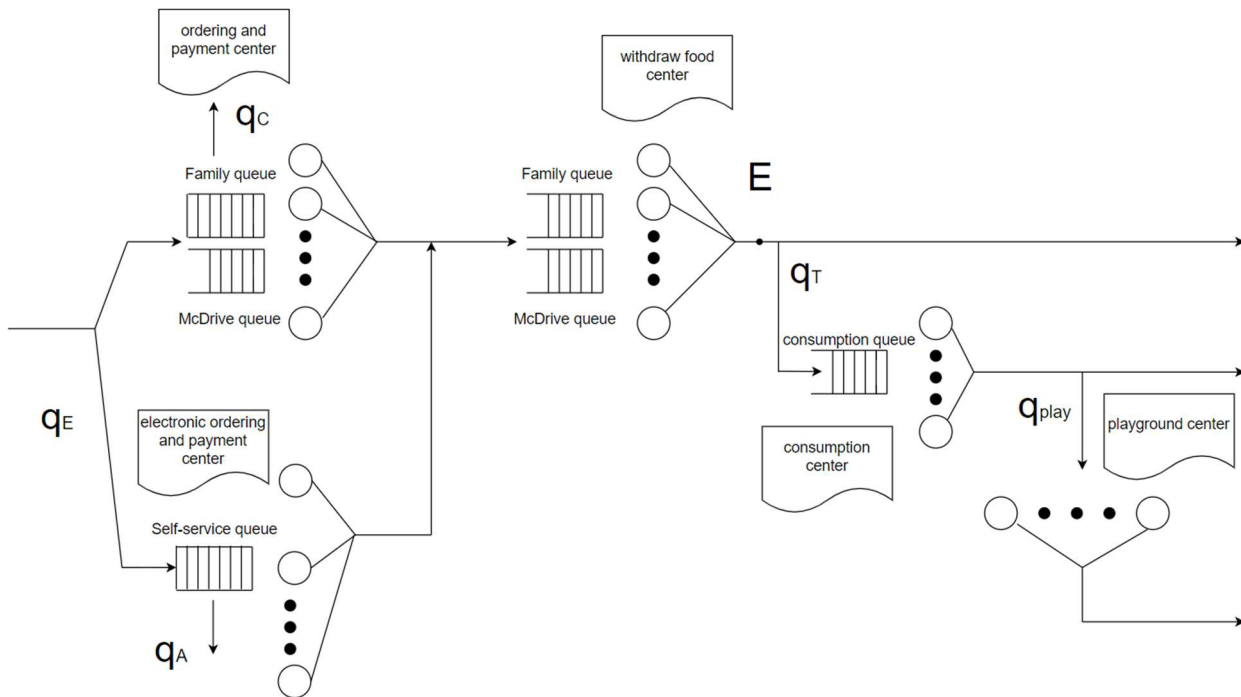
Nel modello migliorativo abbiamo fissato i medesimi obiettivi del modello base: si vuole massimizzare il profitto (che dipende dal numero di serveri attivati in ciascun centro nelle varie fasce orarie e dal numero di job persi nei centri *ordering and payment* ed *electronic ordering and payment*) e si vuole rispettare i soliti tre requisiti QoS:

- 1) Il tempo medio giornaliero per un cliente per ritirare il cibo deve essere inferiore ai 15 minuti (attesa in coda inclusa).
- 2) Il tempo medio giornaliero in cui un cliente, dopo aver pagato e ricevuto il pasto, rimane in attesa che si liberi un tavolo per poter consumare il pasto deve essere minore di 30 secondi.

3) Almeno l'80% dei bambini che richiedono nell'arco della giornata l'uso dell'area giochi deve trovare un posto libero.

Modello concettuale.

Il system diagram del sistema evoluto è illustrato nella seguente figura:



q_E : job che utilizzano le casse automatiche per ordinare e pagare
 q_E^C : job che utilizzano le casse fisiche per ordinare e pagare
 q_A : job che non trovano posto nel centro relativo alle casse automatiche
 q_C : job che non trovano posto nel centro relativo alle casse fisiche
 q_T : job che vanno a consumare il pasto al tavolo
 q_T^C : job che, dopo aver ricevuto il pasto, escono dal sistema
 q_{play} : job che vanno verso l'area giochi
 q_{play}^C : job che, dopo aver consumato il pasto, escono dal sistema
 E : punto del sistema a cui giungono tutti e soli i job che apportano un ricavo

Politiche di scheduling dei job all'interno delle code.

Le politiche di scheduling rimangono identiche rispetto a quelle definite nel modello base.

Le automobili hanno priorità sulle famiglie ma, per come è fatta fisicamente la struttura del McDrive, è impossibile servire più automobili per volta all'interno dello stesso centro. In altre parole, se un'automobile è già in servizio, gli altri serventi possono occuparsi solo delle famiglie, lasciando eventuali altre automobili in attesa.

I centri *electronic ordering and payment* e *consumption* presentano un'unica coda caratterizzata da uno scheduling FIFO. Infine, il centro playground non ha una coda.

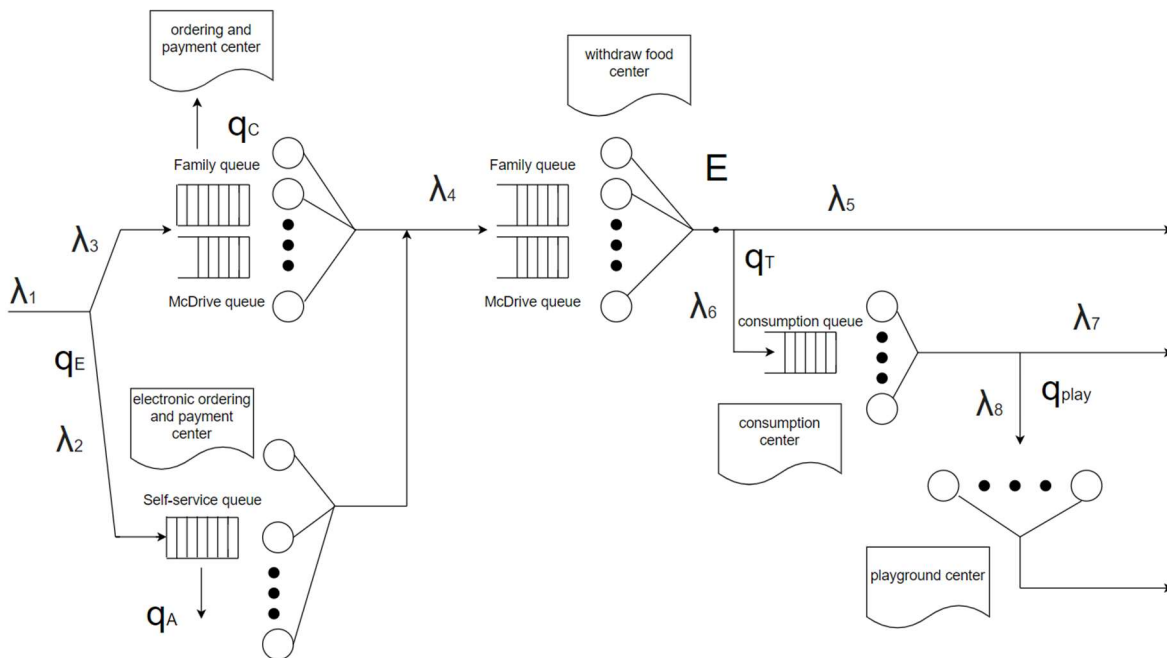
Stato di code e serventi.

In qualunque istante, lo stato di ciascuna coda può essere *empty* (vuota) o *not empty* (non vuota), lo stato di ciascun servente dei centri con una singola coda può essere *idle* (libero) o *busy* (occupato), e lo stato di ciascun servente dei centri con due code (coda del McDrive e coda delle famiglie) può essere *idle* (libero), *busy_a* (occupato con un'automobile) oppure *busy_f* (occupato con una famiglia).

Eventi.

Gli eventi considerati rimangono gli stessi rispetto al modello base, con la differenza che gli eventi di perdita per i centri *ordering and payment* ed *electronic ordering and payment* occorrono in condizioni diverse. In particolare, quando arriva una famiglia, se all'interno del centro si è già raggiunto il numero massimo di famiglie che attendono in coda, allora la famiglia non trova posto ed è costretta ad abbandonare immediatamente il sistema.

Modello delle specifiche.



$$\lambda_1 = \lambda_{MD} + \lambda_{FF} + \lambda_{FE}$$

$$\lambda_2 = \lambda_{FE}$$

$$\lambda_3 = \lambda_{MD} + \lambda_{FF}$$

$$\lambda_4 = \lambda_{MD} + \lambda_{FF} * (1 - q_c) + \lambda_{FE} * (1 - q_A)$$

$$\lambda_5 = \lambda_{MD} + [\lambda_{FF} * (1 - q_c) + \lambda_{FE} * (1 - q_A)] * (1 - q_T)$$

$$\lambda_6 = [\lambda_{FF} * (1 - q_c) + \lambda_{FE} * (1 - q_A)] * (q_T)$$

$$\lambda_7 = [\lambda_{FF} * (1 - q_c) + \lambda_{FE} * (1 - q_A)] * (q_T) * (1 - q_{play})$$

$$\lambda_8 = [\lambda_{FF} * (1 - q_c) + \lambda_{FE} * (1 - q_A)] * (q_T) * (q_{play})$$

Modellazione dei centri.

Centro ordering and payment

- Arrivi Markoviani nella coda del McDrive (tasso λ_{MD})

- Arrivi Markoviani nella coda delle famiglie (tasso λ_{FF}) $\rightarrow \lambda_{MD} + \lambda_{FF} = \lambda_3$
- La perdita di un job J dalla coda delle famiglie si verifica nel momento in cui la coda contiene già H job in attesa.
- Tempi di servizio esponenziali (media $E[S_i] = 1,5$ minuti)

Centro electronic ordering and payment (centro M/M/m/n)

- Arrivi Markoviani (tasso $\lambda_{FE} = \lambda_2$)
- La perdita di un job J si verifica nel momento in cui la coda contiene già $H=n-m$ job in attesa.
- Tempi di servizio esponenziali (media $E[S_i] = 3$ minuti)

Per entrambi i centri, il numero massimo di job in coda che abbiamo fissato è $H = 6$.

Per quanto riguarda la matrice di routing, le fasce orarie e la proporzione tra numero di automobili e numero di famiglie, le abbiamo lasciate invariate rispetto al modello base.

Algoritmi per la Next-Event simulation.

CENTRO ORDERING AND PAYMENT

Variabili di stato

- q_A = popolazione nella coda McDrive
- q_F = popolazione nella coda famiglie
- $x_i \in \{idle, busy_a, busy_f\} \forall i$

Evento 1 (arrivo di un'automobile)

```

 $t_{aA\_OP} := GetArrival()$ 
if ( $t_{aA\_OP} > \tau$ ) then
     $t_{aA\_OP} := \infty$ 
endif
if ( $\nexists x_k = busy\_a \wedge q_A = 0 \wedge \exists x_i = idle$ ) then
     $x_i := busy\_a$ 
     $t_{ci} := t + GetService()$ 
else
     $q_A++$ 
endif

```

Evento 2 (arrivo di una famiglia)

```

 $t_{aF\_OP} := GetArrival()$ 
if ( $t_{aF\_OP} > \tau$ ) then
     $t_{aF\_OP} := \infty$ 
endif
if ( $q_F < H$ ) then
    if ( $(q_A = 0 \vee \exists x_k = busy\_a) \wedge q_F = 0 \wedge \exists x_i = idle$ ) then
         $x_i := busy\_f$ 
         $t_{ci} := t + GetService()$ 
    else
         $q_F++$ 
    endif
endif
endif

```

Evento 3 (completamento di un'automobile)

```

if ( $q_A \neq 0$ ) then
   $t_{ci} := t + \text{GetService}()$ 
   $q_A--$ 
else if ( $q_F \neq 0$ ) then
   $t_{ci} := t + \text{GetService}()$ 
   $q_F--$ 
   $x_i := \text{busy}_f$ 
else
   $t_{ci} := \infty$ 
   $x_i := \text{idle}$ 
endif
arrivalsMcDriveWithdrawFoodList.append(t)

```

Evento 4 (completamento di una famiglia)

```

if ( $\nexists x_k = \text{busy}_a \wedge q_A \neq 0$ ) then
   $t_{ci} := t + \text{GetService}()$ 
   $q_A--$ 
   $x_i := \text{busy}_a$ 
else if ( $(q_A = 0 \wedge q_F \neq 0) \vee (q_F \neq 0 \wedge \exists x_k = \text{busy}_a)$ ) then
   $t_{ci} := t + \text{GetService}()$ 
   $q_F--$ 
else
   $t_{ci} := \infty$ 
   $x_i := \text{idle}$ 
endif
arrivalsFamilyWithdrawFoodList.append(t)

```

CENTRO ELECTRONIC ORDERING AND PAYMENT**Variabili di stato**

- I = popolazione nel centro
 - $x_i \in \{\text{idle}, \text{busy}\} \forall i$
-

Evento 1 (arrivo)

```

 $t_{aE} := \text{GetArrival}()$ 
if ( $t_{aE} > \tau$ ) then
   $t_{aE} := \infty$ 
endif
if ( $I - \text{num\_serventi} < H$ ) then
   $I++$ 
  if ( $\exists x_i = \text{idle}$ ) then
     $x_i := \text{busy}$ 
     $t_{ci} := t + \text{GetService}()$ 
  endif
endif

```

Evento 2 (completamento)

```

/--
if (l >= num_serventi) then
    tci := t + GetService()
else
    tci := ∞
    xi := idle
endif
arrivalsFamilyWithdrawFoodList.append(t)

```

Calcolo delle probabilità di perdita del centro *electronic ordering and payment*.

Per ricavare il valore della probabilità di perdita per il centro *electronic ordering and payment*, abbiamo risolto la relativa catena di Markov ottenendo la formula generale per il caso M/M/m/n:

$$\pi_i = (1/i!)(\lambda/\mu)^i \pi_0 \quad \text{per } i = 1, \dots, m$$

$$\pi_i = (1/m!)(1/m^{i-m})(\lambda/\mu)^i \pi_0 \quad \text{per } i = m+1, \dots, n$$

Da qui, la probabilità di perdita del centro è data da π_n .

Per quanto riguarda il centro *ordering and payment*, il valore della probabilità di perdita è stato ricavato con la simulazione.

Modello computazionale.

Le differenze nel modello computazionale tra i due modelli di sistema riguardano gli algoritmi dei centri *ordering and payment* ed *electronic ordering and payment*. Per come sono state considerate le perdite nel modello avanzato, le strutture dati che modellano la lista degli abbandoni non vengono più utilizzate.

Per separare l'esecuzione della simulazione del modello base rispetto a quella del modello avanzato, sono state utilizzate le direttive del pre-processor *#ifdef* e *#ifndef*.

Verifica.

Nel processo di verifica, si è utilizzata la configurazione (3, 3, 8, 43, 9) e la quinta fascia oraria e si è forzata a 0 la probabilità per un job di essere un'automobile. Assumendo una capacità massima per i centri *ordering and payment* ed *electronic ordering and payment* pari a 9 (inclusi i 3 serventi), abbiamo utilizzato la formula descritta nel paragrafo "Calcolo delle probabilità di perdita del centro *electronic ordering and payment*" e la legge di Little per calcolarci il valore delle metriche $E[T_s]$, $E[T_q]$, $E[N_s]$, $E[N_q]$, ρ , π_{loss} .

I risultati ottenuti utilizzando i modelli analitici per i centri *ordering and payment* ed *electronic ordering and payment*, per quanto riguarda le metriche sopra elencate, sono rappresentati all'interno della seguente tabella:

CENTRI	$E[T_s]$	$E[T_Q]$	$E[N_s]$	$E[N_Q]$	ρ	π_{loss}
Ordering and payment	96,93 s	6,93 s	1,291949	0,092366	0,399861	0,000347
Electronic ordering and payment	390,9942 s	210,9942 s	6,191667	3,341237	0,950143	0,208216

Gli intervalli di confidenza con livello di confidenza all' $1-\alpha = 0,99$, prodotti dalle simulazioni, sono riportati nella tabella di seguito:

CENTRI	$E[T_s]$	$E[T_Q]$	$E[N_s]$	$E[N_Q]$	ρ	π_{loss}
Ordering and payment	96,216408 +/- 0,976528 sec	6,711820 +/- 0,474466 sec	1,284393 +/- 0,017518	0,089844 +/- 0,006583	0,398183 +/- 0,004270	0,000267 +/- 0,000219
Electronic ordering and payment	390,226728 +/- 5,836877 sec	210,502484 +/- 4,528636 sec	6,186870 +/- 0,067611	3,336059 +/- 0,059143	0,950270 +/- 0,003470	0,206360 +/- 0,006582

Controlli di consistenza.

Nel processo di verifica sono stati effettuati i seguenti controlli di consistenza:

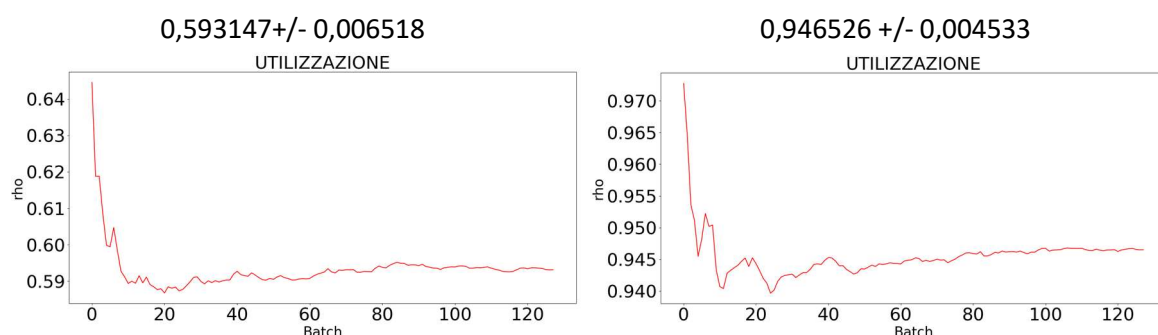
- > $E[T_s] = E[T_Q] + E[S_i]$
- > $E[N_s] = E[N_Q] + m \cdot \rho$
- > $\rho \geq 0$ & $\rho \leq 1$ & prob. perdita ≥ 0 & prob. perdita ≤ 1
- > $N_Q(t) + m \leq n \quad \forall t$ nella simulazione

Validazione.

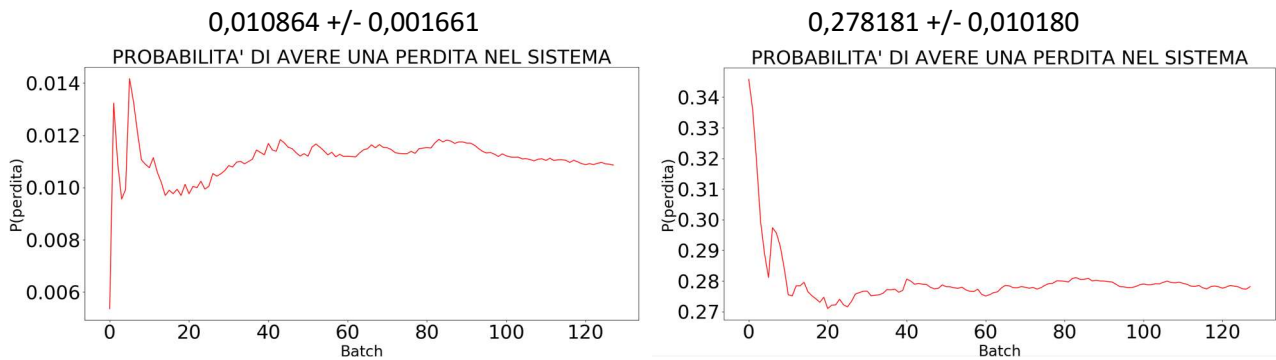
All'interno del sistema reale, a parità di serventi nei centri *ordering and payment* ed *electronic ordering and payment*, ci aspettiamo che nelle fasce orarie più affollate si abbia una probabilità di perdita maggiore ma al contempo un'utilizzazione maggiore.

Sono state eseguite delle simulazioni considerando le fasce orarie del pranzo e della cena e impostando un numero di serventi pari a 1 per entrambi i centri. Considerando che il tasso di arrivo al centro 2 è maggiore del tasso di arrivo al centro 1, che in quest'ultimo, oltre alle famiglie, è possibile che ci siano anche le automobili, e che il tasso di servizio del centro 2 è la metà di quello del centro 1, ci aspettiamo che l'utilizzazione e la probabilità di perdita del centro 2 siano più elevate rispetto a quelle del centro 1. In seguito, sono mostrati gli intervalli di confidenza e i grafici:

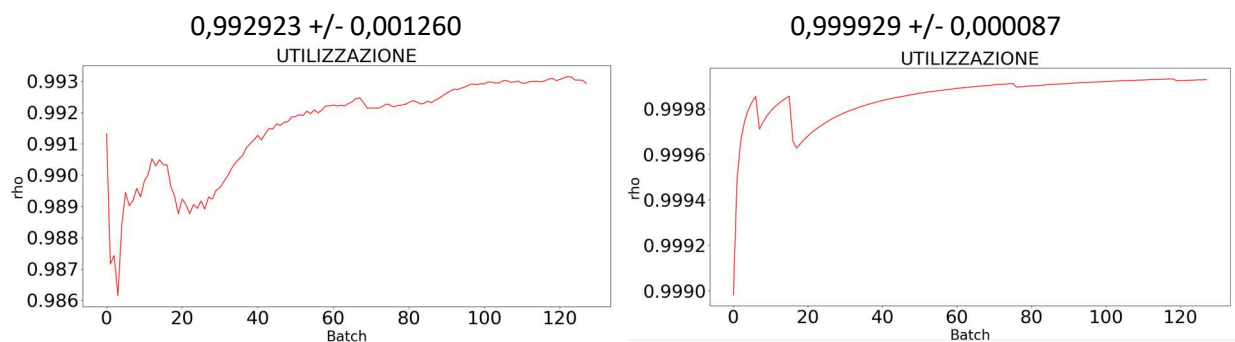
Utilizzazione del centro ordering and payment nella terza (sx) e nella quinta (dx) fascia oraria:



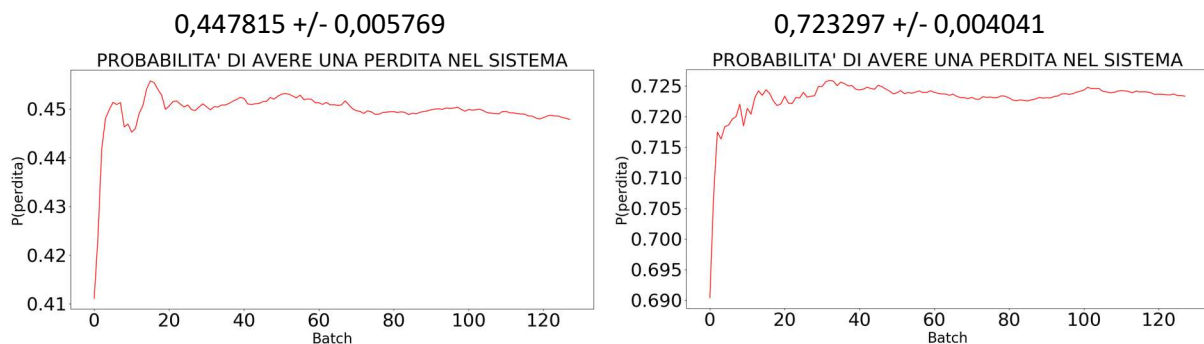
Probabilità di perdita del centro ordering and payment nella terza (sx) e nella quinta (dx) fascia oraria:



Utilizzazione del centro electronic ordering and payment nella terza (sx) e nella quinta (dx) fascia oraria:



Prob. perdita del centro electronic ordering and payment nella terza (sx) e nella quinta (dx) fascia oraria:



Come ipotizzato, dai grafici è possibile osservare come le due metriche peggiorino nella fascia di cena per entrambi i centri.

Progettazione degli esperimenti.

La progettazione degli esperimenti si articola in tre fasi principali:

- 1) Analisi del collo di bottiglia.
- 2) Esecuzione delle simulazioni a orizzonte finito.
- 3) Esecuzione delle simulazioni a orizzonte infinito.

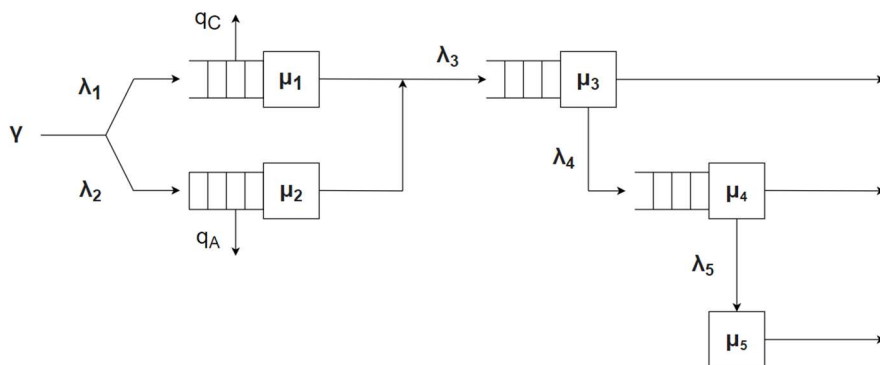
Come per il modello base, abbiamo deciso di eseguire la simulazione a orizzonte finito con l'obiettivo di determinare la configurazione che massimizzi il profitto del sistema.

Siamo partiti dalla configurazione che permetteva di massimizzare il profitto per il modello base. Più precisamente, il numero di serventi dei centri 3, 4 e 5 è rimasto inalterato, mentre il numero di serventi dei

centri 1 e 2 è stato settato a 1. Questa scelta è stata conseguenza della seguente intuizione. La massimizzazione del profitto è in funzione del numero di perdite e, di conseguenza, si dovrà trovare quella configurazione che minimizzi il numero di perdite nei centri 1 e 2. La configurazione dei centri 1 e 2 potrebbe essere differente da quella trovata per il modello base a causa di una capacità finita di tali centri; tuttavia, ipotizziamo che il tasso di uscita dalla sottorete composta dai centri 1 e 2 sia approssimativamente lo stesso di quello che si ha nel modello base nel momento in cui si massimizza il guadagno.

In seguito, siamo ricorsi alle simulazioni a orizzonte infinito per vedere come si comporta il sistema allo stazionario utilizzando le configurazioni finali ricavate con l'orizzonte finito. Abbiamo considerato i centri 3 e 4 concentrandoci sulle metriche di prestazione ρ , $E[T_Q]$, $E[T_S]$. Per i due centri, nelle fasce orarie più rilevanti, abbiamo verificato se le metriche convergono a un valore oppure divergono. Nel primo caso, abbiamo confrontato il valore del QoS con il valore di convergenza. Nel caso in cui divergono, vuol dire che la configurazione trovata a seguito delle simulazioni a orizzonte finito è ottima se dimensionata all'interno della giornata di lavoro ma è inadeguata in un contesto di simulazione più lungo.

Analisi del collo di bottiglia.



λ_i = tasso di ingresso nel centro i

γ = tasso di ingresso nel sistema

Poiché rispetto al modello base cambiano le probabilità di perdita q_C , q_A e i throughput relativi ai centri 1 e 2, le espressioni in funzione di q_C , q_A della domanda media dei singoli centri rimangono le stesse per i centri 3, 4, 5, mentre cambiano per i centri 1 e 2. Infatti:

$$v_1 = [\lambda_1 P_{car} + \lambda_1 (1 - P_{car})(1 - q_C)] / \gamma = 0,04 + 0,36(1 - q_C)$$

$$v_2 = \lambda'_2 / \gamma = \lambda_2 (1 - q_A) / \gamma = 0,6(1 - q_A)$$

In definitiva, la domanda media dei centri risulta essere:

$$D_1 = v_1 S_1 = [3,6 + 32,4(1 - q_C)] \text{ sec}$$

$$D_2 = v_2 S_2 = 108(1 - q_A) \text{ sec}$$

$$D_3 = v_3 S_3 = [9,6 + 86,4(1 - q_C) + 144(1 - q_A)] \text{ sec}$$

$$D_4 = v_4 S_4 = [453,6(1 - q_C) + 756(1 - q_A)] \text{ sec}$$

$$D_5 = v_5 S_5 = [136,08(1 - q_C) + 226,8(1 - q_A)] \text{ sec}$$

Il valore di q_A è stato possibile calcolarlo analiticamente mediante la risoluzione della catena di Markov relativa al centro M/M/1/7, mentre non abbiamo gli strumenti per calcolare q_C a causa della presenza della coda delle macchine al centro 1 e della particolare politica di scheduling ivi adottata. Di conseguenza,

abbiamo utilizzato il valore di q_c ottenuto mediante la simulazione a orizzonte infinito. I valori di q_A , q_c e della domanda media dei vari centri sono mostrati nella seguente tabella:

	09:00 → 11:00	11:00 → 12:00	12:00 → 15:00	15:00 → 18:00	18:00 → 22:00	22:00 → 23:00
q_c	0,000020	0	0,010864	0	0,278181	0,000020
q_A	0,01138873	$2,26247 \times 10^{-5}$	0,4485145	$5,01672 \times 10^{-4}$	0,722248	0,01138873
D_1 (sec)	35,999352	36	35,6480064	36	26,9869356	35,999352
D_2 (sec)	106,770017	107,997557	59,560434	107,945819	29,997216	106,770017
D_3 (sec)	238,358295	239,996742	174,475262	239,927759	111,961449	238,358295
D_4 (sec)	1200,981048	1209,582895	865,5951276	1209,220736	537,3976104	1200,981048
D_5 (sec)	360,294314	362,874869	259,678538	362,766221	161,219283	360,294314

Anche qui, in tutte le fasce orarie il centro 4 (*consumption*) rappresenta il collo di bottiglia, mentre il centro 5 (*playground*) è il secondo centro con domanda media più elevata.

Esecuzione delle simulazioni.

Simulazione a orizzonte finito.

Nella simulazione a orizzonte finito siamo partiti dalle configurazioni qui riportate:

- > Fascia 1: (1, 1, 1, 43, 9)
- > Fascia 2: (1, 1, 1, 43, 9)
- > Fascia 3: (1, 1, 5, 43, 9)
- > Fascia 4: (1, 1, 1, 43, 9)
- > Fascia 5: (1, 1, 8, 43, 9)
- > Fascia 6: (1, 1, 1, 43, 9)

Con tali configurazioni il profitto mensile risultante è di 253.900€, che risulta essere basso a causa dell'elevato numero di perdite nei centri *ordering and payment* ed *electronic ordering and payment*. Infatti, gli intervalli di confidenza della probabilità di perdita in questi due centri sono:

- *Ordering and payment*: 0,152250 +/- 0,013408
- *Electronic ordering and payment*: 0,536718 +/- 0,008230

Come prima cosa abbiamo provato ad aumentare il profitto incrementando il numero di serventi del centro *electronic ordering and payment* con lo scopo di ridurre la probabilità di perdita. La prima fascia oraria di cui abbiamo cambiato il numero di serventi è stata quella relativa alla cena, in quanto è quella che comporta la maggior parte delle perdite. Un ragionamento analogo è stato fatto per il centro *ordering and payment*.

Osservando gli intervalli di confidenza mostrati in precedenza, ci aspettiamo di dover modificare la configurazione per il centro 2 molto più rispetto al centro 1. Effettivamente, variando le configurazioni nelle simulazioni, siamo stati costretti a incrementare il numero di serventi per il centro 2 per la metà delle fasce orarie. Al contrario, è stato necessario incrementare di un'unità il numero di serventi del centro 1 solamente nella fascia relativa all'ora di cena. Con queste modifiche, abbiamo ottenuto i seguenti intervalli di confidenza per la probabilità di perdita per i primi due centri:

- *Ordering and payment*: 0,004766 +/- 0,002080
- *Electronic ordering and payment*: 0,001003 +/- 0,000697

Come si può osservare, la probabilità relativa al centro 1 è diminuita notevolmente. Ciò è dovuto al fatto che la maggior parte delle perdite dal centro 1 occorre nella fascia oraria di cena.

La configurazione finale per i centri *ordering and payment* ed *electronic ordering and payment* è riportata di seguito:

- > Fascia 1: (1, 1, 1, 43, 9)

- > Fascia 2: (1, 1, 1, 43, 9)
- > Fascia 3: (1, 4, 5, 43, 9)
- > Fascia 4: (1, 1, 1, 43, 9)
- > Fascia 5: (2, 7, 8, 43, 9)
- > Fascia 6: (1, 3, 1, 43, 9)

Il profitto che si ha con tali configurazioni è pari a 415.221,25€.

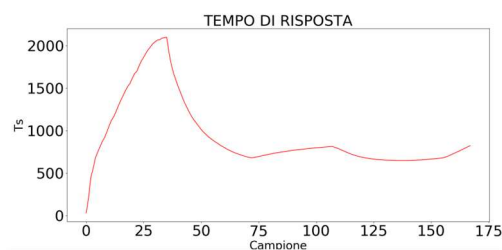
Inoltre, tutti e tre i QoS sono rispettati. Infatti, gli intervalli di confidenza relativi al tempo di risposta del centro 3, al tempo di attesa del centro 4 e alla probabilità di perdita dal centro 5 sono i seguenti:

- Tempo di risposta nel centro *withdraw food*: 824,876598 +/- 56,709237 sec.
- Tempo di attesa nel centro *consumption*: 17,022160 +/- 8,967431 sec.
- Probabilità di perdita nel centro *playground*: 0,157034 +/- 0,012065.

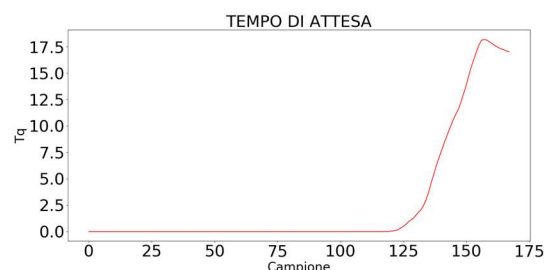
A questo punto, abbiamo provato a diminuire il numero di serventi nei centri 3, 4 e 5 cercando di incrementare ulteriormente il profitto. Tuttavia, decrementando il valore del numero di serventi per un qualsiasi centro e in una qualsiasi fascia oraria, i QoS non sono più contemporaneamente soddisfatti. Osservando le configurazioni finali con le quali si riesce a massimizzare il profitto, possiamo notare che le configurazioni per i centri 3, 4 e 5 sono rimaste inalterate rispetto al modello base. Come descritto nel paragrafo “*Progettazione degli esperimenti*”, probabilmente il flusso di arrivo alla sottorete composta dai centri 3, 4 e 5 è approssimativamente lo stesso di quello che si ha nel modello base nel momento in cui si massimizza il profitto.

Raggiunta la configurazione finale, il tempo medio di risposta nel centro *withdraw food*, il tempo di attesa nel centro *consumption* e la probabilità di perdita nel centro *playground* sono rappresentati dai seguenti grafici.

Tempo medio di risposta giornaliero del centro withdraw food:



Tempo medio di attesa giornaliero del centro consumption:



Probabilità media di perdita giornaliera nel centro playground:

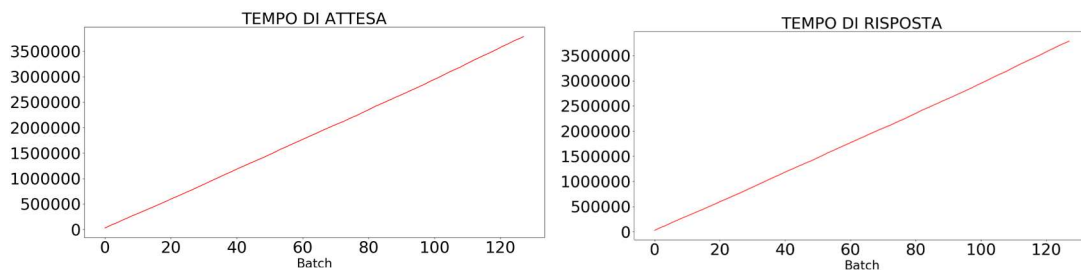


Simulazione a orizzonte infinito.

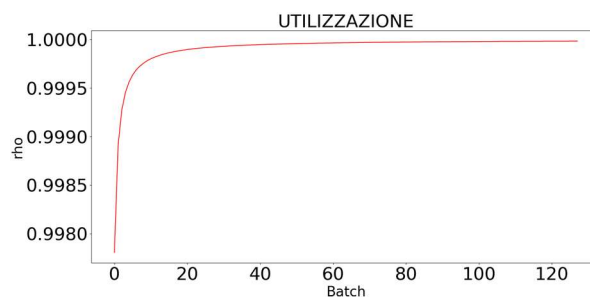
Nella simulazione a orizzonte infinito, abbiamo osservato il comportamento allo stazionario delle metriche ρ , $E[T_Q]$, $E[T_S]$ per i centri 3 e 4 del sistema. Poiché nella simulazione a orizzonte finito le fasce orarie sono molto brevi (durano al più 4 ore), non è detto che le configurazioni che ottimizzano il profitto mensile portino i singoli centri a una condizione di stazionarietà in tutte le fasce orarie per quanto riguarda la simulazione a orizzonte infinito.

Fascia 1 (09:00 → 11:00).

Tempo di attesa (s_x) e di risposta (d_x) del centro withdraw food (configurazione (1, 1, 1, 43, 9)):

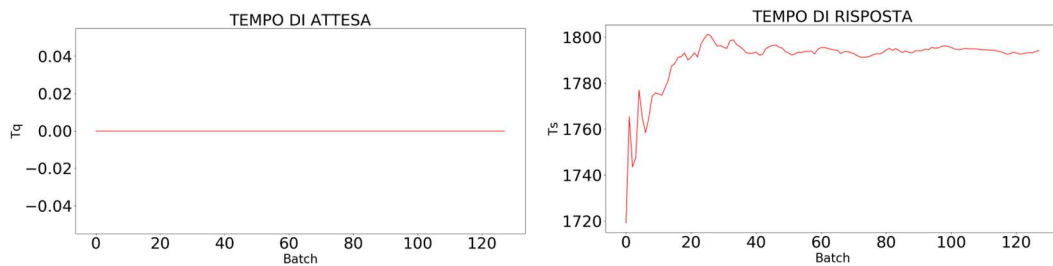


Utilizzazione del centro withdraw food (configurazione (1, 1, 1, 43, 9)):

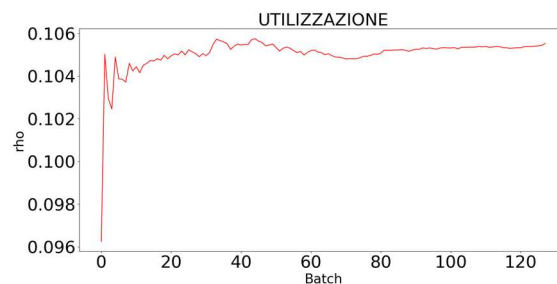


È possibile notare come le metriche prestazionali del centro 3 divergono nella prima fascia oraria. Ciò implica che tale configurazione non potrebbe essere sfruttata nel caso in cui l'intervallo temporale della simulazione fosse molto lungo. Nel caso del McDonald's, invece, è una situazione che si ha per 120 minuti (dalle 09:00 alle 11:00), all'interno dei quali arrivano mediamente al più $120 \times 1/3 = 40$ job (dove $1/3$ job/min è il tasso di arrivo al ristorante nella prima fascia oraria). Questo significa che, alla fine della fascia oraria, in coda si trovano relativamente pochi job poiché siamo ancora all'inizio della fase di crescita delle code. Di conseguenza, nelle fasce orarie successive sarà necessario smaltire il carico di lavoro che è stato accumulato all'ora di colazione.

Tempo di attesa (sx) e di risposta (dx) del centro consumption (configurazione (1, 1, 1, 43, 9)):



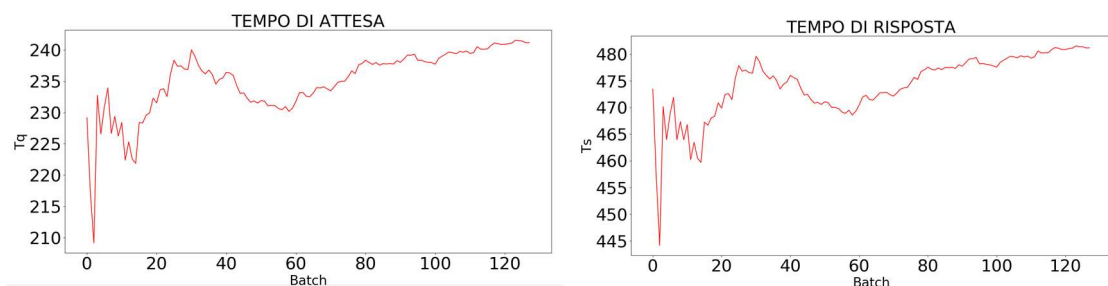
Utilizzazione del centro consumption (configurazione (1, 1, 1, 43, 9)):



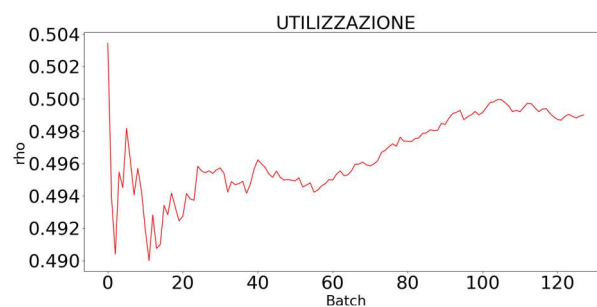
Per quanto riguarda il centro 4, invece, non essendoci alcuna attesa, il QoS sul tempo di attesa è senz'altro rispettato per la sola fascia oraria 1. Le metriche di prestazione sono molto buone perché viene utilizzato un numero di serveri tarato per l'ora di cena.

Fascia 2 (11:00 → 12:00).

Tempo di attesa (sx) e di risposta (dx) del centro withdraw food (configurazione (1, 1, 1, 43, 9)):

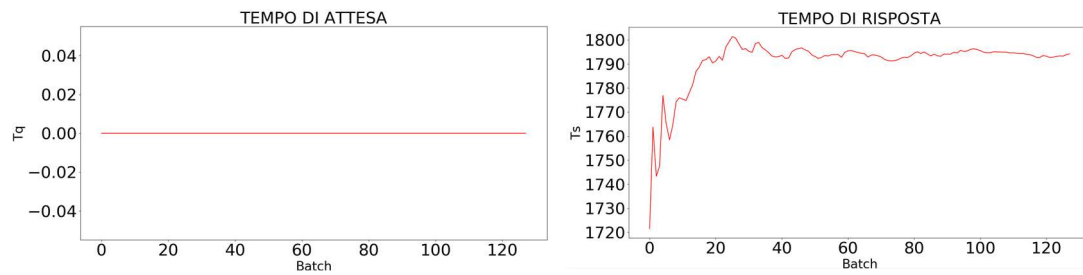


Utilizzazione del centro withdraw food (configurazione (1, 1, 1, 43, 9)):

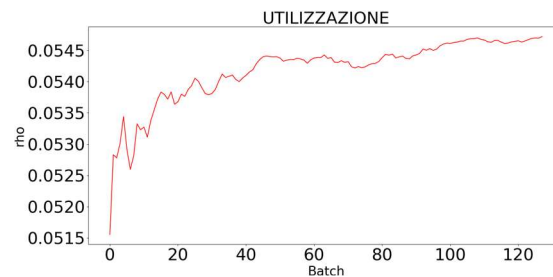


Come si può vedere, l'utilizzazione nella seconda fascia oraria non è molto elevata e di conseguenza si riesce a smaltire il carico di lavoro accumulato in precedenza. Si può anche notare come il tempo medio di risposta allo stazionario converga a circa 8 minuti, che sono molti di meno rispetto ai 15 dettati dal QoS.

Tempo di attesa (sx) e di risposta (dx) del centro consumption (configurazione (1, 1, 1, 43, 9)):



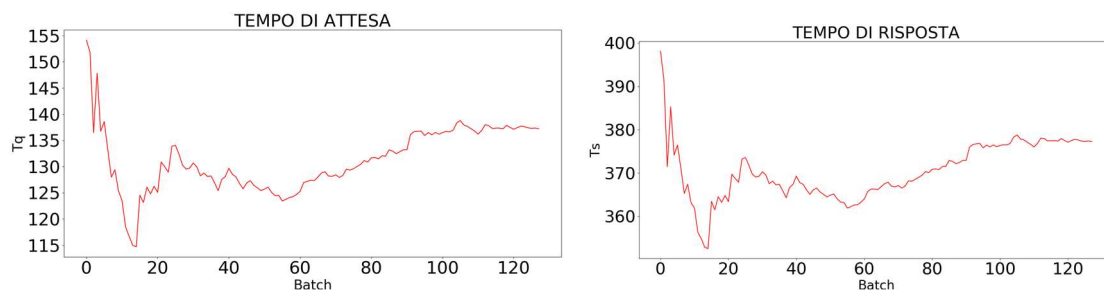
Utilizzazione del centro consumption (configurazione (1, 1, 1, 43, 9)):



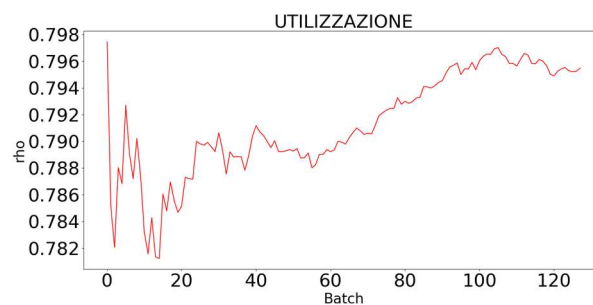
Per quanto riguarda il centro 4, valgono le medesime osservazioni fatte per la prima fascia oraria.

Fascia 3 (12:00 → 15:00).

Tempo di attesa (sx) e di risposta (dx) del centro withdraw food (configurazione (1, 4, 5, 43, 9)):

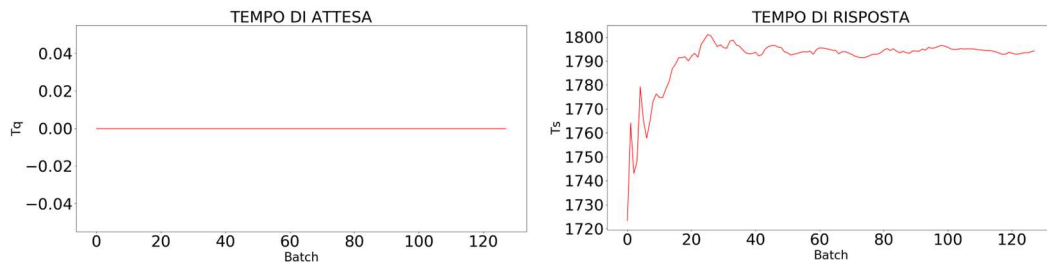


Utilizzazione del centro withdraw food (configurazione (1, 4, 5, 43, 9)):

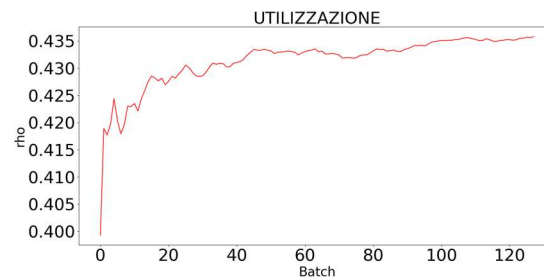


Anche qui il tempo medio di risposta allo stazionario, essendo pari a poco più di 6 minuti, rientra nei 15 minuti dettati dal QoS.

Tempo di attesa (sx) e di risposta (dx) del centro consumption (configurazione (1, 4, 5, 43, 9)):



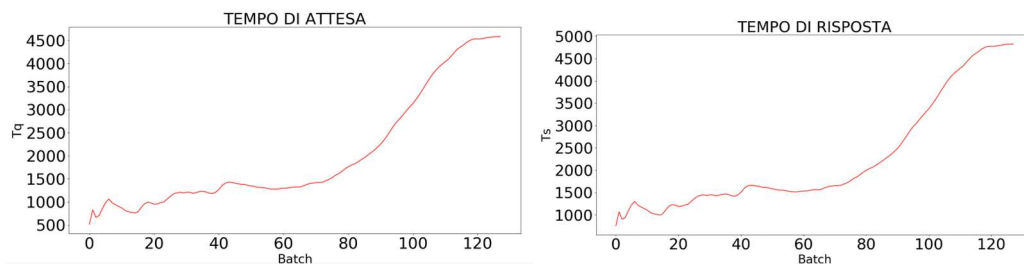
Utilizzazione del centro consumption (configurazione (1, 4, 5, 43, 9)):



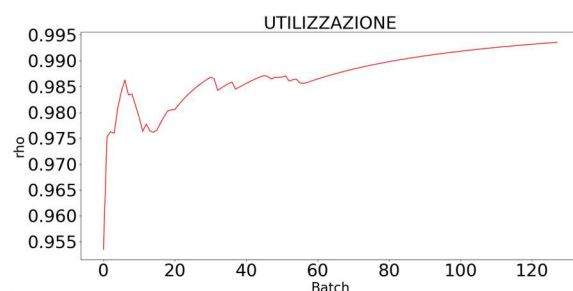
Anche qui, per il centro *consumption*, valgono le medesime osservazioni fatte per la prima fascia oraria.

Fascia 5 (18:00 → 22:00).

Tempo di attesa (sx) e di risposta (dx) del centro withdraw food (configurazione (2, 7, 8, 43, 9)):

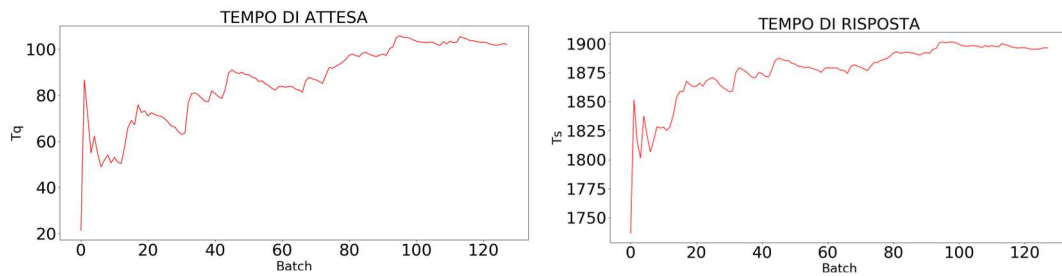


Utilizzazione del centro withdraw food (configurazione (2, 7, 8, 43, 9)):

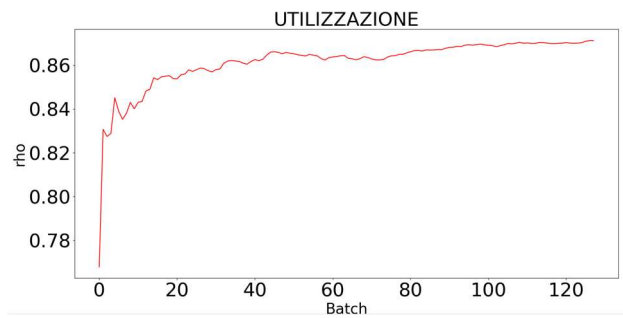


Dal grafico osserviamo che il tempo medio di risposta converge a un valore molto lontano da quello specificato dal QoS. Inoltre, la convergenza è piuttosto lenta (si può vedere come avvenga dopo il centesimo batch), per cui i valori del tempo medio di attesa e di risposta relativi all'inizio della simulazione sono bassi rispetto ai valori teorici. Nella pratica, questo significa che si è utilizzata una configurazione certamente buona per degli scenari di simulazione sufficientemente brevi, come la quarta fascia oraria della giornata di lavoro, ma molto meno buona per degli scenari di simulazione particolarmente lunghi.

Tempo di attesa (sx) e di risposta (dx) del centro consumption (configurazione (2, 7, 8, 43, 9)):



Utilizzazione del centro consumption (configurazione (2, 7, 8, 43, 9)):



Dai grafici è possibile osservare che solamente nella fascia relativa all'ora di cena, allo stazionario, non viene rispettato il QoS associato al tempo di attesa nel centro 4. Di conseguenza, questa configurazione non può essere utilizzata per intervalli di tempo molto lunghi se si vuole rispettare il QoS.