

Simulazione PMCSN

Luca Mastrobattista, 0292461

Indice

1	Traccia della Simulazione	2
1.1	Caso di studio	2
1.2	Obiettivi	2
2	Modello concettuale	3
2.1	Visualizzazione grafica	3
2.2	Eventi del sistema e variabili di stato	3
2.3	Eventi	3
2.4	Variabili di stato	3
3	Modello delle specifiche	4
3.1	Periodo di osservazione	4
3.2	Distribuzione degli arrivi	4
3.3	Assunzioni	5
3.4	Guadagni e costi	5
4	Modello analitico	5
5	Modello computazionale	6
5.1	Makefile	7
5.2	Dipendenze	7
6	Verifica e validazione	7
6.1	Tempo di interarrivo medio	8
6.2	Numero medio job in coda e tempo medio in coda	8
6.3	Tempo di risposta e numero medio di job nel nodo	12
6.4	Utilizzazione	12
7	Conclusioni	13
8	Immagini	13
8.1	Distribuzioni gaussiane per gli arrivi	13

1 Traccia della Simulazione

1.1 Caso di studio

Si vuole valutare l'idea di aprire un locale in un piccolo paese. L'attività dovrà offrire ai clienti servizi di bar e di pizzeria. Il locale è già provvisto di tutto l'arredamento e lo si affitterà per un costo 1500 € al mese. Il pizzaiolo scelto per i servizi di pizzeria ha comunicato che, nel forno presente, si possono preparare contemporaneamente al massimo 2 pizze, ognuna delle quali può essere preparata con un tempo medio di 3 minuti. Inoltre, si è già trovato un accordo con lui: lavorerà ogni giorno della settimana dalle ore 19:00 alle ore 23:00, percependo una paga di 50 € al giorno con il vincolo che tutte le ordinazioni arrivate precedentemente alle 23:00 verranno sempre completate, anche se per farlo dovrà continuare a sfornare pizze oltre questo orario. Per quanto riguarda le richieste al bar, si vuole che "l'ultimo giro" venga chiamato alle ore 03:00, senza accettare altre richieste successive a quell'orario ma completando tutte quelle ancora presenti.

Dopo un'osservazione settimanale di altri locali che offrono servizi simili, si è notato che il numero di clienti che arrivano al locale si differenzia per fasce orarie della giornata diverse. Inoltre, nel fine settimana, la frequenza delle richieste nelle fasce orarie identificate è maggiore rispetto a quella settimanale. Infine, si è osservato che nella fascia oraria tra le 15:00 e le 18:00 le richieste sono talmente poche che non è conveniente mantenere il locale aperto. Si riportano di seguito delle tabelle riassuntive per le frequenze di arrivo:

Fascia oraria	$\lambda_{B,W}$	$\lambda_{P,W}$
07:00 → 11:00	30 j/h	✗
11:00 → 15:00	12.5 j/h	✗
15:00 → 18:00	✗	✗
18:00 → 19:00	25 j/h	✗
19:00 → 23:00	12.5 j/h	10 j/h
23:00 → 02:00	10 j/h	✗

Frequenze di arrivo settimanali

Fascia oraria	$\lambda_{B,WE}$	$\lambda_{P,WE}$
07:00 → 13:00	30 j/h	✗
13:00 → 18:00	20 j/h	✗
15:00 → 18:00	✗	✗
18:00 → 19:00	45 j/h	✗
19:00 → 23:00	22.5 j/h	30 j/h
23:00 → 02:00	20 j/h	✗

Frequenze di arrivo fine-settimanali

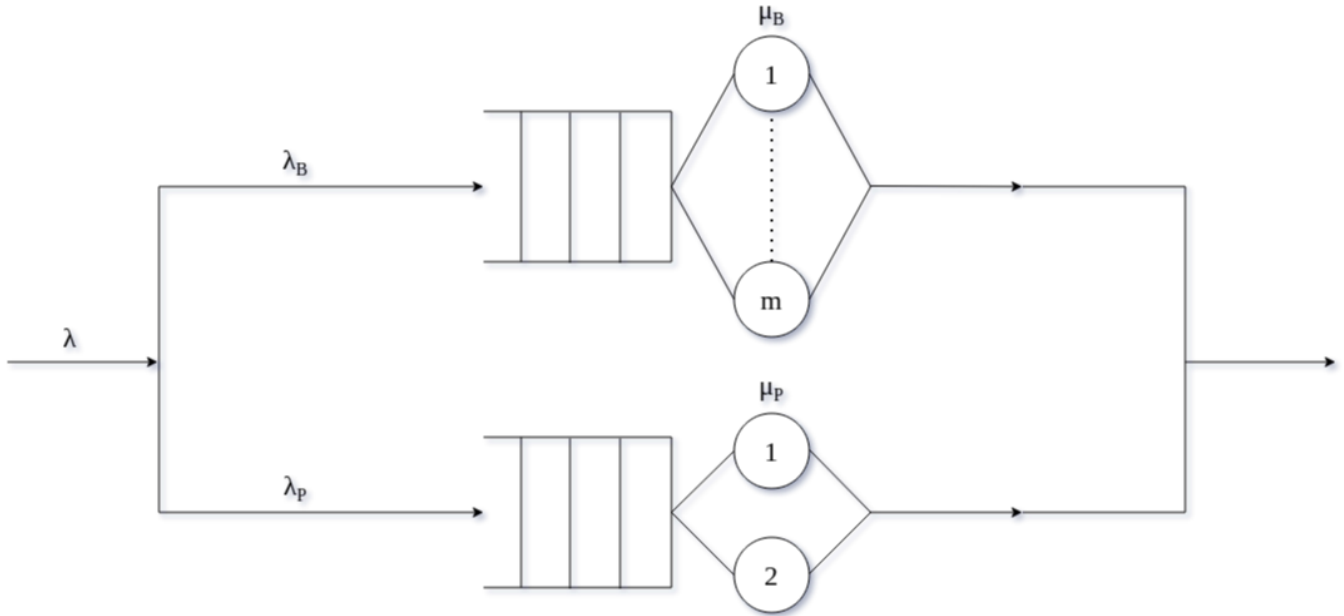
1.2 Obiettivi

Obiettivo dell'analisi è la valutazione del numero di baristi da assumere, col fine ultimo di massimizzare i guadagni. Si considera una paga di 40 € al giorno per ognuno di loro, assumendo per loro turni di 8 ore. Si assume che ogni barista sia in grado di servire un'ordinazione in 2 minuti, durante i quali si dedica esclusivamente a quella richiesta. Si assume inoltre che il prezzo medio delle richieste di tipo B sia di 5 €, mentre quello delle richieste di tipo P sia di 7 €. Si vuole, però, che i seguenti vincoli siano sempre rispettati:

- Ogni ordinazione al bar deve essere servita in un tempo strettamente minore di 3 minuti;
- Ogni ordinazione per la pizzeria sia servita in un tempo strettamente minore di 10 minuti

2 Modello concettuale

2.1 Visualizzazione grafica



La frequenza di arrivo λ si compone della frequenza di arrivo λ_B e λ_P , che sono rispettivamente i tassi di arrivo per richieste al bar e alla pizzeria. Una opportuna coda per ogni tipologia rappresenta la lista di attesa della tipologia stessa. Ogni servente di tipo B rappresenta un barista assunto, che lavora con una frequenza μ_B . Ogni servente di tipo P , invece, rappresenta una delle due richieste che il pizzaiolo è in grado di gestire contemporaneamente.

2.2 Eventi del sistema e variabili di stato

2.3 Eventi

Indice	Descrizione	Attributo 1	Attributo 2
0	Arrivo di tipo B	t	x
1	Completamento dal server B_1	t	x
..	..	t	x
m	Completamento dal server B_m	t	x
m + 1	Arrivo di tipo P	t	x
m + 2	Completamento dal server P_1	t	x
m + 3	Completamento dal server P_2	t	x
m + 4	Evento di campionamento	t	x

L'attributo t identifica il tempo schedulato per la successiva occorrenza dell'evento di quel tipo; l'attributo x identifica lo stato di attività dell'evento.

2.4 Variabili di stato

- $l_B(t)$: numero di richieste di tipo B al centro all'istante t
- $l_P(t)$: numero di richieste di tipo P al centro all'istante t

- $X_s(t)$: stato del server s all'istante t , con $s \in \mathcal{B} \cup \mathcal{P}$, dove $\mathcal{B} \cup \mathcal{P}$ è l'insieme dei server di tipo B unito all'insieme dei server di tipo P .

$$X_s(t) = \begin{cases} 1 & \text{se server } s \text{ è occupato} \\ 0 & \text{altrimenti} \end{cases}$$

3 Modello delle specifiche

3.1 Periodo di osservazione

Il periodo di osservazione è quello di un intero anno e ogni giorno si osserva l'intera giornata lavorativa costituita dalle due fasce orarie riportate precedentemente nelle *tabelle riassuntiva dei tassi di arrivo*.

3.2 Distribuzione degli arrivi

I valori dei vari tassi di arrivo sono stati raccolti analizzando un caso reale, anche se si tratta comunque di una stima. Per rappresentare il processo degli arrivi è stata utilizzata la distribuzione esponenziale, utilizzando λ diversi per ogni fascia oraria. Inoltre, all'interno della singola fascia oraria, gli arrivi potrebbero essere modellati come una distribuzione gaussiana, centrata attorno all'ora in cui le richieste sono più probabili. A partire da questa osservazione, si è scelto di utilizzare la distribuzione esponenziale per modellare gli arrivi, ma la media utilizzata è pesata opportunamente per una probabilità che è tanto più alta quanto più il tempo di simulazione è vicino all'ora di massima affluenza per quella fascia oraria. Per modellare questo, si definiscono delle frequenze di interarrivo medie per ogni fascia oraria, riportate qui in minuti:

Fascia oraria	$\lambda_{B,W}$	$\lambda_{P,W}$
07:00 → 11:00	0.5 j/min	✗
11:00 → 15:00	0.21 j/min	✗
18:00 → 19:00	0.42 j/min	✗
19:00 → 23:00	0.21 j/min	0.17 j/min
23:00 → 02:00	0.17 j/min	✗

Frequenze di arrivo settimanali

Fascia oraria	$\lambda_{B,WE}$	$\lambda_{P,WE}$
07:00 → 11:00	0.5 j/min	✗
11:00 → 15:00	0.34 j/min	✗
18:00 → 19:00	0.75 j/min	✗
19:00 → 23:00	0.375 j/min	0.5 j/min
23:00 → 02:00	0.34 j/min	✗

Frequenze di arrivo fine-settimanali

Per ogni fascia oraria si definisce una distribuzione di probabilità gaussiana:

Fascia oraria	μ	σ
07:00 → 11:00	8	1.2
11:00 → 15:00	13.5	2
18:00 → 19:00	18.5	0.4
19:00 → 23:00	22.5	2
23:00 → 02:00	24	0.9

Parametri delle gaussiane per richieste di tipo B

Fascia oraria	μ	σ
19:00 → 23:00	20.5	1

Parametri delle gaussiane per richieste di tipo P

La loro rappresentazione grafica è riportata in fondo al documento.

Ora, supponiamo di essere all'istante di simulazione t_0 di un giorno settimanale, nella prima fascia oraria; in questo caso $\lambda = 0.5 \text{ j/min}$. Per generare il prossimo tempo di interarrivo, si usa:

$$\text{Exponential}(1/\lambda) * f^n(t_0)$$

dove $f^n(t_0)$ è il valore della distribuzione normale relativa alla fascia oraria valutata in t_0 e normalizzata rispetto alla fascia oraria. Nell'esempio:

$$f^n(t_0) = \frac{f(t_0)}{F(11) - F(7)}$$

con $F(x)$ funzione cumulativa della distribuzione gaussiana relativa alla fascia oraria 07:00 → 11:00.

Non si è usata una gaussiana direttamente come distribuzione del tempo di interarrivo perché avrebbe modellato una cosa diversa: in quel caso i tempi sarebbero molto più vicini al valor medio della distribuzione all'interno dell'intera fascia in esame, invece si vuole modellare che il tempo di interarrivo diminuisce in un intorno di un certo tempo.

La simulazione permette di disabilitare questa opzione inserendo l'opportuno *flag* nel lancio del programma.

3.3 Assunzioni

- Stato iniziale vuoto:

$$l_B(0) + l_P(0) = P(0) + B(0) = 0$$

Come conseguenza, il primo evento deve essere necessariamente un arrivo e, in particolare, è un arrivo di tipo B : la pizzeria apre alle 19.

- Stato finale di ogni giorno vuoto:

$$X_s(T) = 0 \quad \forall s \in \mathcal{B} \cup \mathcal{P}$$

Con T tempo di chiusura giornaliero e $\mathcal{B} \cup \mathcal{P}$ l'unione dell'insieme dei serventi di tipo B e P . Come conseguenza, l'ultimo evento non può essere un arrivo, e sarà quindi o una partenza o un campionamento.

- I tempi di servizio di ognuno dei serventi si assumono esponenziali e indipendenti dalla fascia oraria. In particolare, ogni servente di tipo B lavora con frequenza media pari a $\mu_B = \frac{1}{2} j/min$; ogni servente di tipo P lavora con frequenza media pari a $\mu_P = \frac{1}{3} j/min$.

3.4 Guadagni e costi

4 Modello analitico

Di seguito sono rappresentate le formule utilizzate per l'analisi teorica del sistema. In particolare, tutti i centri sono stati modellati come M/M/m per i quali si hanno le seguenti formule teoriche:

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

$$E(TS) = E(TQ) + E(S_i)$$

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

$$E(NQ)_{Erlang} = \frac{P_Q \rho}{1-\rho}$$

$$E(TQ) = \frac{P_Q E(S)}{1-\rho}$$

$$E(NS) = E(NQ) + m\rho$$

5 Modello computazionale

Il modello computazionale è stato sviluppato in *Python* ed è il programma `simulation.py`; i parametri configurabili sono definiti invece nel file `configurations/Config.py`. Il programma è lanciabile da riga comando con diversi *flag* che ne influenzano il comportamento, listabili con il comando: `python simulation.py -h` (oppure `python simulation.py --help`), ottenendo il seguente messaggio:

```
~/Scrivania/PMCSN/PMCSN-simulation (main*) » python simulation.py -h
usage: simulation.py [-h] [-cf FILEPATH] [-scf FILEPATH] [-fh] [-ih] [-cc OPTION VALUE] [-fb THRESHOLD] [-s SEED] [-ngf] [-of FILE] [-wed]

PMCSN project command line interface

options:
  -h, --help            show this help message and exit
  -cf FILEPATH, --configFile FILEPATH
                        specify a configuration file to load
  -scf FILEPATH, --storeConfigFile FILEPATH
                        specify an output file where to store config
  -fh, --finite_horizont
                        simulate a finite horizon case
  -ih, --infinite_horizont
                        simulate an infinite horizon case. Using this, gaussian factor is automatically disabled
  -cc OPTION VALUE, --change_config OPTION VALUE
                        specify configuration to change
  -fb THRESHOLD, --find_b_value THRESHOLD
                        find the value of b such that autocorrelation lag j=1 is <= THRESHOLD
  -s SEED, --seed SEED  use the given SEED as random seed. if SEED = 0 then the initial seed is to be supplied interactively; if SEED < 0
                        then the initial seed is obtained from the system clock; if SEED > 0 > 0 then it is the initial seed (unless too
                        large). default value is 0
  -ngf, --no_gaussian_factor
                        don't use the gaussian probability value to weight interarrival times
  -of FILE, --output_file FILE
                        save output in ./output/OUTPUTFILE.csv. The file format .csv is added if not already present
  -wed, --weekend_day
                        simulate a weekend day, with the proper system variables. Default this option is disabled, meaning that a week day
                        is simulate
```

In particolare, col flag `-cc` è possibile cambiare una qualsiasi impostazione presente nel file di configurazione specificandone il valore a riga comando, mentre col flag `-ngf` (cioè `-no_gaussian_factor`) si disabilita la normalizzazione del tempo di interarrivo col fattore gaussiano determinato secondo le modalità descritte nella [sezione dedicata alla distribuzione degli arrivi](#).

Il programma è realizzato seguendo l'approccio della *next-event simulation* e perciò sono state create opportune classi python, memorizzate ognuna in un file diverso.

Le funzioni per processare gli arrivi sono `processArrivalB()` e `processArrivalP()` al cui interno è implementata la logica per recuperare il λ corretto in base alla fascia oraria, il cui inverso va usato come parametro per la funzione `getArrivalB(m)` e `getArrivalP(m)` dove viene invocata la `Exponential(m)` dopo aver selezionato opportunamente uno *stream* diverso. Le funzioni per processare le partenze sono `processDepartureB()`, `processDepartureP()`.

La funzione per la selezione del server è diversa per richieste di tipo *B* e *P*: si tratta di `FindOneB()` e `FindOneP()`. Queste funzioni utilizzano politiche di assegnazione ai server diverse: nel primo caso la politica è di *equity*, ricercando quindi il server che risulta libero da più tempo, nel secondo, invece, dovendo simulare un pizzaiolo che può infornare contemporaneamente 2 pizze, la richiesta è assegnata al primo server libero di tipo *P* che si trova, scandendo la lista in ordine crescente.

I vari *sample* che vengono raccolti durante la simulazione vengono memorizzati in una opportuna istanza di `SamplingList`, il cui metodo `append()`, oltre ad aggiungere il nuovo campione in fondo alla lista, implementa il *one-pass alghoritm* secondo *Welford* per ognuna delle grandezze raccolte; è quindi necessario, a fine simulazione, invocare i metodi `makeCorrectVariance` che divide per il numero di campioni tutte le varianze e `computeConfidenceInterval` per calcolare l'intervallo di confidenza per ognuna delle grandezze.

5.1 Makefile

Nella directory principale è presente un `Makefile` che aiuta ad eseguire correttamente il programma. Usando le configurazioni predisposte, nessun parametro di configurazione viene modificato: il flag `-cc` non viene mai usato. In particolare, il comando `make clean` elimina tutti i file di output `.csv` generati da precedenti esecuzioni. Il comando `make`, che corrisponde a `make all`, esegue, in ordine:

```
rm -f ./output/*.csv
rm -f ./output/finite/*.csv
rm -f ./output/infinite/*.csv
python3 simulation.py -ih -s 123 -of week
python3 simulation.py -ih -s 123 -wed -of weekend
python3 simulation.py -fh -s 123 -of week_gauss
python3 simulation.py -fh -s 123 -of week -ngf
python3 simulation.py -fh -s 123 -of weekend_gauss -wed
python3 simulation.py -fh -s 123 -of weekend -wed -ngf
```

Questo comando è quello utilizzato per generaer gli output che verranno successivamente discussi.

5.2 Dipendenze

Le librerie esterne importate, di cui il programma necessita per funzionare, sono le seguenti e sono riportate anche nel file `dipendenze.txt`:

- `copy`
- `argparse`
- `importlib`
- `ast`
- `math`
- `scipy.stats`
- `time`

6 Verifica e validazione

I risultati ottenuti dall'esecuzione di *simulation.py* con $m = 2$ e inserendo come seme il valore *123* su un periodo di 365 giorni sono i seguenti:

Per le richieste di tipo B

Grandezza	Tempistica
$\frac{1}{\lambda_{\text{medio}}}$	5.22 min
$E[T_S]$	2.18
$E[N]$	0.42
$E[T_Q]$	0.18
$E[N_Q]$	0.03

Per le richieste di tipo P

Grandezza	Tempistica
$\frac{1}{\lambda_{\text{medio}}}$	3.84 min
$E[T_S]$	4.51
$E[N]$	1.17
$E[T_Q]$	1.51
$E[N_Q]$	0.39

Statistiche dei server B

Numero del server	Utilizzazione	$E[S]$	Share
1	0.190	1.98	0.500
2	0.190	1.98	0.500

Statistiche dei server P

Numero del server	Utilizzazione	$E[S]$	Share
4	0.392	2.99	0.503
5	0.387	2.99	0.497

6.1 Tempo di interarrivo medio

Per calcolare la frequenza di arrivo media, consideriamo come arco di tempo di riferimento una settimana.

Nei giorni settimanali, il numero medio di job che arrivano è:

$$E[N] = \frac{1}{6} \cdot 360 + \frac{1}{12} \cdot 300 + \frac{1}{6} \cdot 120 + \frac{1}{4} \cdot 120 + \frac{1}{6} \cdot 240 = 175 \text{ job}$$

Nei giorni fine-settimanali, il numero medio di job che arrivano è:

$$E[N] = \frac{1}{4} \cdot 360 + \frac{1}{6} \cdot 300 + \frac{1}{3} \cdot 120 + \frac{1}{4} \cdot 120 + \frac{1}{2} \cdot 240 = 330 \text{ job}$$

Quindi in media in una settimana arrivano:

$$E[N]_{\text{settimana}} = 175 \cdot 5 + 330 \cdot 2 = 1535 \text{ job}$$

Ora, per trovare la frequenza di arrivo media giornaliera in minuti, dividiamo per $7 \cdot 19h \cdot 60min$, ottenendo:

$$\lambda_{\text{medio, min}} = \frac{1535}{7d \cdot 19h \cdot 60min} = 0.19235 \text{ job/min}$$

L'inverso di questo valore è l'interarrivo medio in minuti, ed è:

$$\frac{1}{\lambda_{\text{medio, min}}} = 5.19 \text{ min}$$

6.2 Numero medio job in coda e tempo medio in coda

Per calcolare il tempo medio in coda per un multiserver abbiamo bisogno della formula *Erlang-C*. Si definiscono:

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

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

$$E(T_Q) = \frac{PQ \cdot E(S)}{1 - \rho}$$

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

$$E[S_i] = \frac{1}{\mu}$$

$$E[S] = \frac{E[S_i]}{m} = \frac{1}{m \cdot \mu}$$

Quindi, ponendo $\mu = \frac{1}{2}$, si ha che, per ogni fascia oraria nei giorni settimanali:

- $7 \rightarrow 13$

$$E[S] = \frac{1}{2 \cdot \mu} = \frac{2}{2} = 1 \text{ min}$$

$$E[S_i] = \frac{1}{\mu} = 2 \text{ min}$$

$$\rho_{7 \rightarrow 13} = \frac{\lambda_{7 \rightarrow 13}}{m \cdot \mu} = \frac{\lambda_{7 \rightarrow 13}}{2 \cdot \frac{1}{2}} = \lambda_{7 \rightarrow 13} = 0.17$$

$$P(0)_{7 \rightarrow 13} = .. = 0.71$$

$$PQ_{7 \rightarrow 13} = .. = 0.05$$

$$E[T_{Q_{7 \rightarrow 13}}] = .. = 0.06 \text{ min}$$

$$E[N_{Q_{7 \rightarrow 13}}] = \lambda_{7 \rightarrow 13} \cdot E[T_{Q_{7 \rightarrow 13}}] = 0.01 \text{ job}$$

- $13 \rightarrow 18$

$$\rho_{13 \rightarrow 18} = \frac{\lambda_{13 \rightarrow 18}}{m \cdot \mu} = \frac{\lambda_{13 \rightarrow 18}}{2 \cdot \frac{1}{2}} = \lambda_{13 \rightarrow 18} = 0.17$$

$$P(0)_{13 \rightarrow 18} = .. = 0.71$$

$$PQ_{13 \rightarrow 18} = .. = 0.05$$

$$E[T_{Q_{13 \rightarrow 18}}] = .. = 0.06 \text{ min}$$

$$E[N_{Q_{13 \rightarrow 18}}] = \lambda_{13 \rightarrow 18} \cdot E[T_{Q_{13 \rightarrow 18}}] = 0.01 \text{ job}$$

- $18 \rightarrow 20$

$$\rho_{18 \rightarrow 20} = \frac{\lambda_{18 \rightarrow 20}}{m \cdot \mu} = \frac{\lambda_{18 \rightarrow 20}}{2 \cdot \frac{1}{2}} = \lambda_{18 \rightarrow 20} = 0.17$$

$$P(0)_{18 \rightarrow 20} = .. = 0.71$$

$$PQ_{18 \rightarrow 20} = .. = 0.05$$

$$E[T_{Q_{18 \rightarrow 20}}] = .. = 0.06 \text{ min}$$

$$E[N_{Q_{18 \rightarrow 20}}] = \lambda_{18 \rightarrow 20} \cdot E[T_{Q_{18 \rightarrow 20}}] = 0.01 \text{ job}$$

- $20 \rightarrow 22$

$$\rho_{20 \rightarrow 22} = \frac{\lambda_{20 \rightarrow 22}}{m \cdot \mu} = \frac{\lambda_{20 \rightarrow 22}}{2 \cdot \frac{1}{2}} = \lambda_{20 \rightarrow 22} = 0.17$$

$$P(0)_{20 \rightarrow 22} = .. = 0.71$$

$$PQ_{20 \rightarrow 22} = .. = 0.05$$

$$E[T_{Q_{20 \rightarrow 22}}] = .. = 0.06 \text{ min}$$

$$E[N_{Q_{20 \rightarrow 22}}] = \lambda_{20 \rightarrow 22} \cdot E[T_{Q_{20 \rightarrow 22}}] = 0.01 \text{ job}$$

- $22 \rightarrow 2$

$$\rho_{22 \rightarrow 2} = \frac{\lambda_{22 \rightarrow 2}}{m \cdot \mu} = \frac{\lambda_{22 \rightarrow 2}}{2 \cdot \frac{1}{2}} = \lambda_{22 \rightarrow 2} = 0.17$$

$$P(0)_{22 \rightarrow 2} = .. = 0.71$$

$$PQ_{22 \rightarrow 2} = .. = 0.05$$

$$E[T_{Q_{22 \rightarrow 2}}] = .. = 0.06 \text{ min}$$

$$E[N_{Q_{22 \rightarrow 2}}] = \lambda_{22 \rightarrow 2} \cdot E[T_{Q_{22 \rightarrow 2}}] = 0.01 \text{ job}$$

Per calcolare il numero medio di job in coda in settimana, si sommano i risultati ottenuti in ogni fascia oraria pesati per la durata della fascia stessa e infine si divide per la durata del giorno lavorativo:

$$E[N_{Q_{\text{giorno-settimana}}}] = \frac{E[N_{Q_{7 \rightarrow 13}}] \cdot 360 + E[N_{Q_{13 \rightarrow 18}}] \cdot 300}{19 \cdot 60} + \frac{E[N_{Q_{18 \rightarrow 20}}] \cdot 120 + E[N_{Q_{20 \rightarrow 22}}] \cdot 120}{19 \cdot 60} + \frac{E[N_{Q_{22 \rightarrow 2}}] \cdot 240}{19 \cdot 60} = 0.01 \text{ job}$$

Per i giorni fine-settimanali, invece:

- $7 \rightarrow 13$

$$E[S] = \frac{1}{2 \cdot \mu} = \frac{2}{2} = 1 \text{ min}$$

$$E[S_i] = \frac{1}{\mu} = 2 \text{ min}$$

$$\rho_{7 \rightarrow 13} = \frac{\lambda_{7 \rightarrow 13}}{m \cdot \mu} = \frac{\lambda_{7 \rightarrow 13}}{2 \cdot \frac{1}{2}} = \lambda_{7 \rightarrow 13} = 0.25$$

$$P(0)_{7 \rightarrow 13} = .. = 0.60$$

$$PQ_{7 \rightarrow 13} = .. = 0.10$$

$$E[T_{Q_{7 \rightarrow 13}}] = .. = 0.13 \text{ min}$$

$$E[N_{Q_{7 \rightarrow 13}}] = \lambda_{7 \rightarrow 13} \cdot E[T_{Q_{7 \rightarrow 13}}] = 0.03 \text{ job}$$

• 13 → 18

$$\rho_{13 \rightarrow 18} = \frac{\lambda_{13 \rightarrow 18}}{m \cdot \mu} = \frac{\lambda_{13 \rightarrow 18}}{2 \cdot \frac{1}{2}} = \lambda_{13 \rightarrow 18} = 0.17$$

$$P(0)_{13 \rightarrow 18} = .. = 0.71$$

$$PQ_{13 \rightarrow 18} = .. = 0.05$$

$$E[T_{Q_{13 \rightarrow 18}}] = .. = 0.06 \text{ min}$$

$$E[N_{Q_{13 \rightarrow 18}}] = \lambda_{13 \rightarrow 18} \cdot E[T_{Q_{13 \rightarrow 18}}] = 0.01 \text{ job}$$

• 18 → 20

$$\rho_{18 \rightarrow 20} = \frac{\lambda_{18 \rightarrow 20}}{m \cdot \mu} = \frac{\lambda_{18 \rightarrow 20}}{2 \cdot \frac{1}{2}} = \lambda_{18 \rightarrow 20} = 0.33$$

$$P(0)_{18 \rightarrow 20} = .. = 0.50$$

$$PQ_{18 \rightarrow 20} = .. = 0.17$$

$$E[T_{Q_{18 \rightarrow 20}}] = .. = 0.25 \text{ min}$$

$$E[N_{Q_{18 \rightarrow 20}}] = \lambda_{18 \rightarrow 20} \cdot E[T_{Q_{18 \rightarrow 20}}] = 0.08 \text{ job}$$

• 20 → 22

$$\rho_{20 \rightarrow 22} = \frac{\lambda_{20 \rightarrow 22}}{m \cdot \mu} = \frac{\lambda_{20 \rightarrow 22}}{2 \cdot \frac{1}{2}} = \lambda_{20 \rightarrow 22} = 0.25$$

$$P(0)_{20 \rightarrow 22} = .. = 0.60$$

$$PQ_{20 \rightarrow 22} = .. = 0.10$$

$$E[T_{Q_{20 \rightarrow 22}}] = .. = 0.13 \text{ min}$$

$$E[N_{Q_{20 \rightarrow 22}}] = \lambda_{20 \rightarrow 22} \cdot E[T_{Q_{20 \rightarrow 22}}] = 0.03 \text{ job}$$

- $22 \rightarrow 2$

$$\rho_{22 \rightarrow 2} = \frac{\lambda_{22 \rightarrow 2}}{m \cdot \mu} = \frac{\lambda_{22 \rightarrow 2}}{2 \cdot \frac{1}{2}} = \lambda_{22 \rightarrow 2} = 0.50$$

$$P(0)_{22 \rightarrow 2} = \dots = 0.33$$

$$PQ_{22 \rightarrow 2} = \dots = 0.33$$

$$E[T_{Q_{22 \rightarrow 2}}] = \dots = 0.67 \text{ min}$$

$$E[N_{Q_{22 \rightarrow 2}}] = \lambda_{22 \rightarrow 2} \cdot E[T_{Q_{22 \rightarrow 2}}] = 0.33 \text{ job}$$

Per calcolare il numero medio di job in coda in settimana, si sommano i risultati ottenuti in ogni fascia oraria pesati per la durata della fascia stessa e infine si divide per la durata del giorno:

$$E[N_{Q_{\text{giorno-finesettimana}}}] = \frac{E[N_{Q_{7 \rightarrow 13}}] \cdot 360 + E[N_{Q_{13 \rightarrow 18}}] \cdot 300}{10 \cdot 60} + \frac{E[N_{Q_{18 \rightarrow 20}}] \cdot 120 + E[N_{Q_{20 \rightarrow 22}}] \cdot 120}{19 \cdot 60} + \frac{E[N_{Q_{22 \rightarrow 2}}] \cdot 240}{19 \cdot 60} = 0.09 \text{ job}$$

Per calcolare la media, allora, basta fare una media pesata sulla settimana:

$$E[N_Q] = \frac{5 \cdot E[N_{Q_{\text{giorno-settimana}}}] + 2 \cdot E[N_{Q_{\text{giorno-finesettimana}}}]}{7} = 0.03 \text{ job}$$

Per calcolare il tempo medio in coda, allora, si sfrutta la legge di Little e il λ_{medio} calcolato prima:

$$E[T_Q] = \frac{E[N_Q]}{\lambda_{\text{medio}}} = 0.18 \text{ min}$$

6.3 Tempo di risposta e numero medio di job nel nodo

Una volta calcolato il tempo medio in coda, basta sommare $\frac{1}{\mu}$ per ottenere il tempo medio di risposta:

$$E[T_S] = E[T_Q] + \frac{1}{\mu} = 2.18 \text{ min}$$

Usando Little, ci ricaviamo facilmente il numero di job nel centro:

$$E[N] = E[T_S] \cdot \lambda_{\text{medio}} = 0.41 \text{ job}$$

6.4 Utilizzazione

Per calcolare l'utilizzazione in un multiserver, basta fare:

$$\rho = \frac{\lambda_{\text{medio}}}{m \cdot \mu} = 0.19$$

7 Conclusioni

Come si nota dai risultati ottenuti, a parte per qualche errore di approssimazione, i risultati della simulazione tendono a quelli teorici.

I risultati dell'analisi mostrano che il numero migliore di baristi da assumere è 2: infatti questo è il numero minimo con cui si riesce a rispettare il vincolo sul tempo di risposta, anche se il guadagno sarebbe stato maggiore con $m = 1$. Continuando invece ad aumentare il numero di baristi, il guadagno diminuisce sempre di più, mentre migliorano i tempi di risposta:

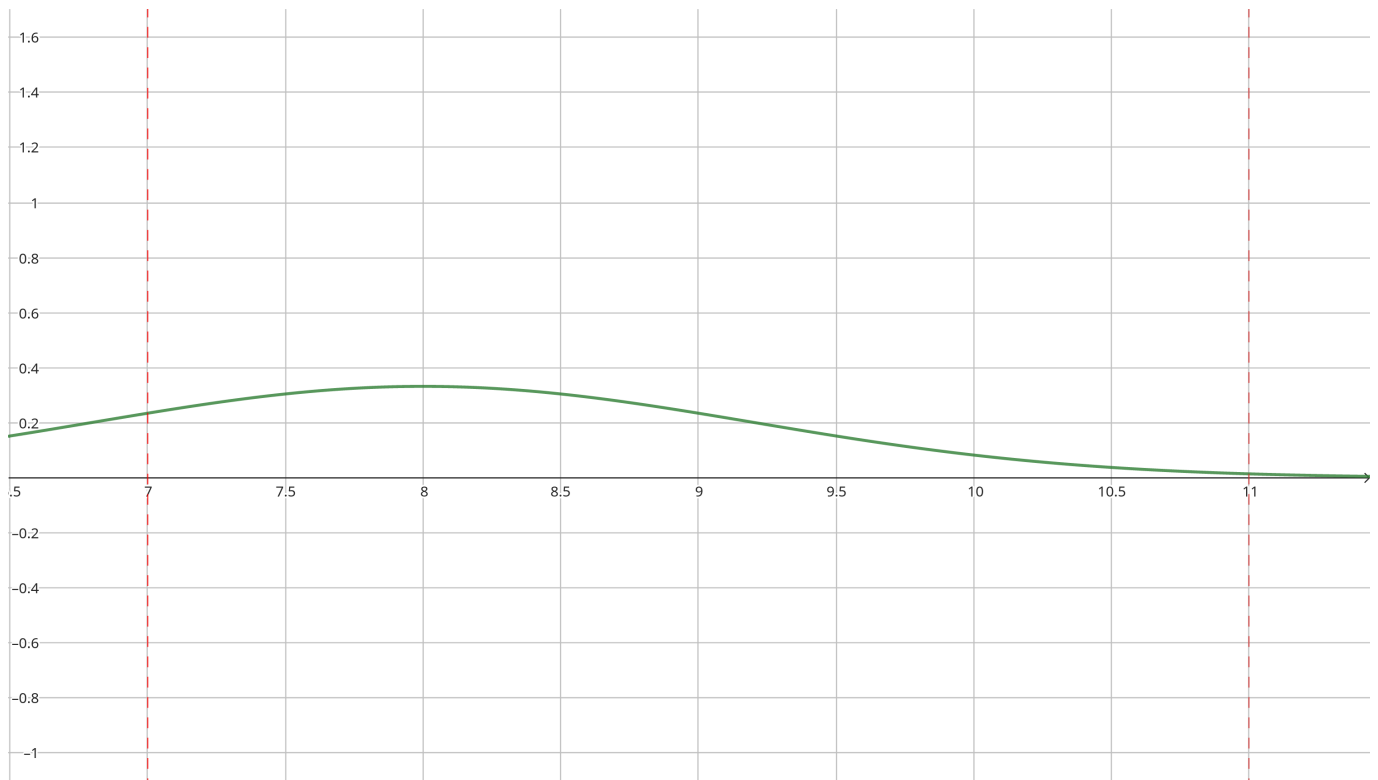
m	$E[T_S]$	$r(\tau)$
1	5.53 min	3070.10 € al mese
2	2.18 min	1853.43 € al mese
3	2.02 min	636.76 € al mese
4	2.00 min	-579.90 € al mese

Si può notare che, al crescere di m , la differenza dei tempi con il caso $m - 1$ è sempre minore: questo si può spiegare considerando che i tassi di arrivo non sono stati cambiati: aumentando m , quindi, si va a diminuire il tempo di coda di ogni job, che tende quindi a 0.

8 Immagini

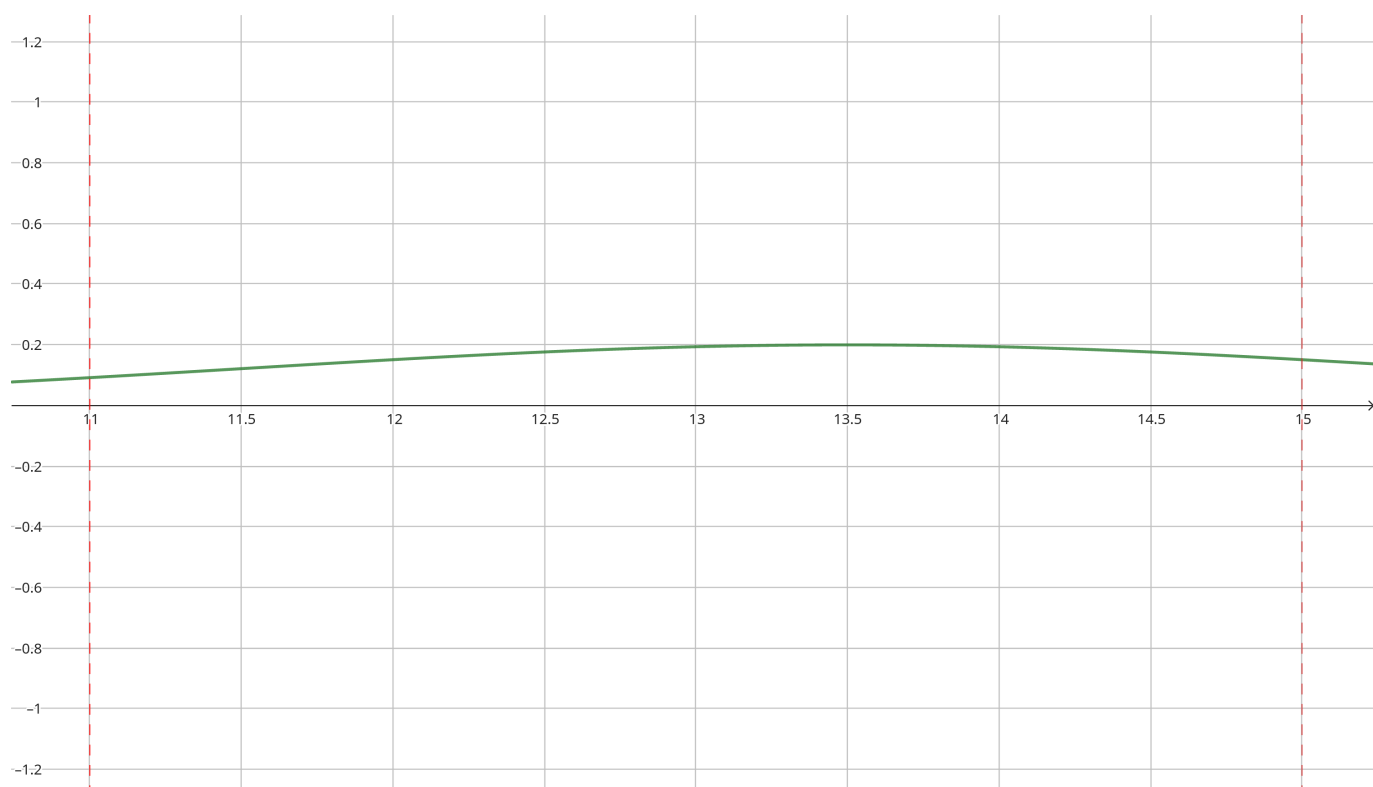
8.1 Distribuzioni gaussiane per gli arrivi

$$y = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



Fascia oraria: 07:00 → 11:00

$$\mu = 8; \sigma = 1.2$$



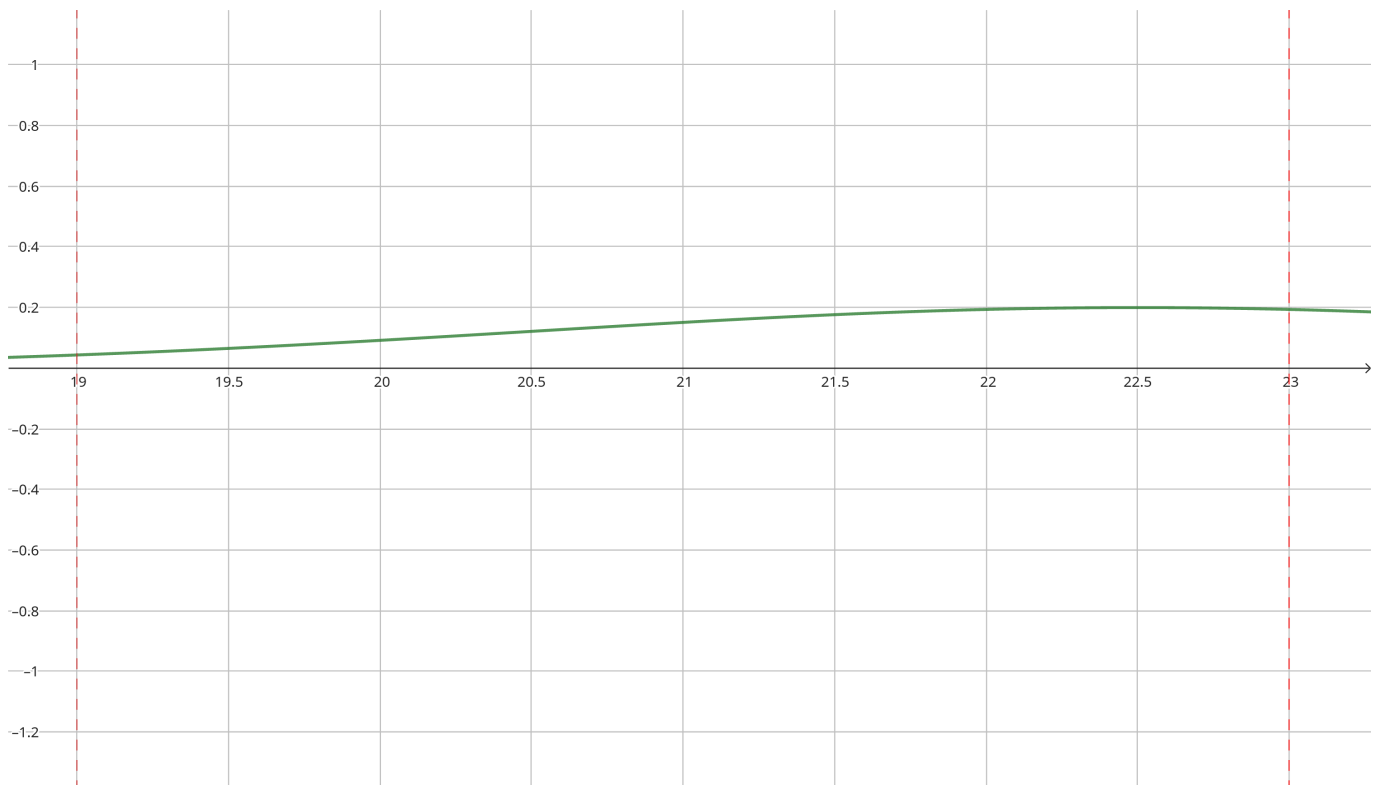
Fascia oraria: 11:00 → 15:00

$$\mu = 13.5; \sigma = 2$$



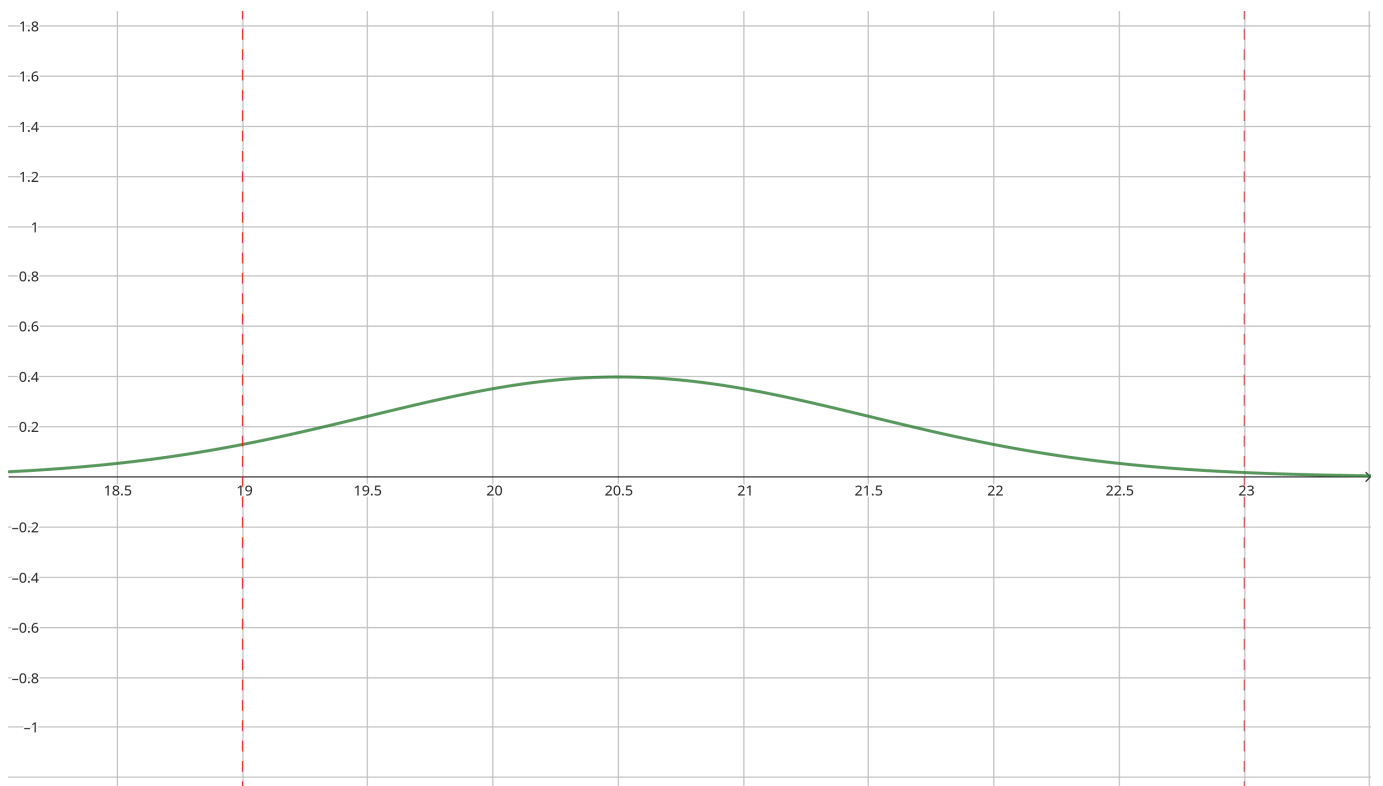
Fascia oraria: 18:00 → 19:00

$$\mu = 18.5; \sigma = 0.4$$



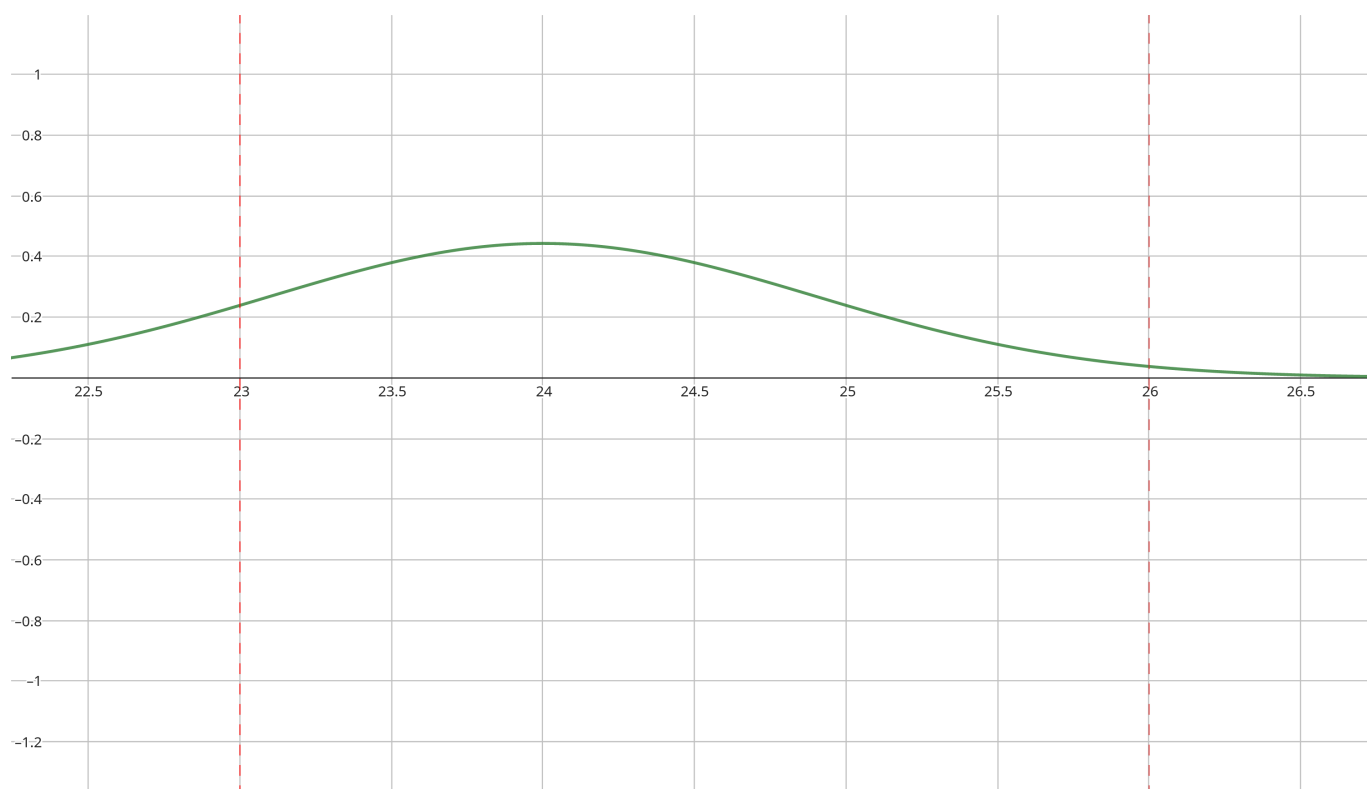
Fascia oraria: 19:00 → 23:00, bar

$$\mu = 22.5; \sigma = 2$$



Fascia oraria: 19:00 → 23:00, pizzeria

$$\mu = 20.5; \sigma = 1$$



Fascia oraria: 23:00 → 02:00

$$\mu = 24; \sigma = 0.9$$