

PMCSN Report

Marco Marcucci

marco.marcucci96@gmail.com
0286352

Giuseppe Lasco

giuseppe.lasco17@gmail.com
0286045

Valentina Valentina Francesca falaschi

valentinayffalas@gmail.com
0295947

University of Roma Tor Vergata — January 5, 2022

Introduzione

Lo studio ha lo scopo di analizzare due sistemi a code, stocastici e dinamici, che simulano lo scenario di una sala giochi. Il primo caso di studio si riferisce ad un modello base descrivendone il comportamento ed i limiti. Il secondo caso di studio propone un modello avanzato, che ha lo scopo di enfatizzare le differenze e i miglioramenti rispetto al primo caso.

L'intera analisi viene condotta procedendo attraverso i passi di modellazione, simulazione ed analisi delle statistiche di output.

1 Modello Base

Il modello base si riferisce al comportamento di una sala giochi, al cui interno sono presenti dei videogiochi (arcade). L'accesso alla struttura può avvenire in due modi:

- Attraverso l'acquisto di un biglietto online, esibendo al momento della prenotazione il proprio green-pass, saltando così i controlli;
- Attraverso l'acquisto di un biglietto in loco, sottponendosi al controllo del proprio green-pass oppure, in mancanza di quest'ultimo, al test antigenico rapido.

L'uscita dalla struttura può avvenire in tre modi:

- In seguito al riscontro della positività al test antigenico rapido;
- In seguito al riscontro di un green-pass non idoneo;
- In seguito al completamento della partita.

Nei primi due casi, il giocatore esce dal sistema senza pagare il biglietto. Inoltre, la politica aziendale della sala giochi prevede che il giocatore abbia diritto ad un rimborso del biglietto proporzionale al tempo di attesa che ha sperimentato.

Inserire figura 1 (alto livello)

1.1 Obiettivi

Controllare tutti i parametri definiti

Il primo obiettivo dello studio è quello di massimizzare i profitti della sala giochi. A tale scopo è stata definita una funzione guadagno:

$$G = \text{costo biglietto} * \#\text{giocatori} - \text{costo elettricità} * \#\text{arcade attivi} - \#\text{giocatori} * \text{rimborso biglietto}(\text{attesa media nella coda}) * \text{costo biglietto} \quad (1)$$

Il costo del biglietto è di 10 €. Per quanto riguarda l'elettricità è stata considerata una spesa giornaliera per Arcade pari a 2.4 €. La funzione *rimborso biglietto*, che calcola la percentuale del *costo biglietto* da restituire al giocatore, è proporzionale al tempo di attesa speso da quest'ultimo in coda nei nodi Arcade. La massima percentuale rimborsabile è del 80%. L'attesa minima in coda per cui è previsto il rimborso è di 8 minuti.

Il secondo obiettivo dello studio è quello di assicurare un tempo medio di risposta del sistema inferiore a 40 minuti. Infatti, da un calcolo teorico basato sulle caratteristiche del sistema, specificate in seguito, si è dedotto che il tempo medio dei servizi del sistema è di circa 19 minuti. Per cui, è stata impostata una tolleranza del tempo medio di attesa nelle code sperimentato dall'utente, pari a circa il tempo di servizio.

1.2 Modello Concettuale

Inserire figura 2 (basso livello)

La sala giochi è modellata attraverso una rete aperta composta da N nodi rappresentanti i videogiochi e da un nodo rappresentante il controllo Covid-19. Ogni nodo si compone di un servente e di una coda infinita, poiché tale configurazione è la più verosimile.

La struttura è aperta H24 e la giornata è suddivisa in 4 fasce orarie caratterizzate da tassi di arrivo differenti:

- 08:00-12:00
- 12:00-17:00
- 17:00-22:00
- 22:00-08:00

Le variabili di stato del sistema considerate sono:

- Numero di clienti in ogni nodo;
- Numero di nodi attivi;
- Tipologia di cliente.

Le code inizialmente vengono considerate vuote, conseguentemente il numero di clienti ad ogni nodo è 0. La tipologia di ogni cliente (Green-pass, tampone o biglietto online) è decisa aleatoriamente al suo arrivo. I nodi per ogni fase vengono definiti aperti o chiusi in base alla configurazione scelta durante lo studio, che può essere ottima o non ottima.

Gli eventi considerati sono:

- Arrivo di un cliente;
- Completamento di un servizio;
- Cambio di fascia oraria.

Tali eventi possono causare cambiamenti dello stato. L'arrivo di un cliente causa l'aumento della popolazione in un dato nodo e l'aumento del numero della popolazione del sistema. Il completamento di un servizio causa il passaggio di un cliente da un nodo ad un altro o la sua uscita dal sistema. Il cambio di fascia oraria può causare l'accensione o lo spegnimento degli Arcade. Se un nodo arcade viene spento quando ha ancora clienti in servizio, questo continuerà a processarli, pur non ricevendo altri arrivi, mentre i clienti in coda vengono distribuiti uniformemente tra i nodi Arcade rimasti operativi. Il tempo in cui un nodo Arcade è idle, ma acceso, viene considerato nel calcolo dei costi.

1.3 Modello delle Specifiche

Controllare tutti i parametri definiti

Gli arrivi sono modellati con un processo di Poisson, con media variabile a seconda della fascia oraria. In particolare, la media per la prima fascia oraria è di 0.07143 minuti (14 minuti/cliente), la media per la seconda fascia oraria è di 0.2 minuti (5 minuti/cliente), la media per la terza fascia oraria è di 0.07143 minuti (15 minuti/cliente), la media per la quarta fascia oraria è di 0.02857 minuti (35 minuti/cliente).

La percentuale di clienti con biglietto online è del 20%. La percentuale di clienti che si sottopone al controllo del Green-pass è del 48%. La percentuale di clienti che si sottopone al tampone è del 32%.

La distribuzione del tempo di servizio per il controllo del green-pass è stata modellata come una distribuzione Normale con media 2 minuti (0.5 passeggeri/minuto) e varianza 2.25 minuti, troncata tra 1 e 3 minuti.

La distribuzione del tempo di servizio per il tampone è stata modellata come una distribuzione Normale con media 10 minuti (0.1 passeggeri/minuto) e varianza 2.25 minuti, troncata tra 8 e 12 minuti.

È stata scelta una Normale troncata con questi valori poiché permette di modellare un'operazione con bassa variabilità e di scartare valori eccessivamente piccoli o grandi che non rispecchierebbero tempi di servizio reali.

La distribuzione del tempo di servizio per un generico arcade è stata modellata come una distribuzione Normale con media 0.06 minuti (0.1 passeggeri/minuto) e varianza 9 minuti, troncata tra 3 e 25 minuti.

La percentuale di clienti che risulta positiva al Covid-19 oppure presenta un green-pass non idoneo è in totale del 5%.

Le probabilità per la scelta del node arcade è stata modellata tramite una distribuzione Uniforme in modo che ciascun servente abbia la stessa probabilità di essere selezionato. Questa scelta permette di mantenere circa la stessa utilizzazione per tutti i serventi dato che hanno lo stesso tasso di servizio.

Infine, si considera un numero di nodi per il controllo Covid-19 pari a 1 ed un numero massimo di nodi arcade pari a 20.

1.4 Modello Computazionale

L'approccio di simulazione utilizzato è stato quello Next-Event Simulation. Il linguaggio di programmazione utilizzato è *Python v.3.8*. Il codice di base per la simulazione segue quanto descritto nell'algoritmo 1. Quest'ultimo si avvale di ulteriori strutture dati e funzioni necessarie al corretto funzionamento.

Algorithm 1 M/G/1 - Rete Aperta - Scheduling FIFO

```

1: procedure MODELLO BASE
2:   Inizializzazione clock
3:   Inizializzazione lista dei nodi
4:   Inizializzazione struttura Integrali
5:   Selezione nodo su cui generare il primo arrivo
6:   Generazione primo arrivo
7:   while ultimo arrivo < STOP or ci sono job nel sistema do
8:     Scelta del prossimo evento
9:     for  $i \leftarrow 0$ , numero nodi do
10:    if nodi non vuoti then
11:      Aggiornamento statistiche
12:      Avanzamento del clock
13:      if Evento = Arrivo then                                 $\triangleright$  Arrivo
14:        if Evento = Cambio fascia oraria then
15:          Selezione tasso arrivi corretto
16:          Aggiornamento stato nodo
17:          Aggiornamento stato sistema
18:          Schedulazione prossimo arrivo
19:          if Jobs nel centro = 1 then
20:            Schedulazione prossimo servizio
21:          else                                                  $\triangleright$  Completamento
22:            Aggiornamento stato nodo
23:            if Nodo Arcade then
24:              Aggiornamento stato sistema
25:            if Jobs nel centro > 0 then
26:              Schedulazione prossimo servizio
27:            if Nodo Covid-Check then
28:              if Covid test negativo and Green-pass valido then
29:                Aggiornamento stato sistema

```

Cambiare o no lo pseudocodice?

1.5 Verifica

Le strutture e le funzioni precedentemente accennate sono state sottoposte ad una fase di testing in modo da garantire il loro corretto funzionamento. Durante l'implementazione del progetto, l'applicazione iterativa di tale fase ha permesso di individuare e correggere diversi difetti, emersi a seguito del manifestarsi di malfunzionamenti del sistema.

1.6 Validazione

Al fine di validare il modello implementato, sono state eseguite diversi run variando determinati parametri. I risultati possono essere osservati in tabella 4 e 5.

Table 1: System statistics

# nodi arcades	avg interarrival time	avg wait	avg # node
2	14.037119	27.6247051	1.967967
5	14.037119	21.808119	1.553597
9	14.037119	20.885323	1.487860

* seed = 1234567891, $\lambda_{system} = \frac{1}{14}$, #jobs = 517028,

Per quanto riguarda la tabella 4, è possibile evincere come il tempo di interarrivo medio del sistema converga al parametro impostato λ_{system}^{-1} .

Table 2: Arcade statistics

# nodi arcades	avg interarrival time	avg wait	avg delay	avg # node	avg # queue	utilization
2	29.243586	23.024591	8.028968	0.787335	0.274554	0.512781
5	73.123549	16.937644	1.940329	0.231630	0.026538	0.205092
9	131.750016	15.999674	0.983205	0.121439	0.007463	0.113977

* seed = 1234567891, $\lambda_{system} = \frac{1}{14}$, #jobs = 517028,

Per quanto riguarda la tabella 5, si può notare come la relazione di Little sia soddisfatta e come l'utilizzazione e il tempo di interarrivo medio, rispettivamente, decresca ed aumenti al crescere del numero del numero degli arcade. Tale comportamento è giustificato dal fatto che i jobs vengono distribuiti in modo equiprobabile tra i diversi nodi arcade.

Come è possibile osservare dalle tabelle, inoltre, all'aumentare dei nodi nel sistema i tempi di attesa e il numero di jobs nei rispettivi centri diminuisce coerentemente a ciò che ci si aspetta.

1.7 Esperimenti di Simulazione

Al fine di raggiungere gli obiettivi prefissati, è utile dedurre il minimo numero di nodi che permette al sistema di raggiungere la stazionarietà e, in seguito, ricavare la configurazione ottimale del sistema.

Il raggiungimento della stazionarietà o meno viene studiato attraverso l'impiego dell'analisi a orizzonte finito, mentre quella a orizzonte infinito permette di confrontare le configurazioni del sistema a partire dai risultati ottenuti dal comportamento transiente.

1.7.1 Analisi del comportamento transiente

Attraverso l'analisi del comportamento transiente viene studiato il tempo medio di risposta del sistema, per ogni fascia oraria, utilizzando il metodo delle repliche. In particolare, vengono effettuate run di simulazione consecutive, utilizzando ogni volta come punto di partenza lo stato del sistema raggiunto dalla replica precedente. Il seed, infatti, viene settato una volta, all'inizio della prima replica. La tecnica delle repliche permette di calcolare stime puntuali e intervallari ad ogni acquisizione, ognuna di queste corrispondente ad un certo numero di richieste processate.

Una prima analisi ha portato ad osservare che il sistema non può raggiungere la stazionarietà per via dei limiti introdotti dalla coda del controllo Covid-19, essendo unica per modellazione. Infatti, per un $\lambda_{arrival} \gtrsim \frac{1}{4.1}$, l'utilizzazione di tale centro è approssimativamente pari ad 1, mentre quella dei centri Arcade risulta essere inferiore. Questo studio permette di concludere che il collo di bottiglia è rappresentato dalla coda del controllo Covid-19, per questo motivo, segue solamente lo studio dei nodi Arcade, considerando un $\lambda_{arrival} < \frac{1}{4.1}$.

Per quanto riguarda la prima fascia e la terza fascia oraria, rispettivamente (08:00-12:00) e (17:00-22:00), è possibile osservare i risultati dell'analisi in figura 1 e figura 2. Tali fasce, infatti, hanno lo stesso $\lambda_{arrival}$ da modellazione.

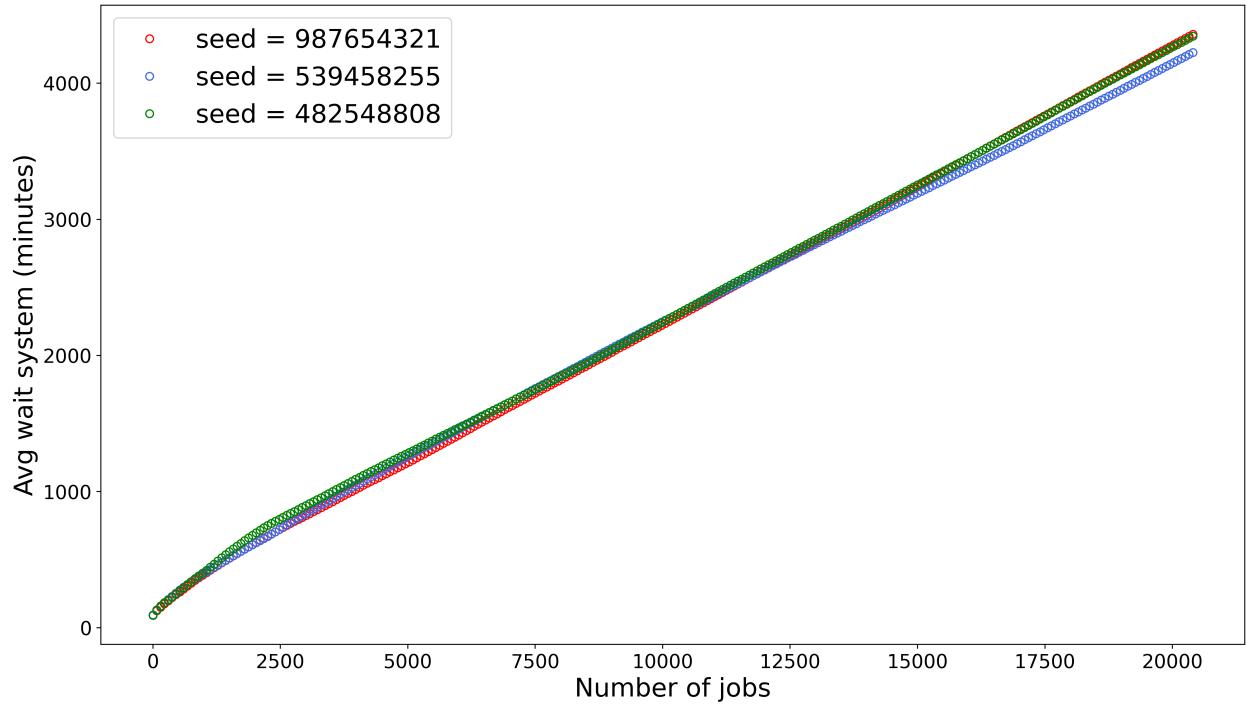


Figure 1: Average Wait System, 08:00-12:00 & 17:00-22:00, $\lambda_{arrival} = \frac{1}{14} \frac{job}{min}$, Arcades: 1, repliche=64, Sampling_freq=75

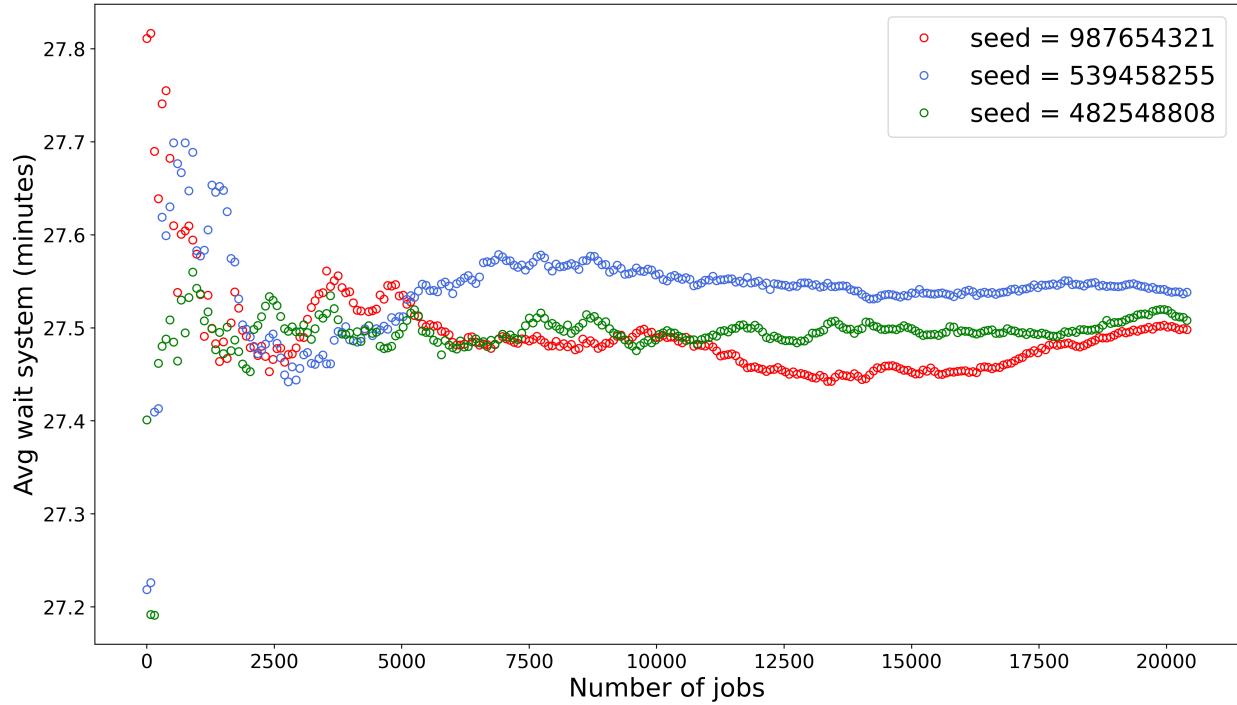


Figure 2: Average Wait System, 08:00-12:00 & 17:00-22:00, $\lambda_{arrival} = \frac{1}{14} \frac{job}{min}$, Arcades: 2, repliche=64, Sampling_freq=75

Come si evince dai grafici, il numero minimo di nodi Arcade necessari per il raggiungimento della stazionarietà, nella fascia oraria 08:00-12:00 e 17:00-22:00, è pari a 3. Infatti, nella figura 2 il tempo medio di risposta del sistema assume un comportamento stabile dopo il processamento di circa 5000 jobs. Inoltre, sono stati utilizzati 3 seed iniziali diversi, in modo tale da simulare il comportamento in condizioni

aleatorie differenti. Tali considerazioni, con le dovute differenze, valgono per le altre fasce orarie.

Per quanto riguarda la seconda fascia oraria (12:00-17:00), è possibile osservare i risultati dell'analisi in figura 3 e figura 4.

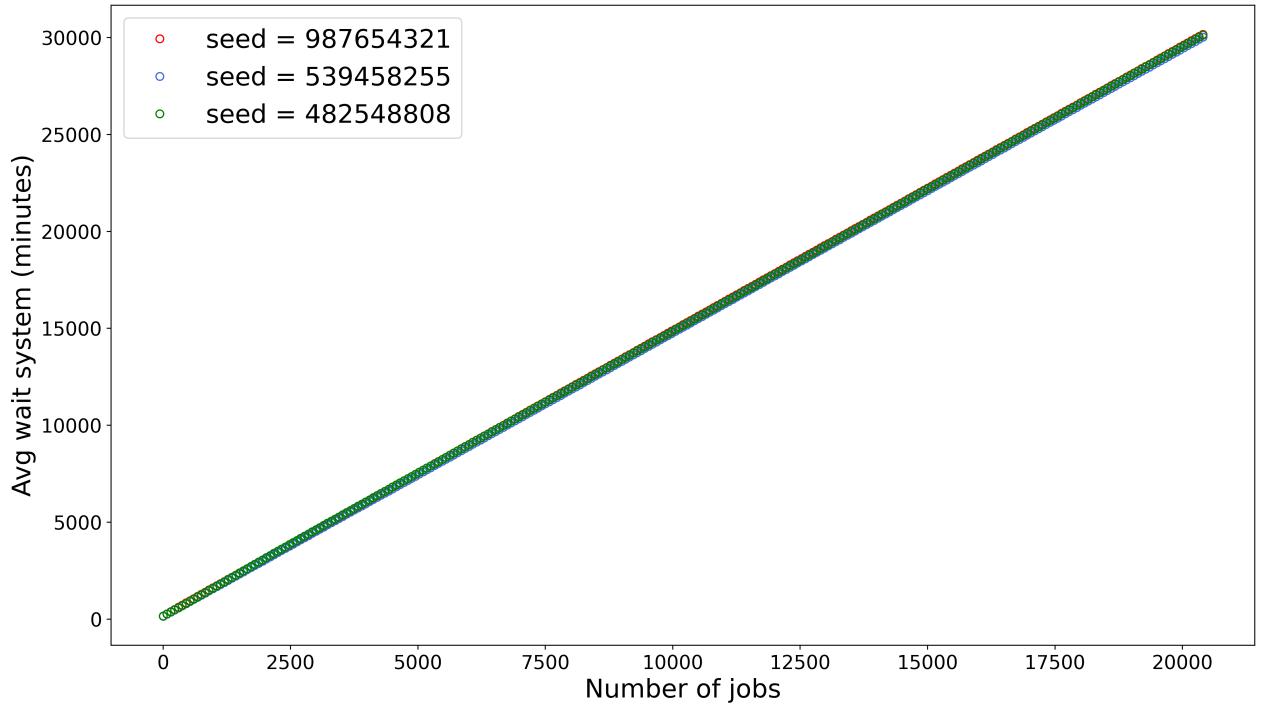


Figure 3: Average Wait System, 12:00-17:00, $\lambda_{arrival} = \frac{1}{5} \frac{job}{min}$, Arcades: 2,
repliche=64, Sampling_freq=75

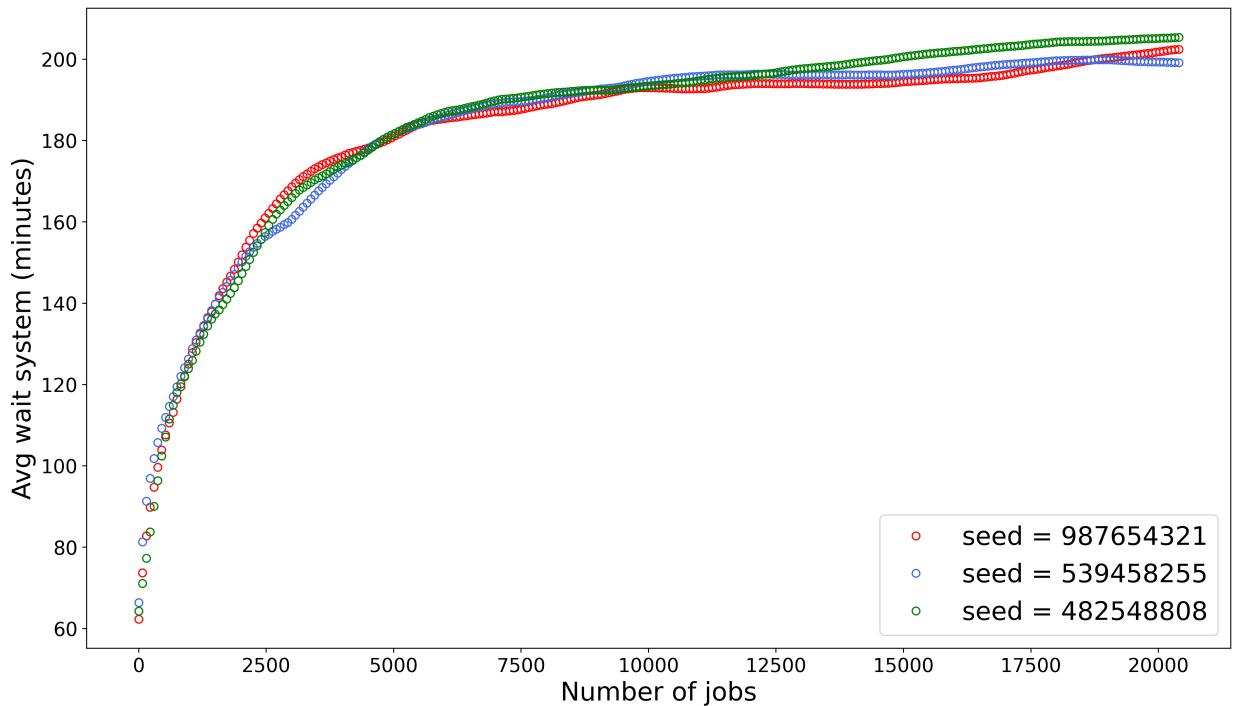


Figure 4: Average Wait System, 12:00-17:00, $\lambda_{arrival} = \frac{1}{5} \frac{job}{min}$, Arcades: 3,
repliche=64, Sampling_freq=75

Per quanto riguarda la quarta fascia oraria (22:00-08:00), è possibile osservare i risultati dell'analisi in figura 5.

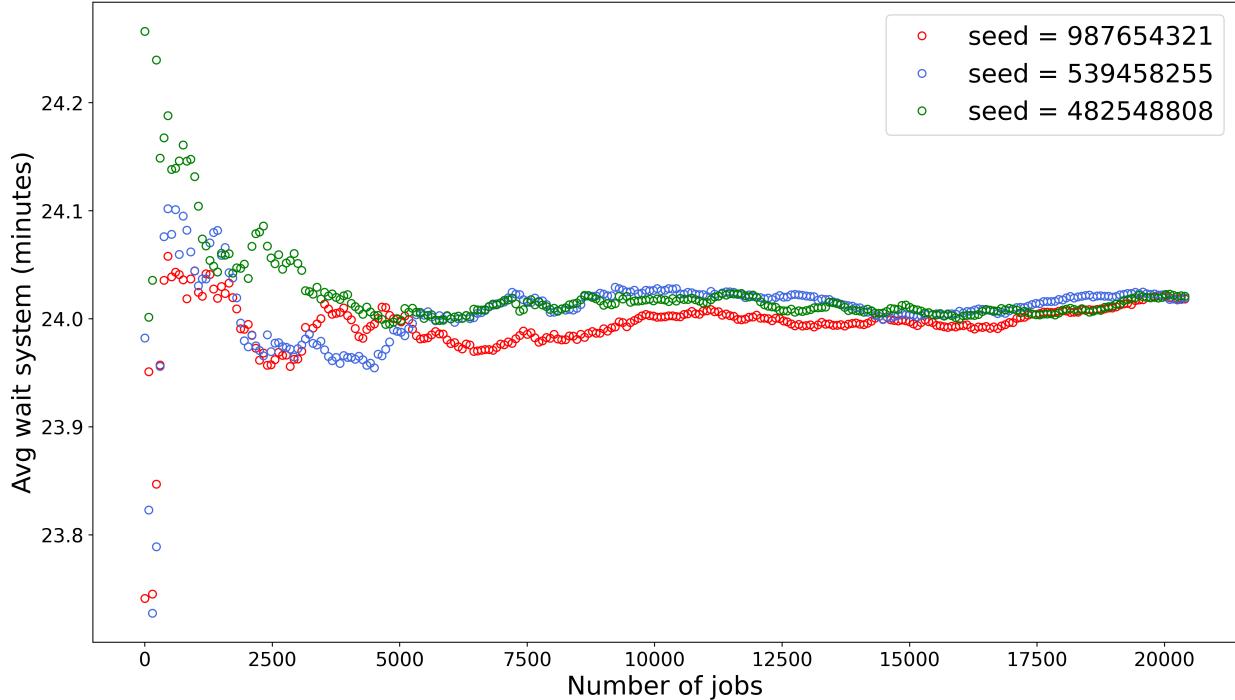


Figure 5: Average Wait System, 22:00-08:00, $\lambda_{arrival} = \frac{1}{35} \frac{job}{min}$, Arcades: 1, repliche=64, Sampling_freq=75

In questo caso, il tempo di risposta medio del sistema si stabilizza anche con l'utilizzo di un solo nodo Arcade.

1.7.2 Analisi del comportamento stazionario

Considerati i risultati dell'analisi transiente, è possibile procedere allo studio del comportamento stazionario. Attraverso l'analisi del comportamento stazionario si vuole trovare la configurazione ottima di nodi arcade, per ogni fascia oraria, che permetta di raggiungere gli obiettivi prefissati in 1.1, partendo dai risultati ottenuti dallo studio precedente. In questa fase sono state effettuate simulazioni con tempi d'osservazione molto lunghi, utilizzando la tecnica delle batch-means per ottenere delle stime degli indici di prestazione in regime stazionario, evitando la problematica del bias dovuto alle condizioni iniziali del sistema. Cruciale in questa fase la scelta di due parametri: b e k , rispettivamente la grandezza del batch e il numero di batch in cui si divide la simulazione. Per la scelta di b è stata utilizzata la linea guida di Banks, Carson, Nelson, e Nicol (2001), la quale afferma che il batch size venga aumentato fintantoché l'autocorrelazione a lag 1 tra batch means sia minore di 0.2. Nel caso in esame è stato constatato che tale direttiva viene rispettata con un b pari a 256.

Controllare ultima frase

Per la scelta del parametro k è stato usato il valore pari a 64 come consigliato dalle linee guida.

In seguito per ogni fascia oraria viene riportato il grafico relativo alla funzione guadagno e del tempo medio di risposta del sistema al variare del numero di nodi Arcade, mediata su i seed. L'asse y "income" riportato nelle figure non rappresenta l'effettivo guadagno della sala giochi in una specifica fascia oraria, che verrà analizzato in uno studio successivo, ma esprime l'effettivo andamento del guadagno, utile ad individuare la miglior configurazione da utilizzare.

Per quanto riguarda la prima fascia e la terza fascia oraria, rispettivamente (08:00-12:00) e (17:00-22:00), è possibile osservare i risultati dell'analisi in figura figura 6. Tali fasce, infatti, hanno lo stesso $\lambda_{arrival}$ da modellazione.

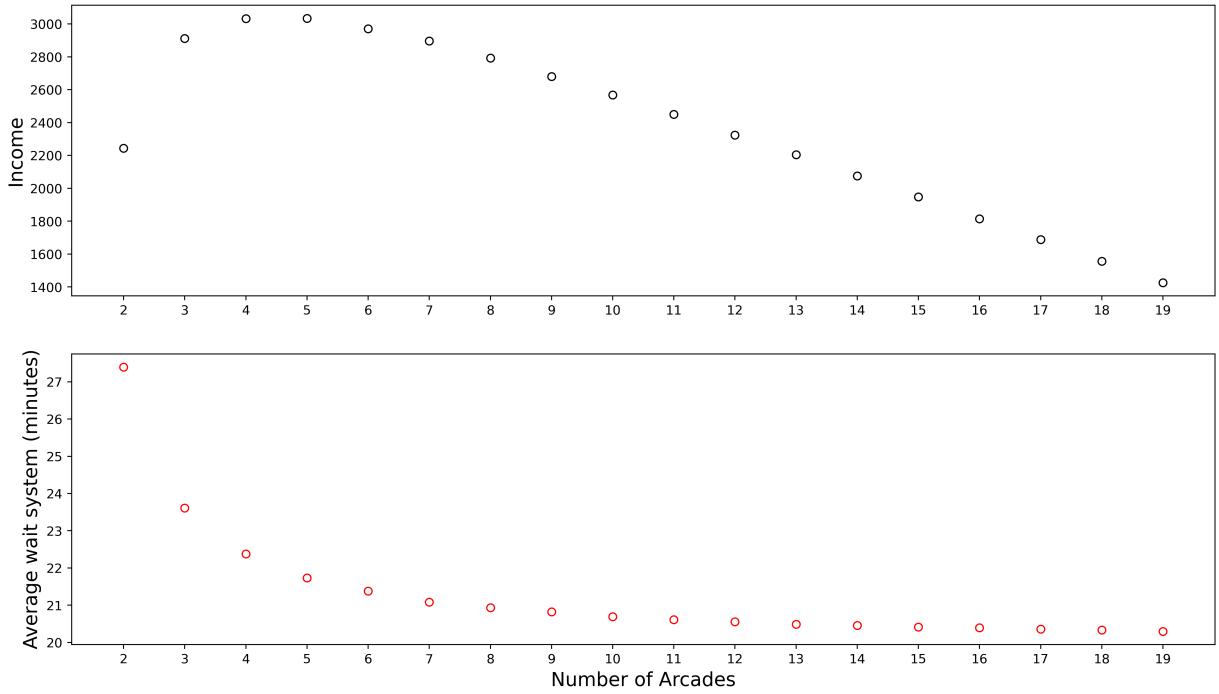


Figure 6: Income, 08:00-12:00 & 17:00-22:00, $\lambda_{arrival} = \frac{1}{14} \frac{job}{min}$, b=256, k=64

Usare 1 seed o fare la media??

Come si evince dal grafico, il numero ottimale di nodi Arcade per la prima e la terza fascia oraria, che permette di raggiungere entrambi gli obiettivi prefissati, è pari a 5.

Per quanto riguarda la seconda fascia oraria (12:00-17:00), è possibile osservare i risultati dell'analisi in figura 7.

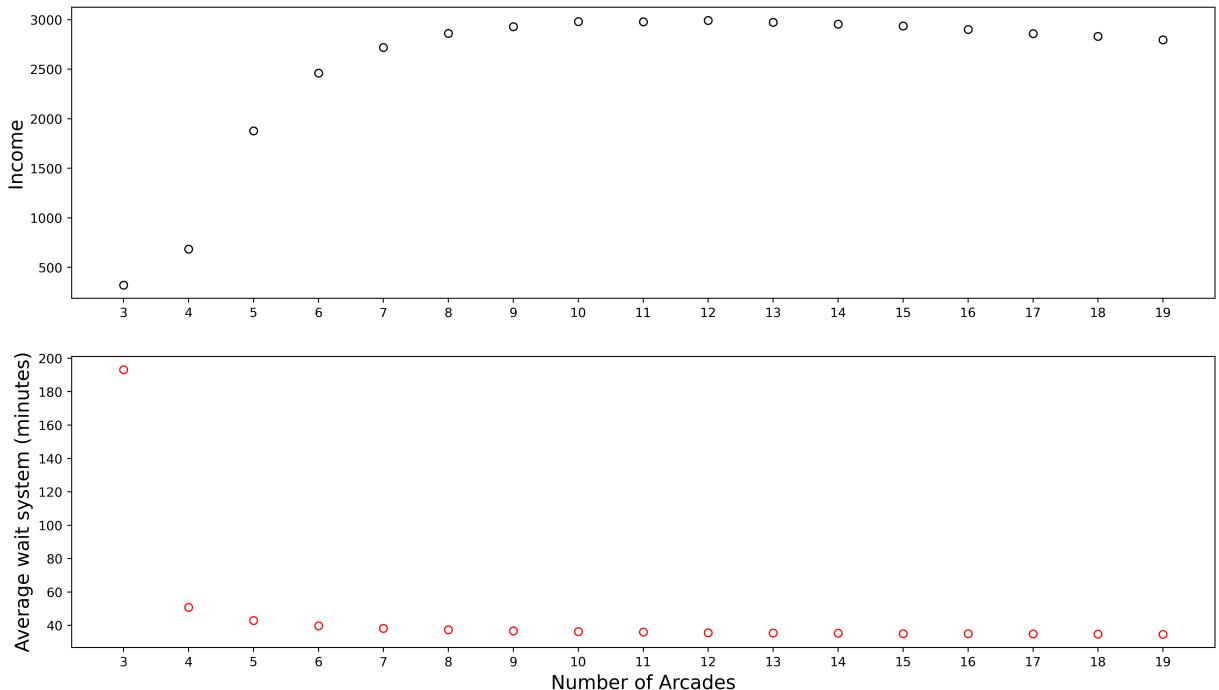


Figure 7: Income, 12:00-17:00, $\lambda_{arrival} = \frac{1}{5} \frac{job}{min}$, b=256, k=64

Il numero ottimale di server Arcade, per la seconda fascia oraria, è pari a 12.

Per quanto riguarda la quarta fascia oraria (22:00-08:00), è possibile osservare i risultati dell'analisi in figura 8.

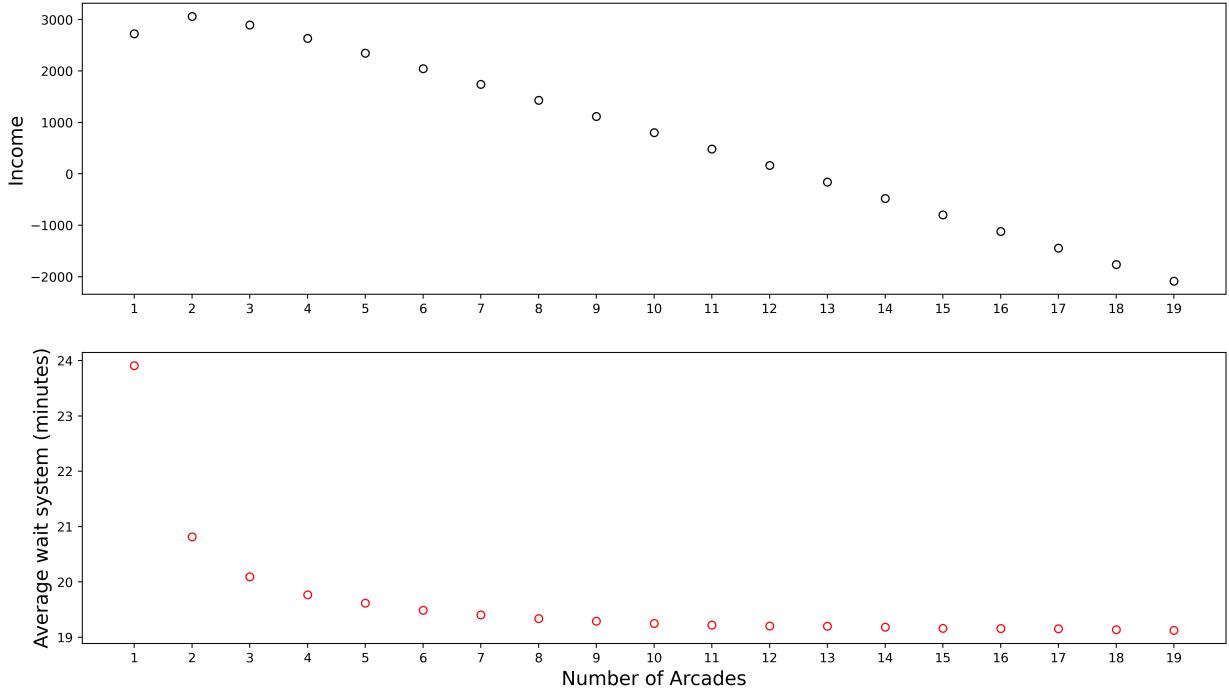


Figure 8: Income, 22:00-08:00, $\lambda_{arrival} = \frac{1}{35} \frac{job}{min}$, b=256, k=64

In questo caso, il numero ottimale di server Arcade da utilizzare è pari a 2.

In tabella 3 è possibile individuare la configurazione ottimale dei nodi Arcade, per ogni fascia oraria.

Table 3: Configurazione ottimale

Fascia oraria	Configurazione ottimale *
08:00-12:00	5
12:00-17:00	12
17:00-22:00	5
22:00-08:00	2

* Numero di nodi Arcade nel sistema.

Sono stati effettuati run con seed diversi con la configurazione ottimale, precedentemente ricavata, per studiare il tempo di risposta medio, per ogni fascia oraria, con un intervallo di confidenza pari al 95%.

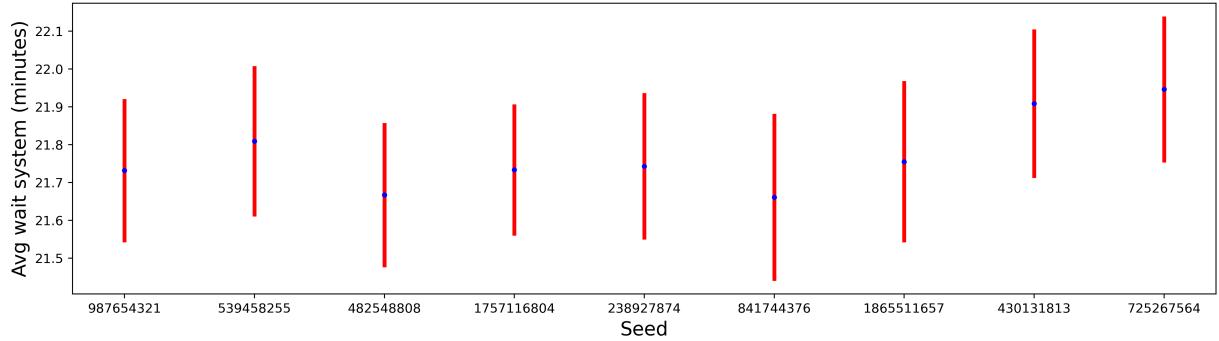


Figure 9: Average Wait System, 08:00-12:00 & 17:00-22:00, $\lambda_{arrival} = \frac{1}{14} \frac{job}{min}$, b=256, k=64

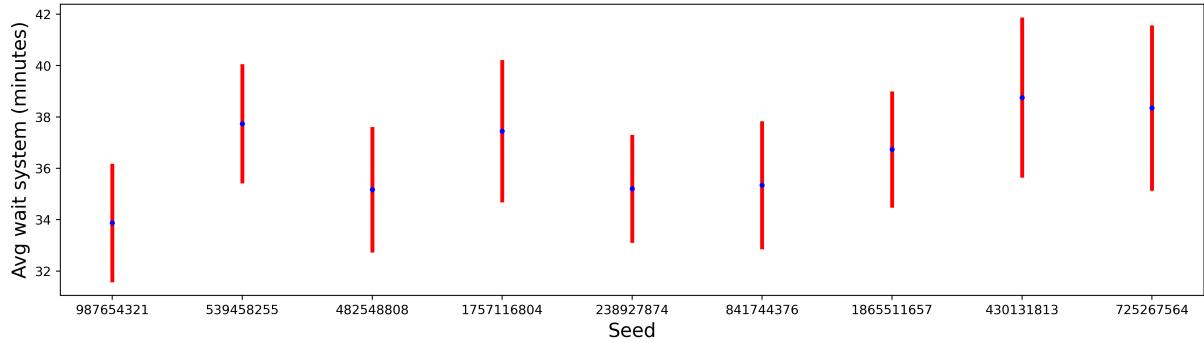
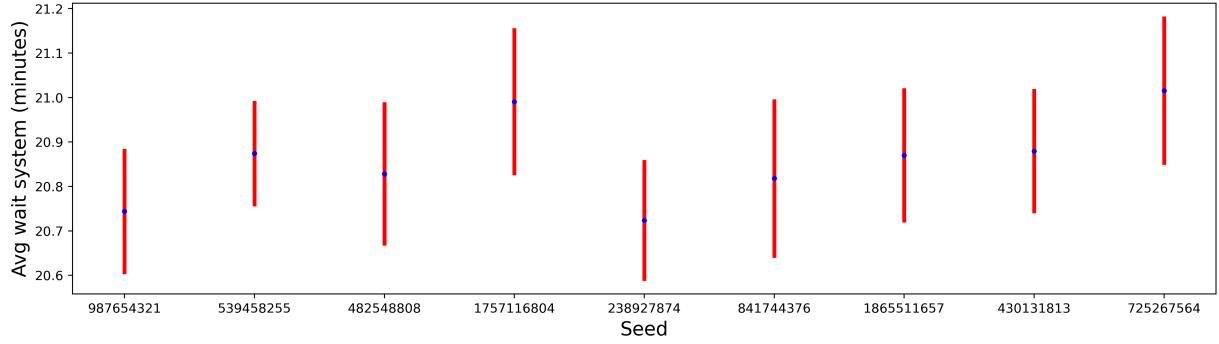


Figure 10: Average Wait System, 12:00-17:00, $\lambda_{arrival} = \frac{1}{5} \frac{job}{min}$, b=256, k=64



SFED	MEAN VALUE	STD	CONFIDENCE INTERVAL	CONFIDENCE LEVEL
987654321	20.74372572771343	0.5405007892785019	± 0.14080446025846807	95%
539458255	20.874077863916316	0.4556085657658852	± 0.1186894070542143	95%
482548808	20.82819102551389	0.6191799666948291	± 0.161300968921248	95%
1757116804	20.990702794608307	0.6338106065720944	± 0.16511235900988094	95%
238927874	20.72363775624641	0.5215049595598914	± 0.13585590587381846	95%
841744376	20.817911117315198	0.6838168478927295	± 0.17813935537764883	95%
1865511657	20.870077510579396	0.5796530520031625	± 0.1510039147832292	95%
482548808	20.82819102551389	0.6191799666948291	± 0.161300968921248	95%
430131813	20.87957925011826	0.5366561033916704	± 0.13980289109909408	95%
725267564	21.015500744854428	0.6405610735009728	± 0.16687090566007576	95%

Figure 11: Average Wait System, 22:00-08:00, $\lambda_{arrival} = \frac{1}{35} \frac{job}{min}$, b=256, k=64

1.7.3 Analisi guadagno giornaliero

Attraverso questa analisi viene effettuata una simulazione concreta di una tipica giornata lavorativa, con tre configurazioni differenti:

- Configurazione minima;
- Configurazione ottimale;
- Configurazione massima.

È possibile osservare il risultato di tale analisi in figura 12, 13 e 14.

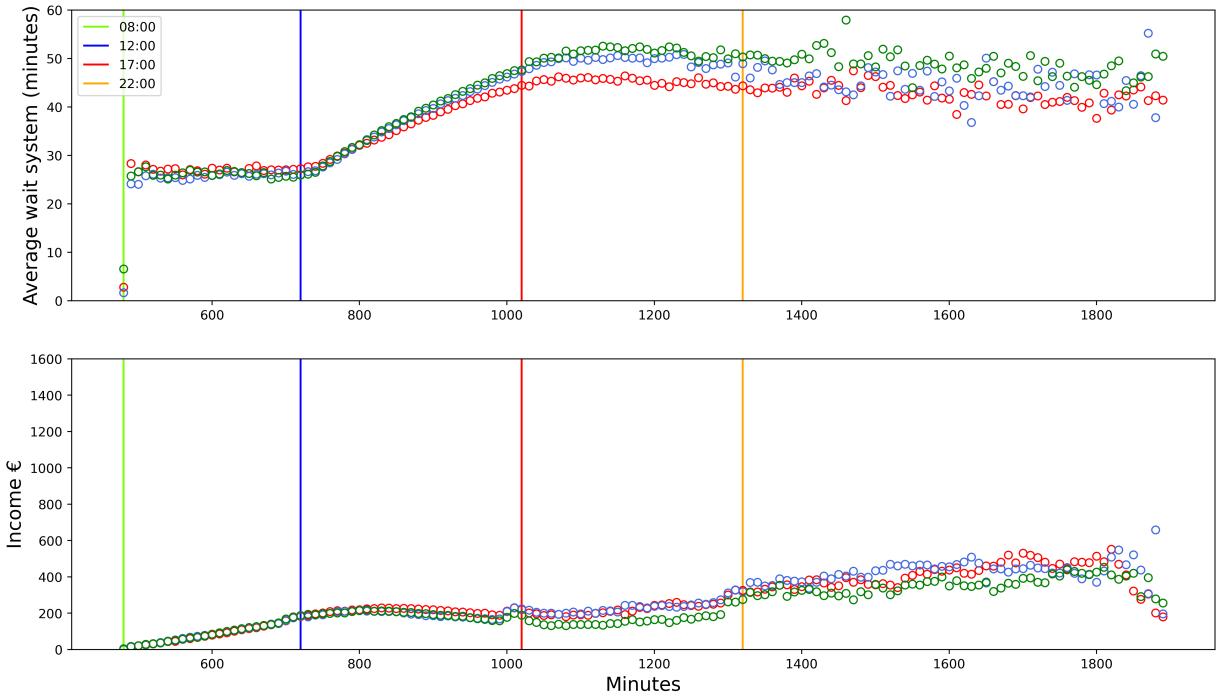


Figure 12: Average Wait System & Income Daily, minimal configuration,
sampling frequency=10 min

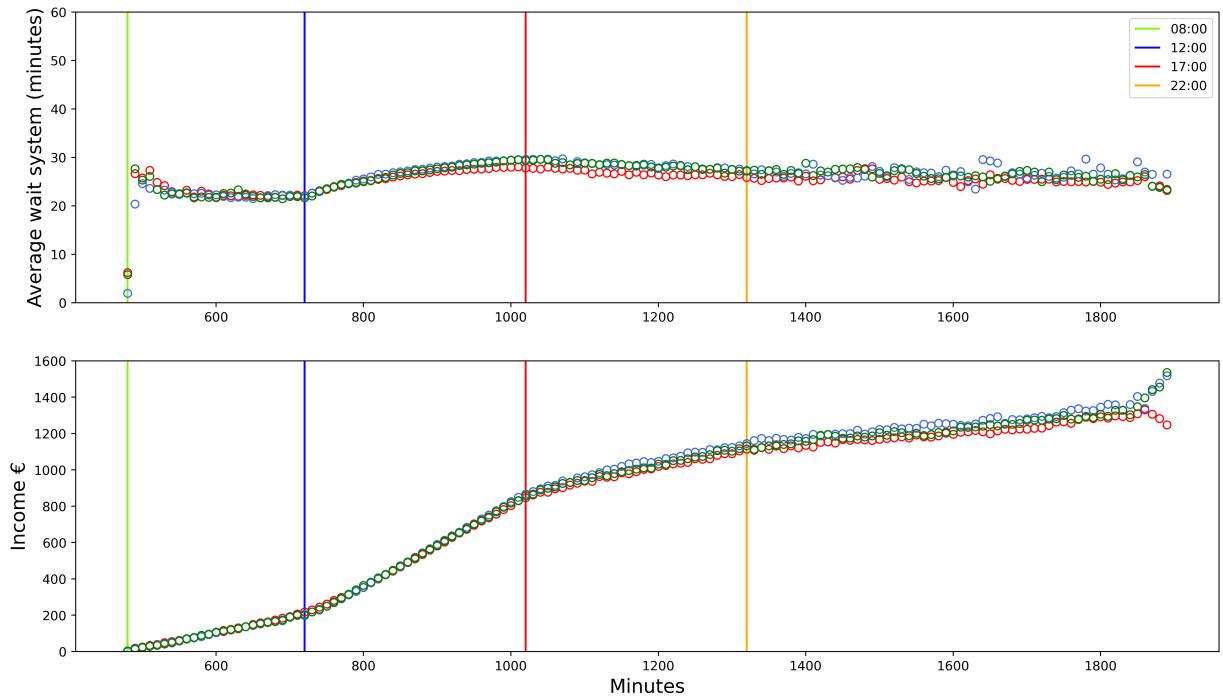


Figure 13: Average Wait System & Income Daily, optimal configuration,
sampling frequency=10 min

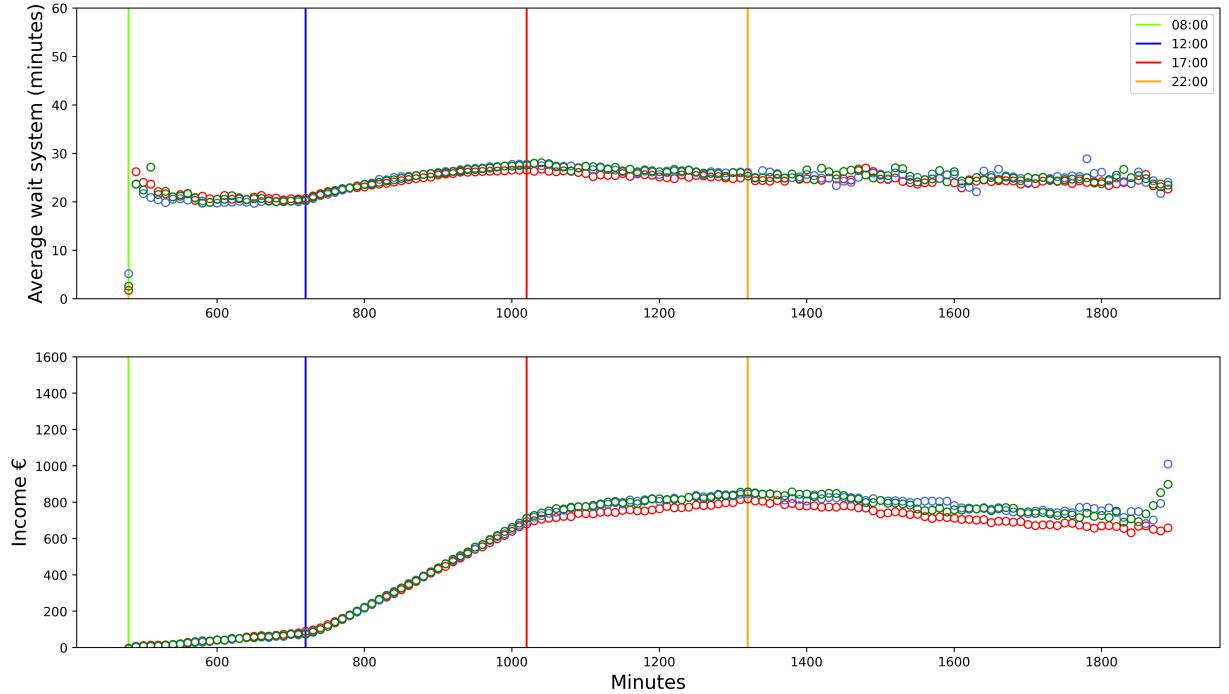


Figure 14: Average Wait System & Income Daily, maximum configuration, sampling frequency=10 min

Come si evince da tali risultati, oltre a rispettare i QoS imposti, la configurazione ottimale porta ad un guadagno giornaliero migliore. Inoltre, si può osservare come il tempo di risposta medio del sistema della configurazione massimale è comparabile con quello della configurazione ottimale, nonostante quest'ultimo abbia un numero minore di nodi Arcade.

2 Modello Avanzato

Si procede alla realizzazione di un modello avanzato che sia in grado di adattarsi ai nuovi obiettivi, descritti in seguito, a costo zero. Per tale scopo, è stata introdotta nella sala giochi, la possibilità, da parte di un utente, di acquistare un biglietto *premium* che gli permette di passare avanti agli utenti che sono in possesso di un biglietto *standard*.

Inserire figura 2 (alto livello)

2.1 Obiettivi

Controllare tutti i parametri definiti

Il primo obiettivo dello studio avanzato è quello di massimizzare i profitti della sala giochi. A tale scopo è stata definita una funzione guadagno:

$$G = \text{costo biglietto standard} * \# \text{giocatori standard} + \text{costo biglietto premium} * \# \text{giocatori premium} - \text{costo elettricità} * \# \text{arcade attivi}$$

$$\begin{aligned} G = & \text{costo biglietto standard} * \# \text{giocatori standard} + \text{costo biglietto premium} * \# \text{giocatori premium} \\ & - \text{costo elettricità} * \# \text{arcade attivi} - \# \text{giocatori standard} * \text{rimborso biglietto(attesa media nella coda standard)} * \\ & - \# \text{giocatori premium} * \text{rimborso biglietto(attesa media nella coda premium)} \end{aligned} \quad (2)$$

Il costo del biglietto standard è di 10 €, mentre il costo del biglietto premium è di 20 €. Per quanto riguarda l'elettricità è stata considerata una spesa giornaliera per Arcade pari a 2.4 €. La funzione *rimborso biglietto*, che calcola la percentuale del *costo biglietto* da restituire al giocatore, è proporzionale al tempo di attesa speso da quest'ultimo in coda nei nodi Arcade. La massima percentuale rimborsabile è del 80%. L'attesa minima in coda per cui è previsto il rimborso è di 8 minuti.

Il secondo obiettivo dello studio è quello di assicurare un tempo medio di attesa nel nodo Covid-19, inferiore a quello del modello base, per gli utenti che possiedono il green-pass.

2.2 Modello Concettuale

Inserire figura 2 (basso livello)

Nel modello avanzato la sala giochi è modellata attraverso una rete aperta composta da N nodi rappresentanti i videogiocchi (Arcade) e da un nodo rappresentante il controllo Covid-19. Ogni nodo Arcade si compone di un servente e di due code infinite con priorità astratta senza prelazione. L'accesso alla coda con priorità più alta è riservato agli utenti che acquistano il biglietto premium. Il nodo Covid-19 si compone di un servente e di due code infinite con priorità size-based senza prelazione. L'accesso alla prima coda è riservato agli utenti che possiedono il green-pass, dato che questi ultimi hanno un tempo di servizio inferiori rispetto a chi esegue il test antigenico rapido.

Non sono state scelte code di priorità con prelazione poiché, nonostante esse avrebbero avuto prestazioni migliori per gli obiettivi prefissati, non rispecchiano uno scenario reale.

La struttura è aperta H24 e la giornata è suddivisa in 4 fasce orarie caratterizzate da tassi di arrivo differenti:

- 08:00-12:00
- 12:00-17:00
- 17:00-22:00
- 22:00-08:00

Le variabili di stato del sistema considerate sono:

- Numero di clienti standard in ogni nodo;
- Numero di clienti premium in ogni nodo;
- Numero di nodi attivi;
- Tipologia di cliente.

Le code inizialmente vengono considerate vuote, conseguentemente il numero di clienti ad ogni nodo è 0. La tipologia di ogni cliente (Green-pass, tampone o biglietto online) è decisa aleatoriamente al suo arrivo. I nodi per ogni fase vengono definiti aperti o chiusi in base alla configurazione scelta durante lo studio, che può essere ottima o non ottima.

Gli eventi considerati sono:

- Arrivo di un cliente standard;
- Arrivo di un cliente premium;
- Completamento di un servizio;
- Cambio di fascia oraria.

Tali eventi possono causare cambiamenti dello stato. L'arrivo di un cliente causa l'aumento della popolazione in un dato nodo e l'aumento del numero della popolazione del sistema. Il completamento di un servizio causa il passaggio di un cliente da un nodo ad un altro o la sua uscita dal sistema. Il cambio di fascia oraria può causare l'accensione o lo spegnimento degli Arcade. Se un nodo arcade viene spento quando ha ancora clienti in servizio, questo continuerà a processarli, pur non ricevendo altri arrivi, mentre i clienti in coda vengono distribuiti uniformemente tra i nodi Arcade rimasti operativi. Il tempo in cui un nodo Arcade è idle, ma acceso, viene considerato nel calcolo dei costi.

2.3 Modello delle Specifiche

Controllare tutti i parametri definiti

Gli arrivi sono modellati con un processo di Poisson, con media variabile a seconda della fascia oraria. In particolare, la media per la prima fascia oraria è di 0.07143 minuti (14 minuti/cliente), la media per la seconda fascia oraria è di 0.2 minuti (5 minuti/cliente), la media per la terza fascia oraria è di 0.07143 minuti (15 minuti/cliente), la media per la quarta fascia oraria è di 0.02857 minuti (35 minuti/cliente).

La percentuale di clienti con biglietto online è del 20%. La percentuale di clienti che si sottopone al controllo del Green-pass è del 48%. La percentuale di clienti che si sottopone al tampone è del 32%.

La distribuzione del tempo di servizio per il controllo del green-pass è stata modellata come una distribuzione Normale con media 2 minuti (0.5 passeggeri/minuto) e varianza 2.25 minuti, troncata tra 1 e 3 minuti.

La distribuzione del tempo di servizio per il tampone è stata modellata come una distribuzione Normale con media 10 minuti (0.1 passeggeri/minuto) e varianza 2.25 minuti, troncata tra 8 e 12 minuti.

È stata scelta una Normale troncata con questi valori poiché permette di modellare un'operazione con bassa variabilità e di scartare valori eccessivamente piccoli o grandi che non rispecchierebbero tempi di servizio reali.

La distribuzione del tempo di servizio per un generico arcade è stata modellata come una distribuzione Normale con media 0.06 minuti (0.1 passeggeri/minuto) e varianza 9 minuti, troncata tra 3 e 25 minuti.

La percentuale di clienti che acquista un biglietto premium è pari al 36%.

La percentuale di clienti che risulta positiva al Covid-19 oppure presenta un green-pass non idoneo è in totale del 5%.

Le probabilità per la scelta del node arcade è stata modellata tramite una distribuzione Uniforme in modo che ciascun servente abbia la stessa probabilità di essere selezionato. Questa scelta permette di mantenere circa la stessa utilizzazione per tutti i serventi dato che hanno lo stesso tasso di servizio.

Infine, si considera un numero di nodi per il controllo Covid-19 pari a 1 ed un numero massimo di nodi arcade pari a 20.

2.4 Modello Computazionale

L'approccio di simulazione utilizzato è stato quello Next-Event Simulation. Il linguaggio di programmazione utilizzato è *Python v.3.8*. Il codice di base per la simulazione segue quanto descritto nell'algoritmo 2. Quest'ultimo si avvale di ulteriori strutture dati e funzioni necessarie al corretto funzionamento.

Algorithm 2 M/G/1 Multiqueue Priority - Rete Aperta - Scheduling FIFO

```

1: procedure MODELLO BASE
2:   Inizializzazione clock
3:   Inizializzazione lista dei nodi
4:   Inizializzazione struttura Integrali
5:   Selezione nodo su cui generare il primo arrivo
6:   Generazione primo arrivo
7:   while ultimo arrivo < STOP or ci sono job nel sistema do
8:     Scelta del prossimo evento
9:     for  $i \leftarrow 0$ , numero nodi do
10:    if nodi non vuoti then
11:      Aggiornamento statistiche
12:      Avanzamento del clock
13:      if Evento = Arrivo then                                 $\triangleright$  Arrivo
14:        if Evento = Cambio fascia oraria then
15:          Selezione tasso arrivi corretto
16:          Aggiornamento stato nodo e code di priorità            $\triangleright$  Aggiornamento code di priorità
17:          Aggiornamento stato sistema
18:          Schedulazione prossimo arrivo
19:          if Jobs nel centro = 1 then
20:            Schedulazione prossimo servizio
21:          else                                                  $\triangleright$  Completamento
22:            Aggiornamento stato nodo e code di priorità            $\triangleright$  Aggiornamento code di priorità
23:            if Nodo Arcade then
24:              Aggiornamento stato sistema
25:            if Jobs nel centro > 0 then
26:              Schedulazione prossimo servizio
27:            if Nodo Covid-Check then
28:              if Covid test negativo and Green-pass valido then
29:                Aggiornamento stato sistema

```

2.5 Verifica

Le strutture e le funzioni precedentemente accennate sono state sottoposte ad una fase di testing in modo da garantire il loro corretto funzionamento. Durante l'implementazione del progetto, l'applicazione iterativa di tale fase ha permesso di individuare e correggere diversi difetti, emersi a seguito del manifestarsi di malfunzionamenti del sistema.

2.6 Validazione

Al fine di validare il modello implementato, sono state eseguite diversi run variando determinati parametri. I risultati possono essere osservati in tabella 4 e 5.

Table 4: System statistics

# nodi arcades	avg interarrival time	avg wait	avg # node
2	14.037119	27.424052	1.953673
5	14.037119	21.654822	1.542677
9	14.037119	20.727016	1.476581

* seed = 1234567891, $\lambda_{system} = \frac{1}{14}$, #jobs = 517028,

Per quanto riguarda la tabella 4, è possibile evincere come il tempo di interarrivo medio del sistema converga al parametro impostato λ_{system}^{-1} .

Table 5: Arcades statistics

# nodi arcades	avg interarrival		avg wait		avg # node		avg # queue	
	Priority	N Priority	Priority	N Priority	Priority	N Priority	Priority	N Priority
2	81.467323	45.547404	17.803708	22.031882	0.218535	0.483710	0.034664	0.154467
5	205.391722	113.252959	15.827623	16.397222	0.077060	0.144781	0.004122	0.012311
9	370.111505	203.980331	15.425839	15.681829	0.041678	0.076871	0.001129	0.003296

* seed = 1234567891, $\lambda_{system} = \frac{1}{14}$, #jobs = 517028.

Per quanto riguarda la tabella 5, si può notare come la relazione di Little sia soddisfatta e come l'utilizzazione e il tempo di interarrivo medio, rispettivamente, decresca ed aumenti al crescere del numero del numero degli arcade. Tale comportamento è giustificato dal fatto che i jobs vengono distribuiti in modo equiprobabile tra i diversi nodi arcade.

Come è possibile osservare dalle tabelle, inoltre, all'aumentare dei nodi nel sistema i tempi di attesa e il numero di jobs nei rispettivi centri diminuisce coerentemente a ciò che ci si aspetta.

Inoltre si può constatare che tutte le statistiche risultano inferiori nel caso prioritario, in conformità con quanto ci si figuri.

Se confronta con la tabella sopra so cazzo. Che fare?

2.7 Esperimenti di Simulazione

Al fine di raggiungere gli obiettivi prefissati, è utile effettuare degli esperimenti di simulazione, utilizzando la configurazione ottimale dello studio precedente.

2.7.1 Analisi del comportamento transiente

Attraverso l'analisi del comportamento transiente viene studiato il tempo medio di risposta del sistema, per ogni fascia oraria, utilizzando il metodo delle repliche. In particolare, vengono effettuate run di simulazione consecutive, utilizzando ogni volta come punto di partenza lo stato del sistema raggiunto dalla replica precedente. Il seed, infatti, viene settato una volta, all'inizio della prima replica. La tecnica delle repliche permette di calcolare stime puntuali e intervallari ad ogni acquisizione, ognuna di queste corrispondente ad un certo numero di richieste processate.

Per quanto riguarda la prima fascia e la terza fascia oraria, rispettivamente (08:00-12:00) e (17:00-22:00), è possibile osservare i risultati dell'analisi in figura 15. Tali fasce, infatti, hanno lo stesso $\lambda_{arrival}$ da modellazione.

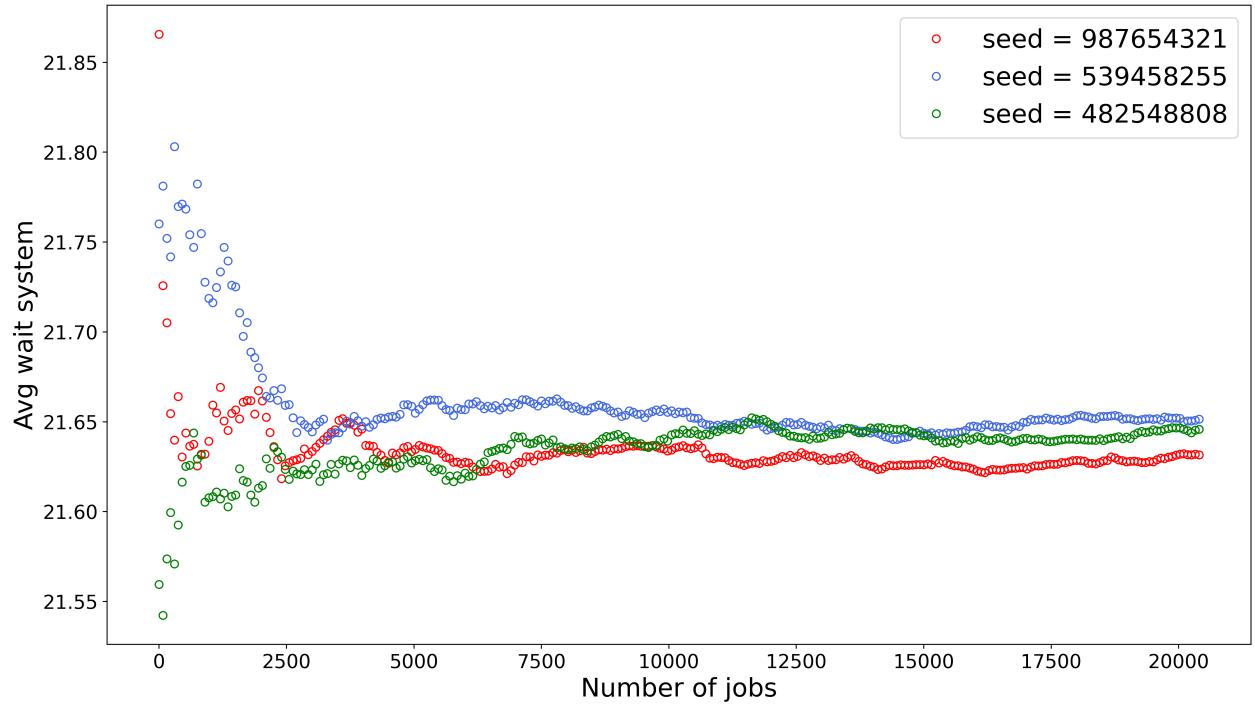


Figure 15: Average Wait System, 08:00-12:00 & 17:00-22:00,
 $\lambda_{arrival} = \frac{1}{14} \text{ min}$, Arcades: 5, repliche=64, Sampling_freq=75

Per quanto riguarda la seconda fascia oraria (12:00-17:00), è possibile osservare i risultati dell'analisi in figura ??.

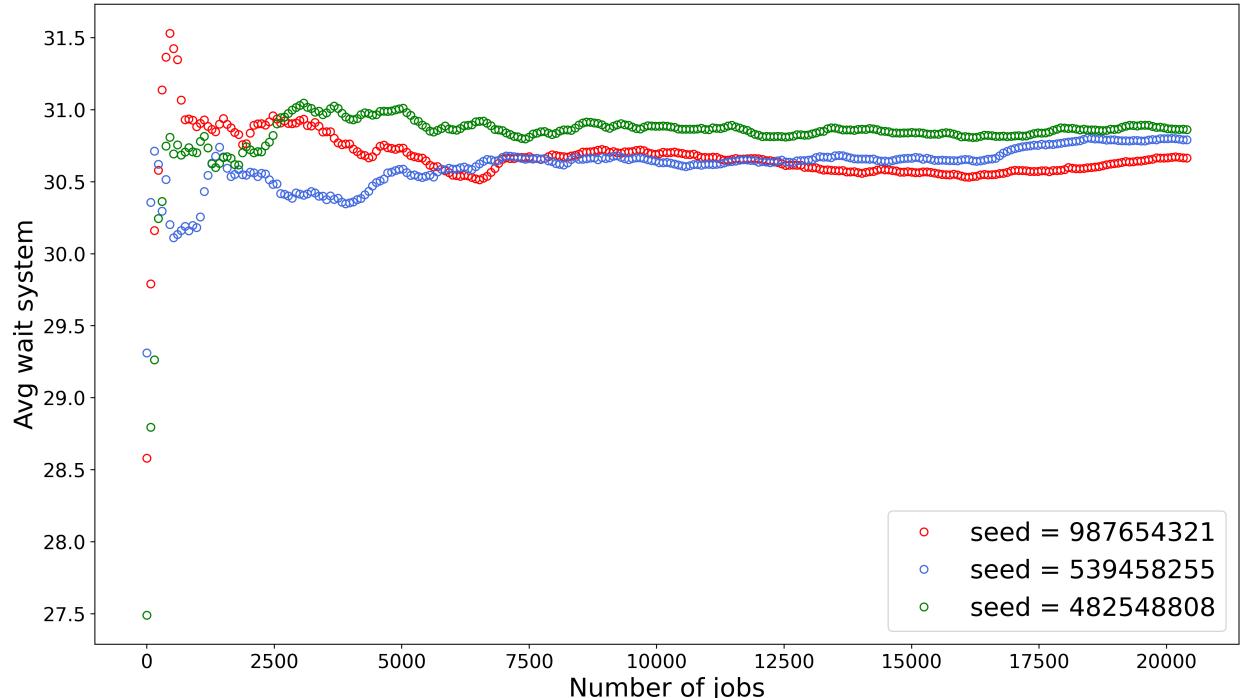


Figure 16: Average Wait System, 12:00-17:00, $\lambda_{arrival} = \frac{1}{5} \text{ min}$, Arcades: 12, repliche=64, Sampling_freq=75

Per quanto riguarda la quarta fascia oraria (22:00-08:00), è possibile osservare i risultati dell'analisi in figura 17.

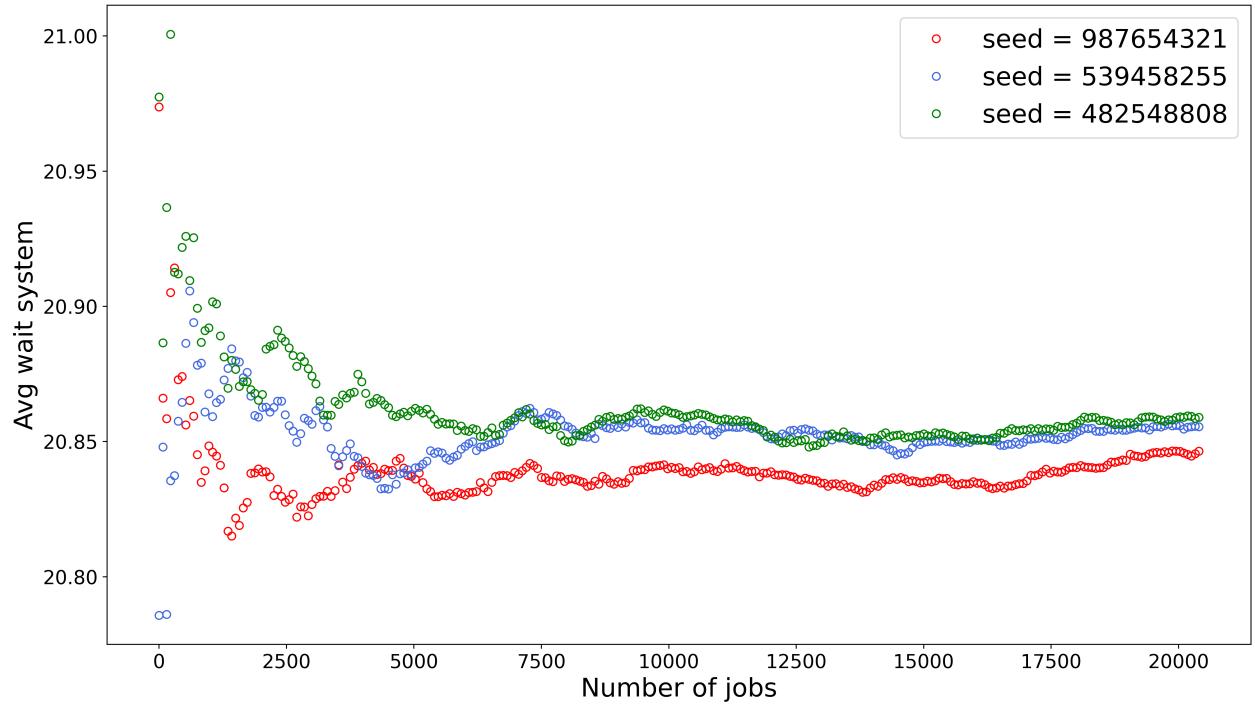


Figure 17: Average Wait System, 22:00-08:00, $\lambda_{arrival} = \frac{1}{35} \frac{job}{min}$, Arcades: 2, repliche=64, Sampling_freq=75

Come si evince dai risultati dell’analisi, il modello avanzato, come previsto, raggiunge la stazionarietà utilizzando la configurazione ottimale ottenuta nel modello base.

2.7.2 Analisi del comportamento stazionario

Considerati i risultati dell’analisi transiente, è possibile procedere allo studio del comportamento stazionario. Attraverso l’analisi del comportamento stazionario si vuole confrontare, con la configurazione ottima di nodi arcade, per ogni fascia oraria, i tempi di attesa nella coda Covid-19 sperimentati dagli utenti con green-pass, per quanto riguarda il modello base e quello avanzato, verificando che l’obiettivo 2 in 2.1 venga raggiunto.

In questa fase sono state effettuate simulazioni con tempi d’osservazione molto lunghi, utilizzando la tecnica delle batch-means per ottenere delle stime degli indici di prestazione in regime stazionario, evitando la problematica del bias dovuto alle condizioni iniziali del sistema. Cruciale in questa fase la scelta di due parametri: b e k , rispettivamente la grandezza del batch e il numero di batch in cui si divide la simulazione. Per la scelta di b è stata utilizzata la linea guida di Banks, Carson, Nelson, e Nicol (2001), la quale afferma che il batch size venga aumentato fintantoché l’autocorrelazione a lag 1 tra batch means sia minore di 0.2. Nel caso in esame è stato constatato che tale direttiva viene rispettata con un b pari a 256. Per la scelta del parametro k è stato usato il valore pari a 64 come consigliato dalle linee guida.

In seguito per ogni fascia oraria viene riportato il grafico relativo al tempo medio di attesa nella coda Covid-19, mediata su i seed, per il modello base e il modello avanzato.