

PMCSN Report

Marco Marcucci

marco.marcucci96@gmail.com
0286352

Giuseppe Lasco

giuseppe.lasco17@gmail.com
0286045

Valentina Falaschi

valentinayffalas@gmail.com
0295947

University of Roma Tor Vergata — July 10, 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, di cui vengono descritti 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 in loco, sottponendosi al controllo del proprio green-pass oppure, in mancanza di quest'ultimo, al test antigenico rapido;
- Attraverso l'acquisto di un biglietto online, esibendo al momento della prenotazione il proprio green-pass, saltando così i controlli.

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 nei nodi Arcade.

In figura 1 è possibile osservare lo schema ad alto livello del modello base.

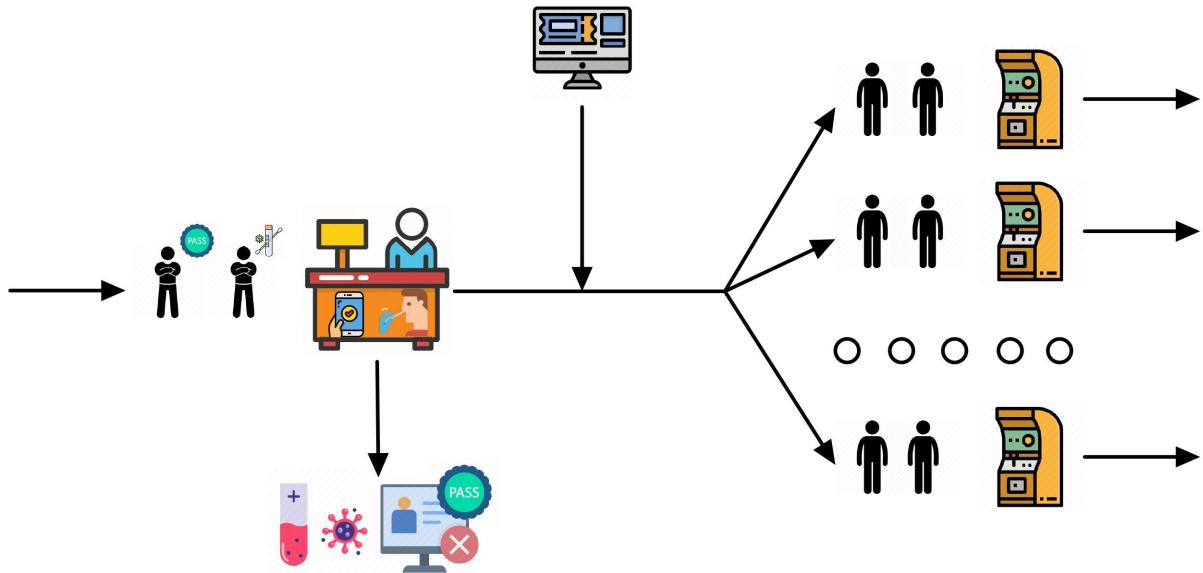


Figure 1: Base Model, High-level view

1.1 Obiettivi

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

$$\begin{aligned}
 G = & \#giocatori * costo biglietto \\
 - & \#giocatori * rimborso biglietto(attesa media coda) * costo biglietto \\
 - & \#arcade attivi * costo elettricità
 \end{aligned}$$

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

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 nel 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 un cliente in servizio, questo continuerà a processarlo, 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.

In figura 2 è possibile osservare lo schema a basso livello del modello base.

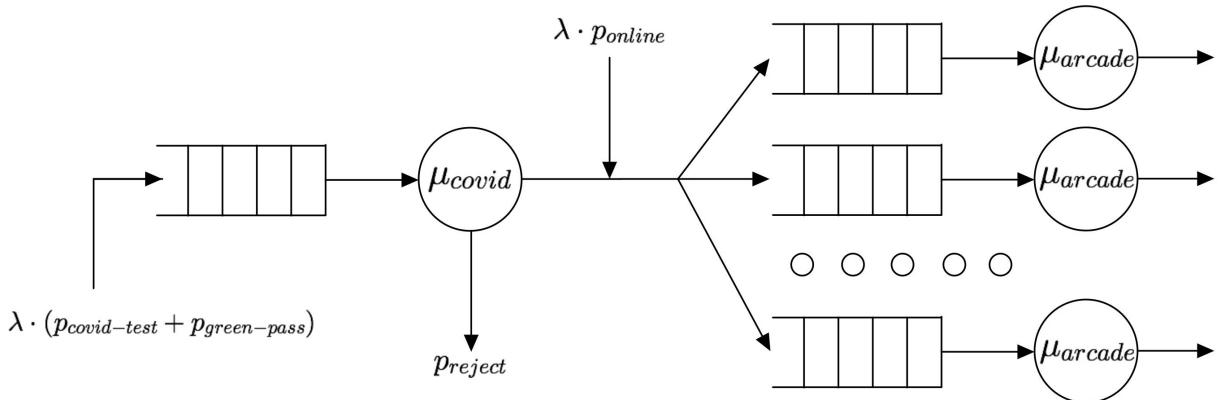


Figure 2: Base Model, Low-level view

1.3 Modello delle Specifiche

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 clienti/minuto (14 minuti/cliente), la media per la seconda fascia oraria è di 0.2 clienti/minuto (5 minuti/cliente), la media per la terza fascia oraria è di 0.07143 clienti/minuto (14 minuti/cliente), la media per la quarta fascia oraria è di 0.02857 clienti/minuto (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 con una distribuzione Normale con media 2 minuti (0.5 clienti/minuto) e varianza 2.25 minuti, troncata tra 1 e 3 minuti.

La distribuzione del tempo di servizio per il tampone è stata modellata con una distribuzione Normale con media 10 minuti (0.1 clienti/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 con una distribuzione Normale con media 15 minuti (0.06 clienti/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*, facendo particolare riferimento alle librerie 3 implementate dall'autore del libro di testo, e forniteci nel corso delle lezioni. 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 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 ← 0, numero nodi do
10:       if nodi non vuoti then
11:         Aggiornamento statistiche
12:         Avanzamento del clock
13:         if Evento = Cambio fascia oraria then
14:           Selezione tasso arrivi corretto
15:         if Evento = Arrivo then                                ▷ Arrivo
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                                              ▷ 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
```

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 stati eseguiti diversi run variando determinati parametri. I risultati possono essere osservati in tabella 1 e 2.

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 |

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

Per quanto riguarda la tabella 1, 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 |

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

Per quanto riguarda la tabella 2, si può notare come la relazione di Little, applicabile in quanto il modello soddisfa le condizioni sufficienti, sia soddisfatta e come l'utilizzazione e il tempo di interarrivo medio, rispettivamente, decresca ed aumenti al crescere 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

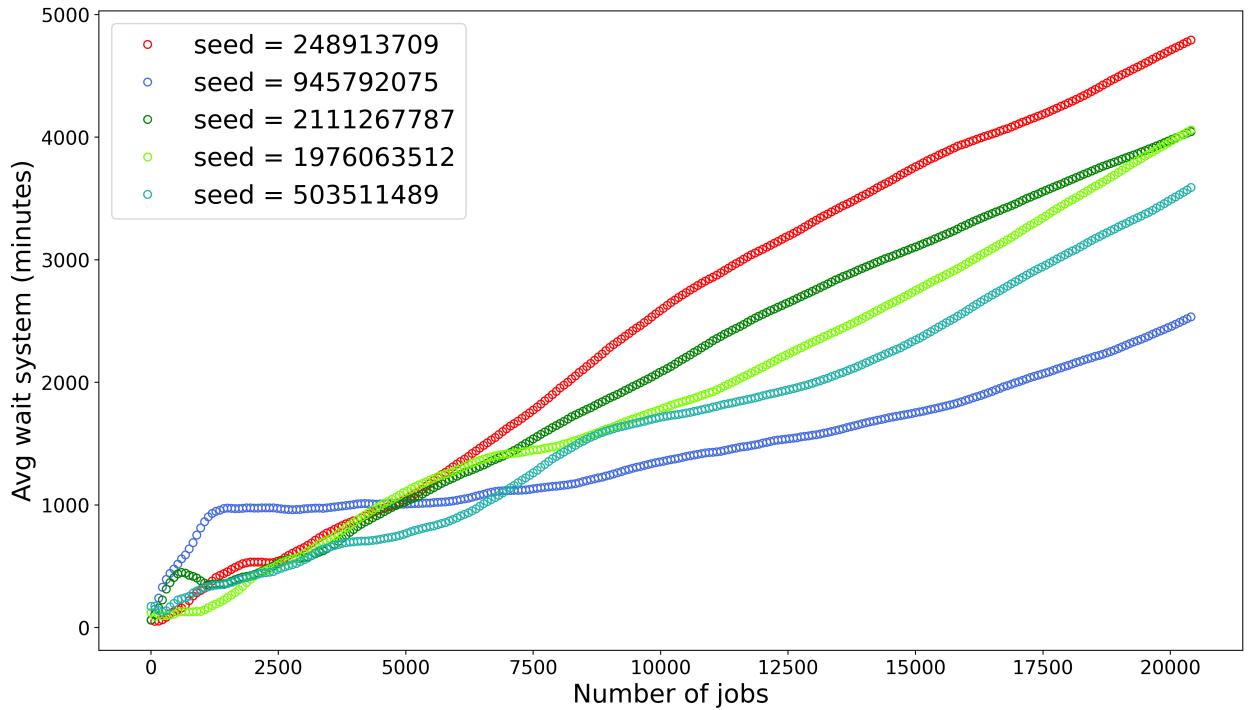
Attraverso l'analisi del comportamento transiente viene studiato il tempo medio di risposta del sistema, per ogni fascia oraria, utilizzando il metodo delle repliche. In particolare, vengono effettuati run di simulazione consecutivi, utilizzando ogni volta come punto di partenza lo stato del sistema raggiunto dalla replica precedente. Il seed, infatti, viene settato una volta, all'inizio della prima replica. La tecnica delle

repliche permette di calcolare stime puntuale 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 $\lambda_{arrival} < \frac{1}{4.1}$.

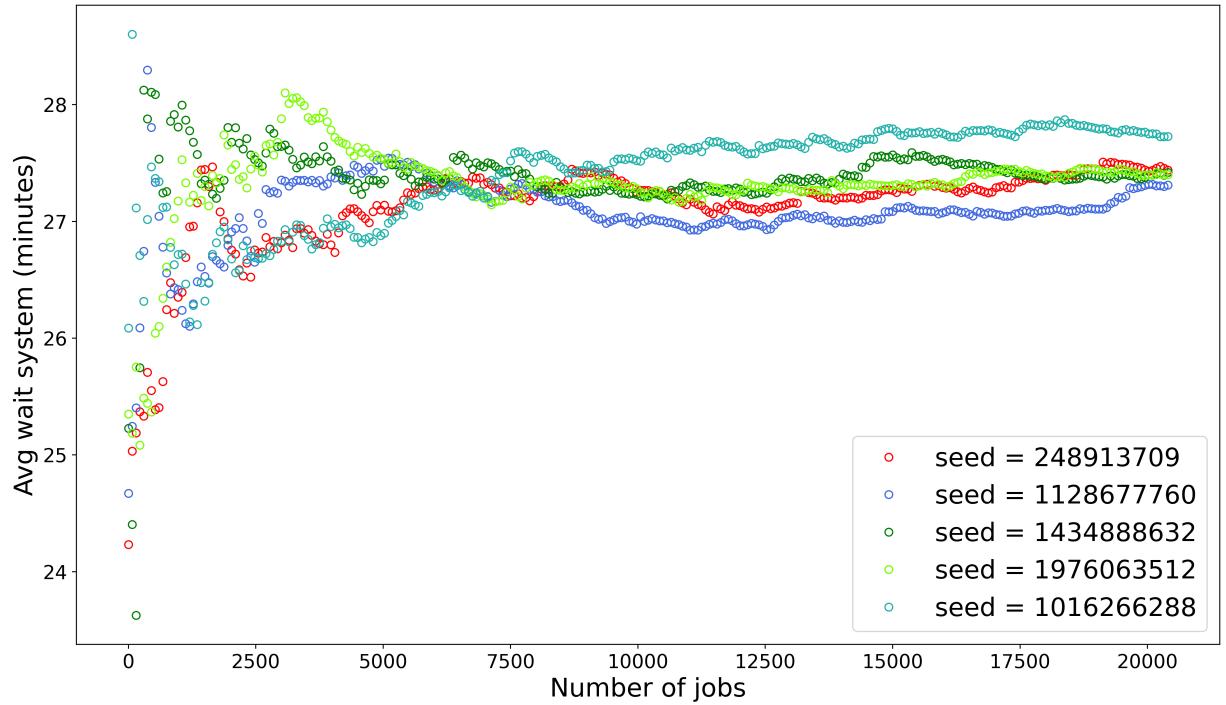
Per ciascuna fascia oraria viene riportato un grafico, il quale ha lo scopo di mostrare l'andamento del tempo medio di risposta del sistema ottenuto per 5 diverse repliche.

Come riportato nell'articolo *MANET Simulation Studies: The Incredibles*, un errore tipico che viene commesso nella simulazione è quello di non settare propriamente il seed, questo comporta la produzione di repliche non indipendenti e quindi nega la possibilità di condurre le comuni tecniche di analisi. Di conseguenza, al fine di garantire repliche indipendenti, al termine di ogni replica viene recuperato il seed con cui quella replica è terminata, tramite l'ausilio della `getSeed()`, così che la replica successiva possa utilizzare come stato iniziale quello raggiunto nella prova precedente. Tali considerazioni valgono per tutti i grafici successivi. 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 nelle successive figure. Tali fasce, infatti, hanno lo stesso $\lambda_{arrival}$ da modellazione.



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

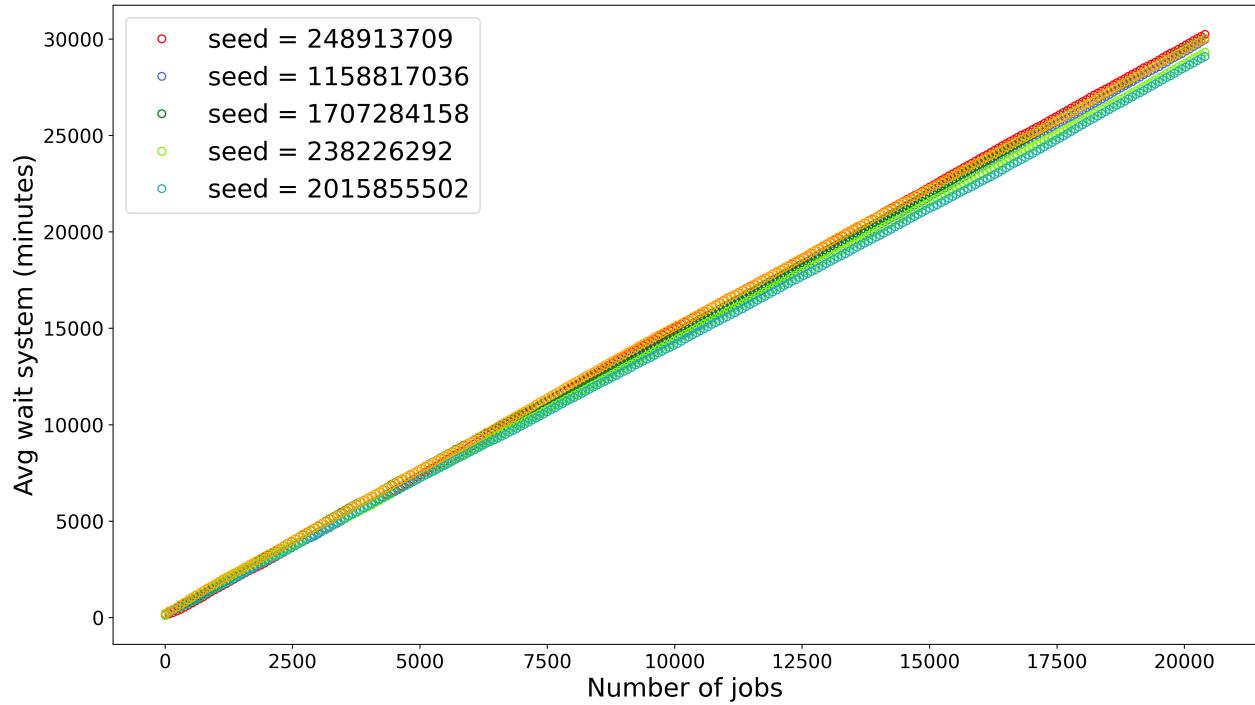
Quello che si può evincere dal precedente grafico è che un solo Arcade, per la fascia oraria 08:00-12:00 e 17:00-22:00, non permette di raggiungere la stazionarietà. Di seguito viene riportato lo stesso studio considerando due Arcade.



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

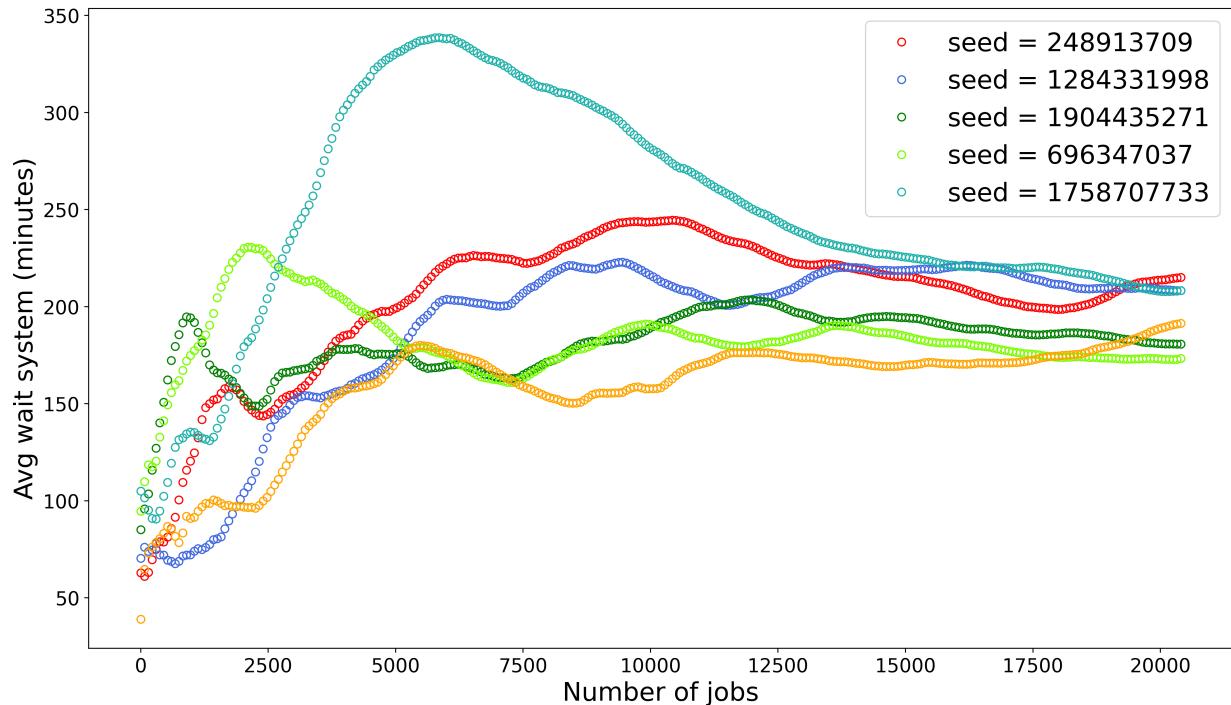
Come si evince dal grafico, il numero minimo di nodi Arcade necessari per il raggiungimento della stazionarietà, nella fascia oraria 08:00-12:00 e 17:00-22:00, è pari a 2. Infatti, il tempo medio di risposta del sistema assume un comportamento stabile dopo il processamento di circa 5000 jobs. Tali considerazioni, 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 nella seguente figura:



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

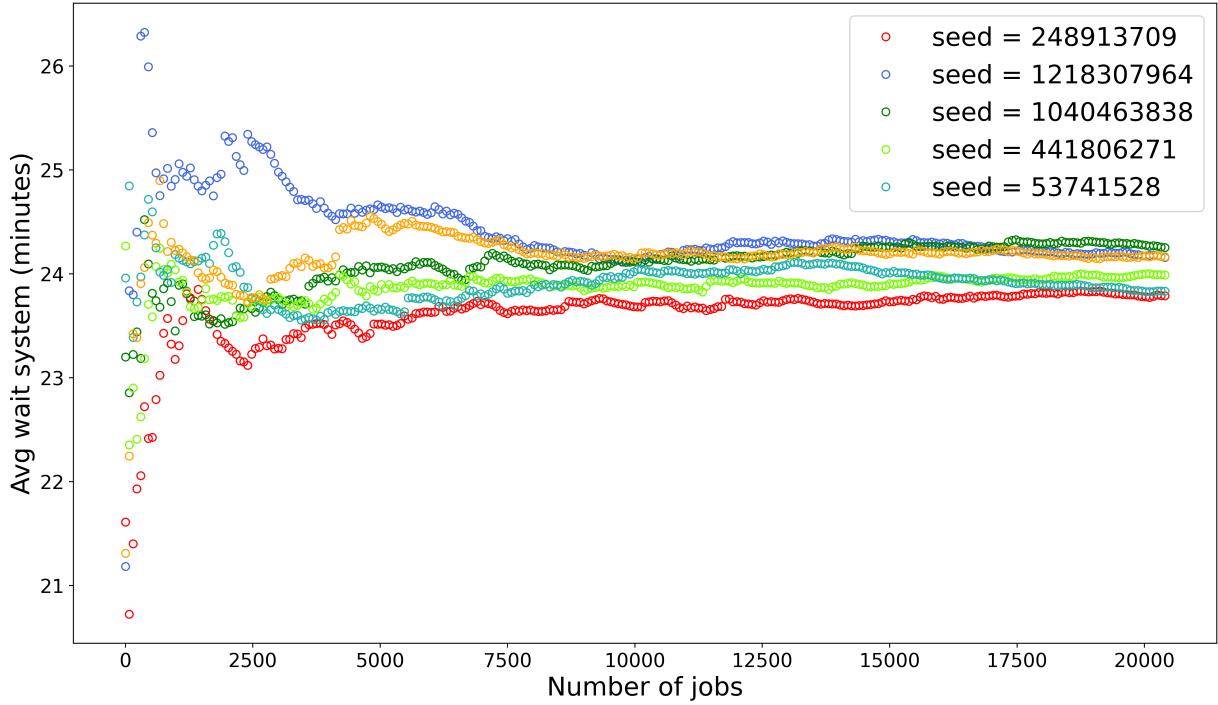
Quello che si può evincere dal precedente grafico è che due Arcade, per la fascia oraria 12:00-17:00, non permettono di raggiungere la stazionarietà. Di seguito viene riportato lo stesso studio considerando tre Arcade.



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

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

nella seguente figura:



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

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

Di seguito vengono riportati i risultati dell'analisi in forma tabellare:

| Fascia oraria | Media tempo di risposta | Deviazione standard | Intervallo di confidenza |
|---------------|-------------------------|---------------------|---------------------------|
| 08:00-12:00 | 27.49805560398169 | 0.2705507433514275 | ± 0.06811581483950424 |
| 12:00-17:00 | 202.45149567658495 | 38.97821056475186 | ± 9.813436624548967 |
| 17:00-22:00 | 27.49805560398169 | 0.2705507433514275 | ± 0.06811581483950424 |
| 22:00-8:00 | 24.018494887900026 | 0.14272563774733274 | ± 0.03593364037081727 |

Tali risultati sono stati ottenuti utilizzando la procedura relativa al metodo delle repliche, descritta precedentemente, utilizzando la configurazione minima di nodi Arcade, per ogni fascia oraria, necessaria per raggiungere la stazionarietà.

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 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 i batch 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 alla funzione guadagno e del tempo medio di risposta del sistema al variare del numero di nodi Arcade, mediata sui 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 3. Tali fasce, infatti, hanno lo stesso $\lambda_{arrival}$ da modellazione.

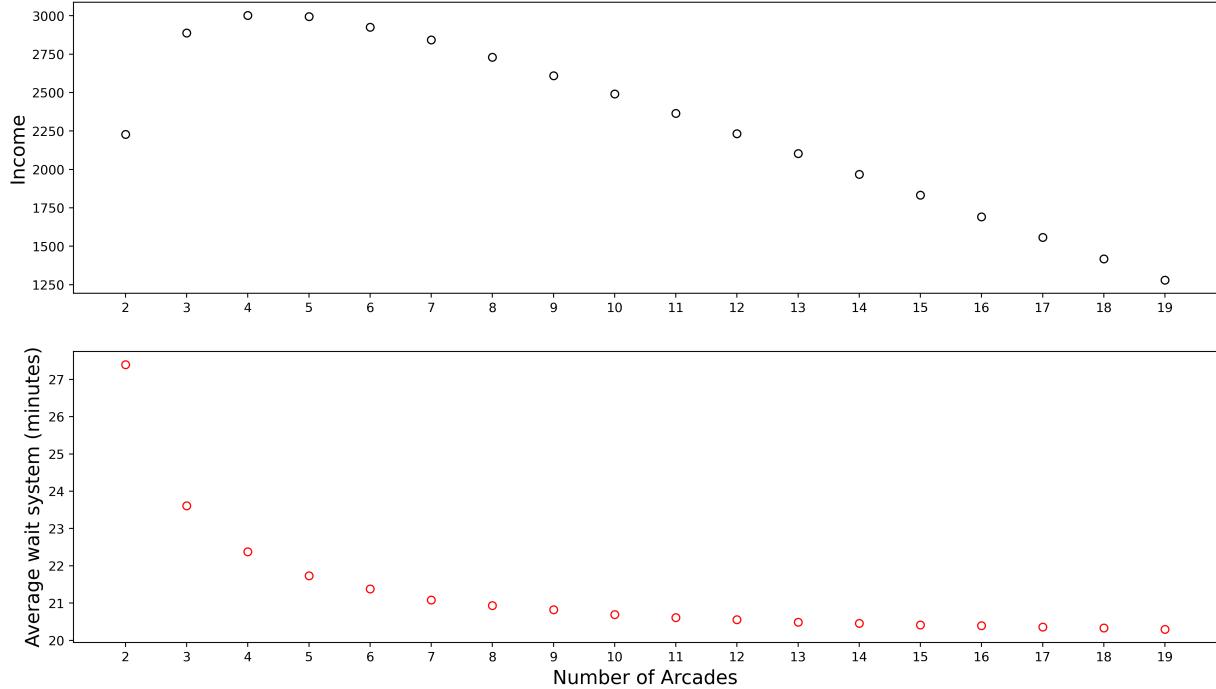


Figure 3: Avg Wait System & Income, 08:00-12:00 & 17:00-22:00,
 $\lambda_{arrival} = \frac{1}{14} \frac{job}{min}$, b=256, k=64

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 4.

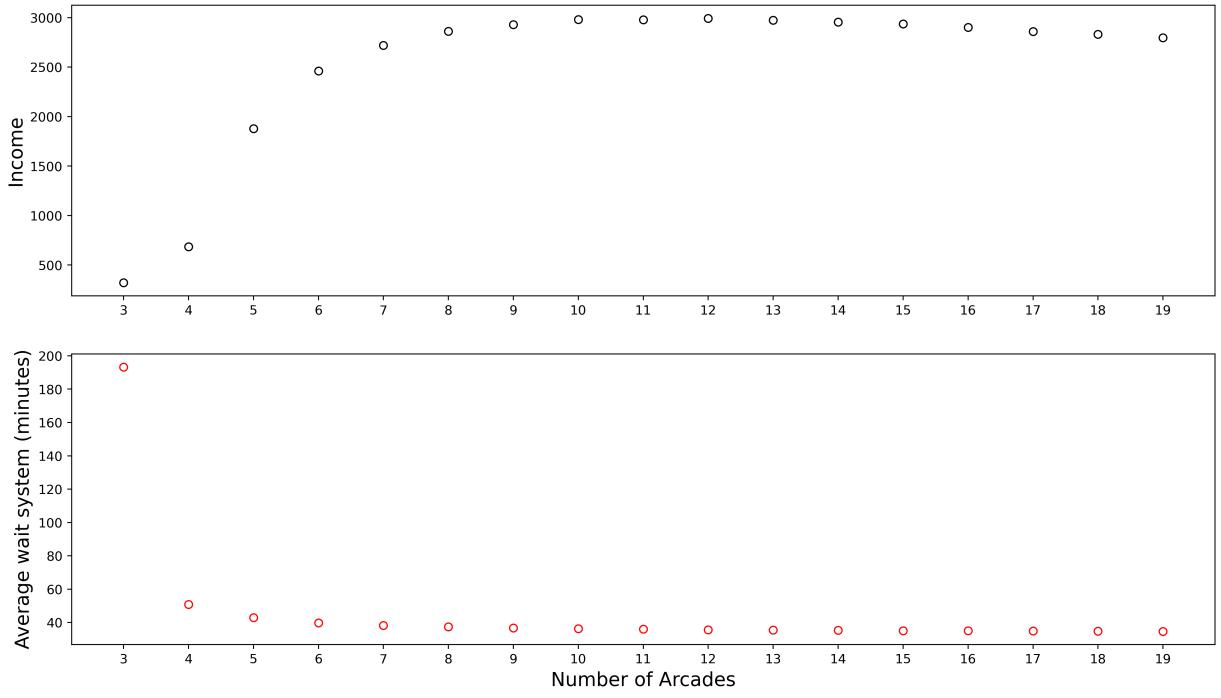


Figure 4: Avg Wait System & Income, 12:00-17:00, $\lambda_{arrival} = \frac{1}{5} \frac{job}{min}$, $b=256$, $k=64$

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 5.

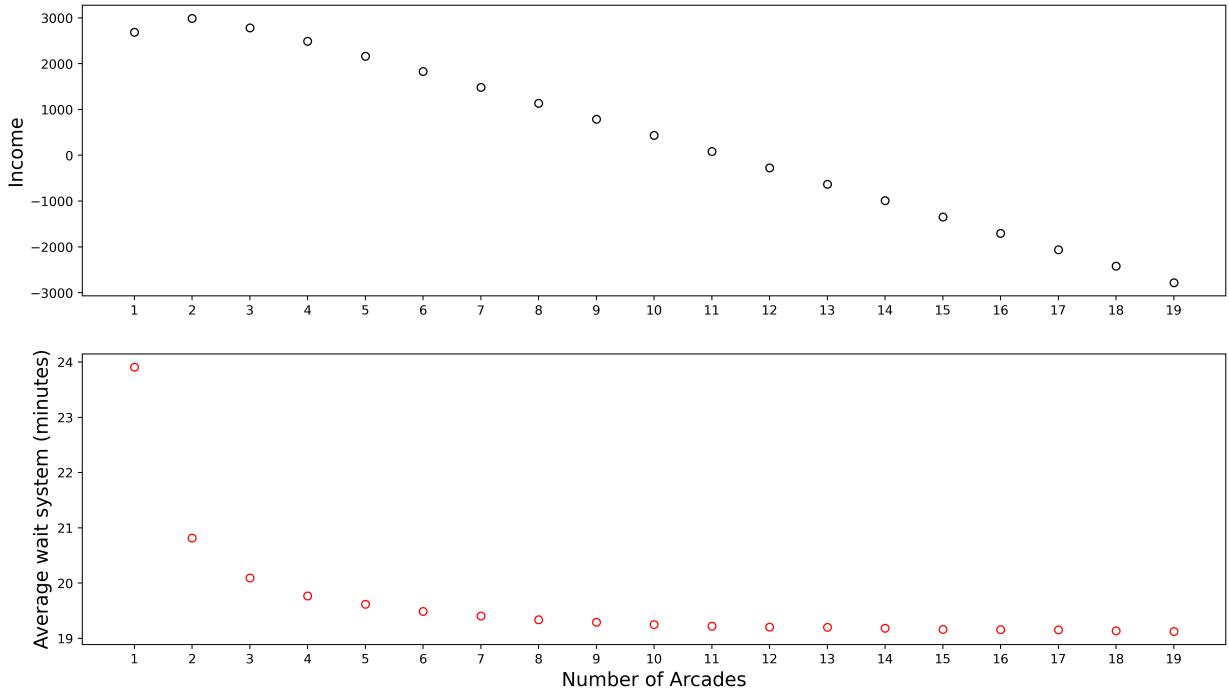


Figure 5: Avg Wait System & Income, 22:00-08:00, $\lambda_{arrival} = \frac{1}{35} \frac{job}{min}$, $b=256$, $k=64$

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

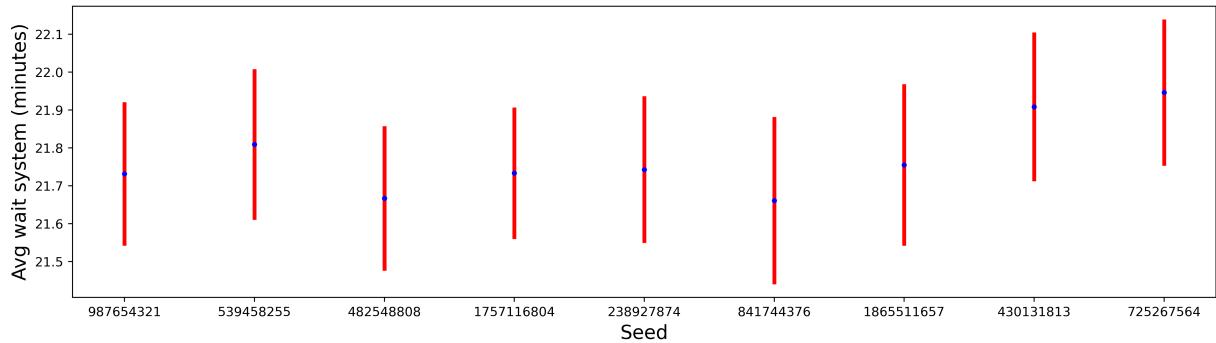
In tabella 3 è possibile individuare la configurazione ottimale dei nodi Arcade, per ogni fascia oraria.

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 |

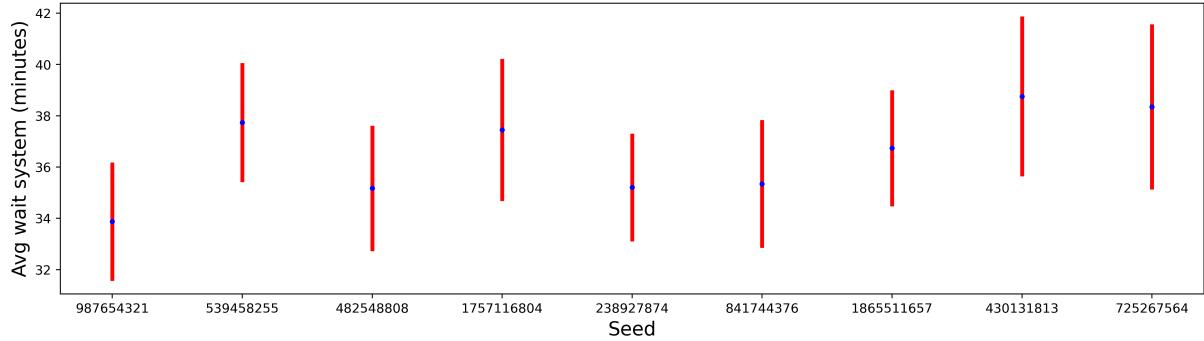
* Numero di nodi Arcade nel sistema.

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%.



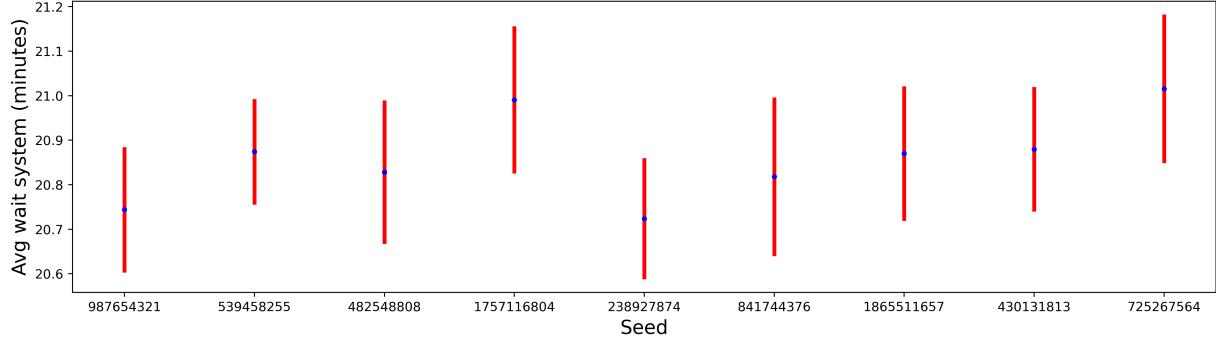
| SEED | MEAN VALUE | STD | CONFIDENCE INTERVAL | CONFIDENCE LEVEL |
|------------|--------------------|--------------------|---------------------------|------------------|
| 987654321 | 21.73139159874175 | 0.7278992755036976 | ± 0.18962315438362418 | 95% |
| 539458255 | 21.809212808024988 | 0.7628715753780063 | ± 0.19873369761589693 | 95% |
| 482548808 | 21.666685360018118 | 0.7320529654790335 | ± 0.19070522139751125 | 95% |
| 1757116804 | 21.733187590110223 | 0.6658261867805824 | ± 0.1734526548624279 | 95% |
| 238927874 | 21.742766801070932 | 0.7434122717449919 | ± 0.19366440484259734 | 95% |
| 841744376 | 21.660830598797475 | 0.8470157213826731 | ± 0.2206538710867113 | 95% |
| 1865511657 | 21.755024295060352 | 0.8186747356682376 | ± 0.21327083432548677 | 95% |
| 482548808 | 21.666685360018118 | 0.7320529654790335 | ± 0.19070522139751125 | 95% |
| 430131813 | 21.90874969573461 | 0.754130418552268 | ± 0.19645656149825008 | 95% |
| 725267564 | 21.946094727089662 | 0.7402658474250582 | ± 0.19284473799492488 | 95% |

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



| SEED | MEAN VALUE | STD | CONFIDENCE INTERVAL | CONFIDENCE LEVEL |
|------------|-------------------|--------------------|--------------------------|------------------|
| 987654321 | 33.87434704959296 | 8.85265866112511 | ± 2.306183171898936 | 95% |
| 539458255 | 37.73964266185599 | 8.903839742578084 | ± 2.3195162228257447 | 95% |
| 482548808 | 35.16960459487452 | 9.37057765306069 | ± 2.441104906637649 | 95% |
| 1757116804 | 37.44713454948432 | 10.645813492771289 | ± 2.773313291295871 | 95% |
| 238927874 | 35.20481120340799 | 8.052652349810858 | ± 2.0977756004348183 | 95% |
| 841744376 | 35.34220932149259 | 9.567809468452161 | ± 2.4924852558672326 | 95% |
| 1865511657 | 36.73704579213175 | 8.689109509003513 | ± 2.2635774060110605 | 95% |
| 482548808 | 35.16960459487452 | 9.37057765306069 | ± 2.441104906637649 | 95% |
| 430131813 | 38.75500315059004 | 11.96718419666216 | ± 3.1175401498931827 | 95% |
| 725267564 | 38.34969784084128 | 12.355261767159236 | ± 3.218637232332599 | 95% |

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



| SEED | MEAN VALUE | STD | CONFIDENCE INTERVAL | CONFIDENCE LEVEL |
|------------|--------------------|--------------------|---------------------------|------------------|
| 987654321 | 20.743725727271343 | 0.540500729785019 | ± 0.14086446025846807 | 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 | 0.521504959598914 | ± 0.13585590587381846 | 95% |
| 841744376 | 20.81791117315198 | 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% |

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

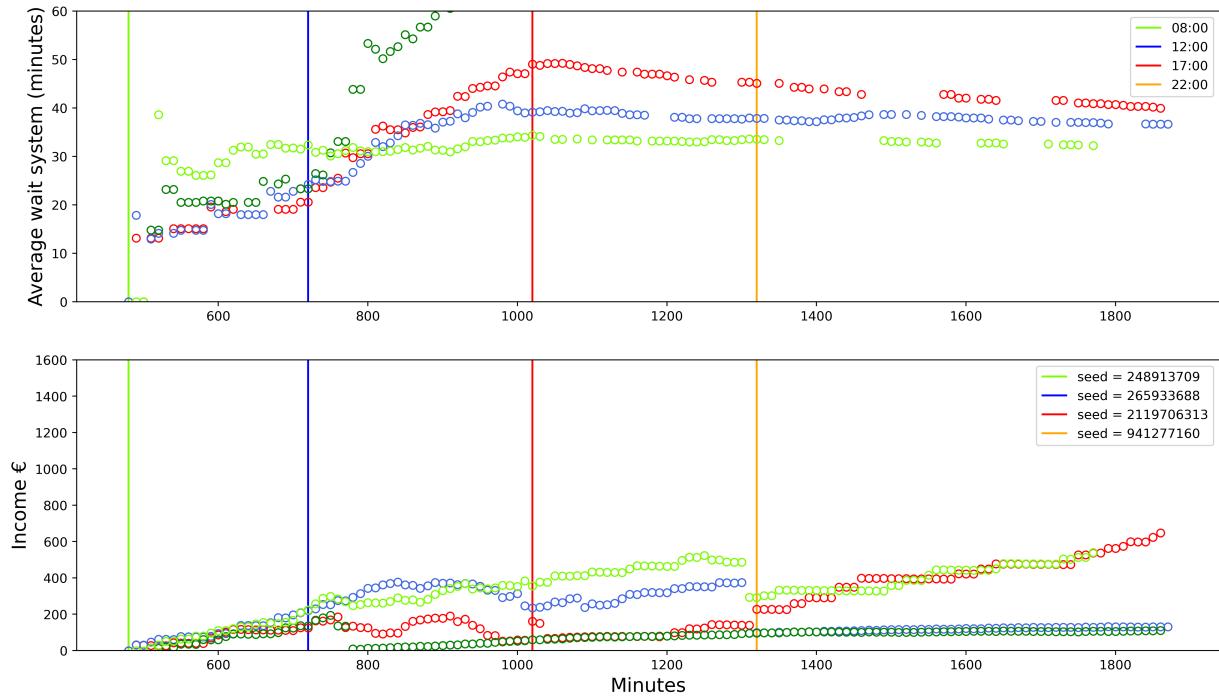
1.7.3 Analisi guadagno giornaliero

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

- Configurazione minima (2 Arcade 08:00-12:00 & 17:00-22:00, 3 Arcade 12:00-17:00, 1 Arcade 22:00-08:00);
- Configurazione ottimale(5 Arcade 08:00-12:00 & 17:00-22:00, 12 Arcade 12:00-17:00, 2 Arcade 22:00-08:00);

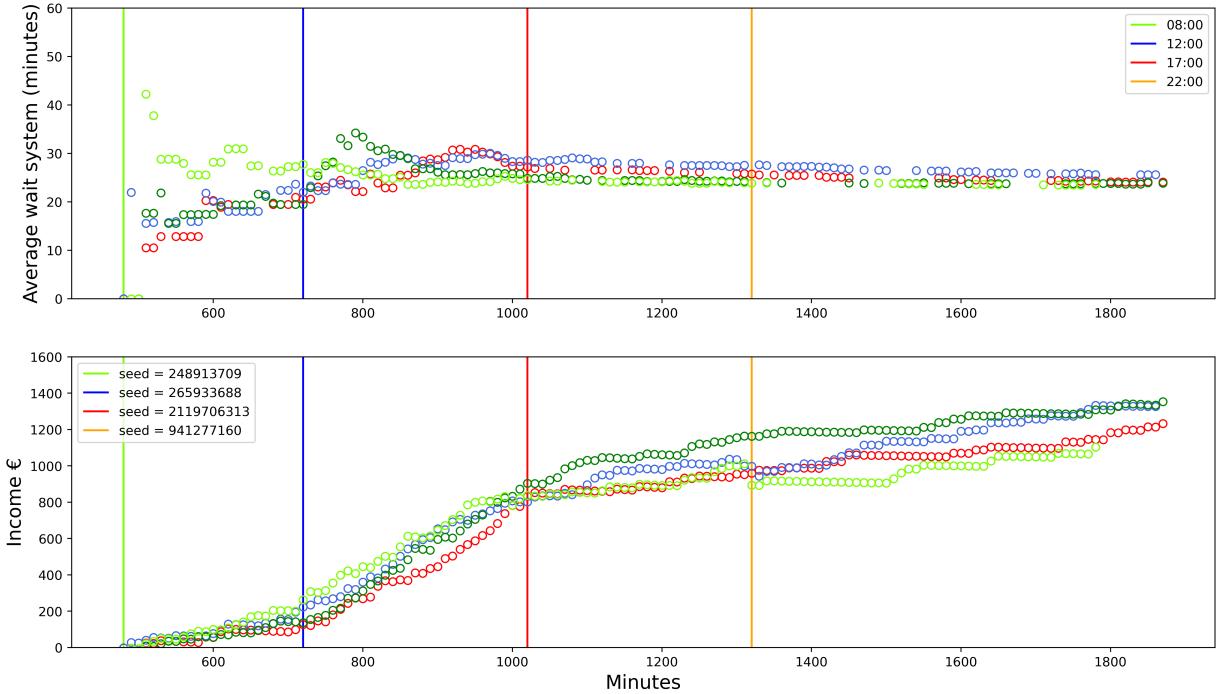
- Configurazione massima(20 Arcade 08:00-12:00 & 17:00-22:00, 20 Arcade 12:00-17:00, 20 Arcade 22:00-08:00).

Di seguito vengono mostrati i risultati dei valori puntuali relativi al tempo medio di risposta del sistema e al guadagno. Per ogni configurazione di nodi Arcade viene riportato il grafico relativo a 4 diverse repliche:



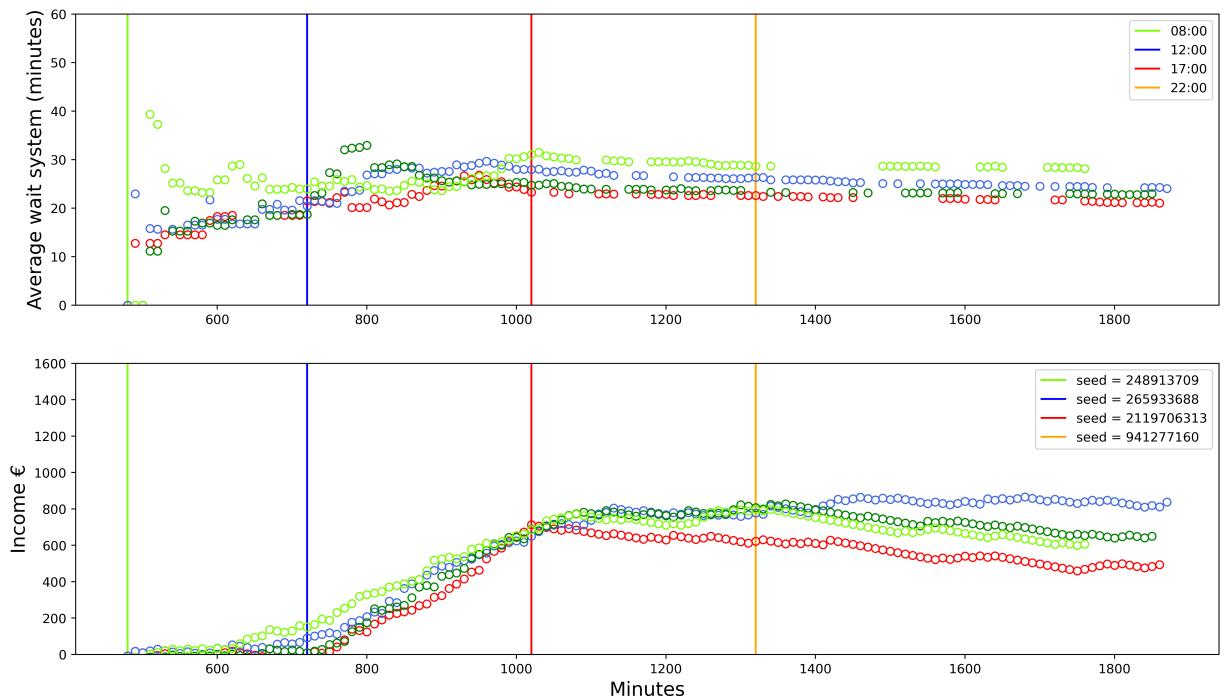
Average Wait System & Income Daily, Base Model, minimal configuration,
sampling frequency=10 min

In questo primo grafico è possibile notare che per la configurazione minimale di nodi Arcade il QoS sul tempo medio di risposta del sistema non viene soddisfatto ed inoltre i guadagni risultano essere bassi, non si superano mai i 1000 euro.



Average Wait System & Income Daily, Base Model, best configuration,
sampling frequency=10 min

Dall'analisi della precedente figura è possibile notare che, per una configurazione ottima di nodi Arcade, il QoS sul tempo medio di risposta del sistema viene sempre rispettato, inoltre anche i guadagni giornalieri risultano più elevati rispetto la configurazione minimale precedentemente descritta.



Average Wait System & Income Daily, Base Model, max configuration,
sampling frequency=10 min

Dall'analisi della precedente figura è possibile notare che per una configurazione massima di nodi

Arcade, sebbene il QoS sul tempo medio di risposta del sistema venga rispettato, i guadagni giornalieri sono bassi, questo è dovuto alle maggiori spese necessarie per installare più Arcade.

Successivamente, è stato svolto un ulteriore studio in cui: per ogni configurazione, i valori ottenuti dalle 64 repliche vengono mediati incrementalmente, utilizzando l'algoritmo di Welford. Inoltre, sono stati utilizzati 3 seed iniziali diversi, in modo tale da simulare il comportamento in condizioni aleatorie differenti. I 3 seed, allo scopo di garantire indipendenza tra i diversi esperimenti, sono stati selezionati tramite la funzione `getSeed()` posta dopo ogni 64 repliche, così che per le prove successive venga usato come stato iniziale quello raggiunto nella prova precedente. In questo modo si evitano possibili sovrapposizioni degli eventi generati dalle singole repliche. È possibile osservare il risultato di tale analisi in figure 6, 7 e 8.

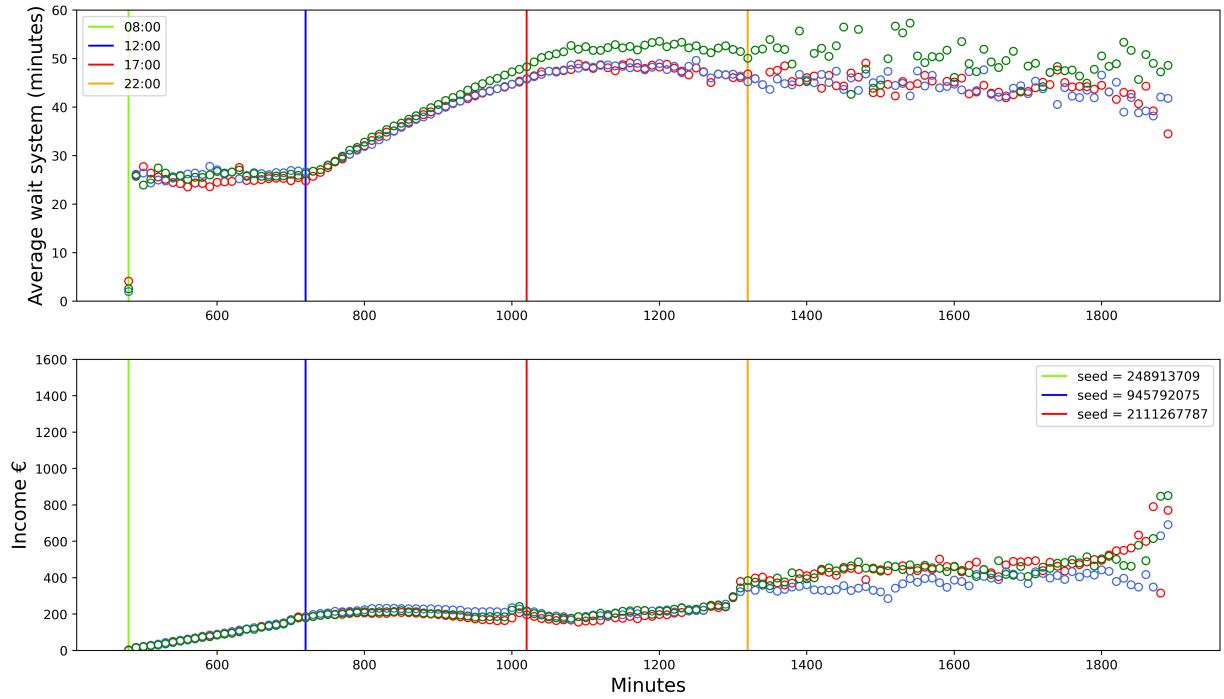


Figure 6: Average Wait System & Income Daily, Base Model, minimal configuration, sampling frequency=10 min

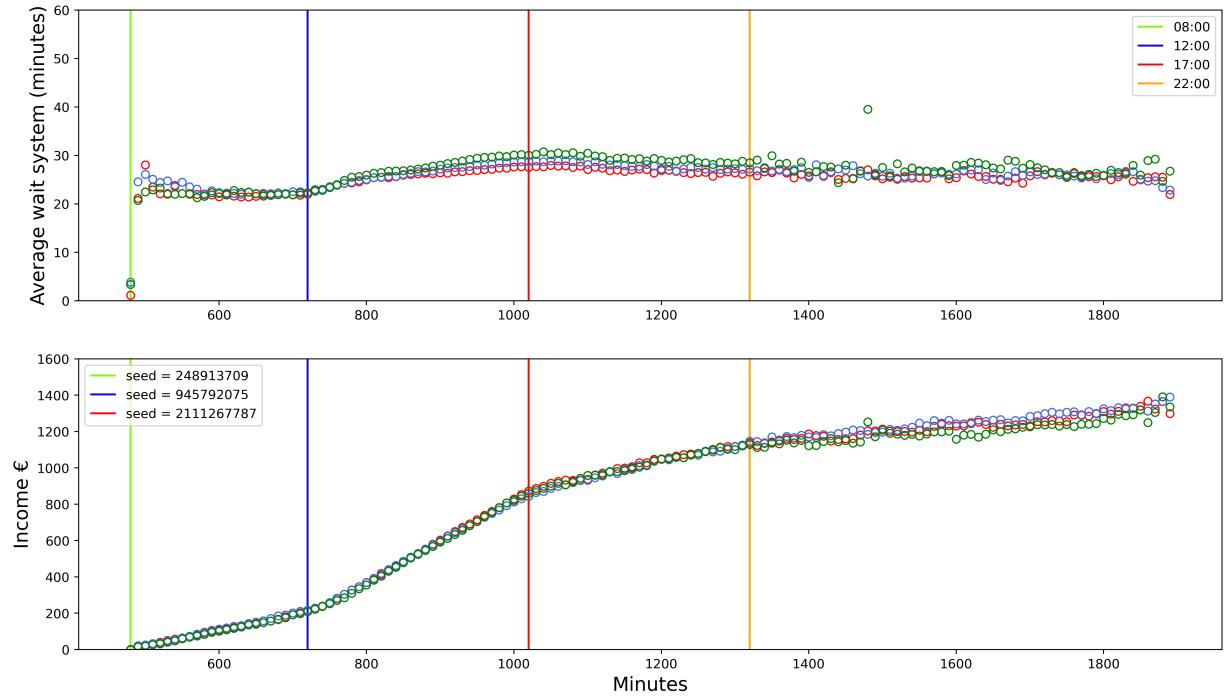


Figure 7: Average Wait System & Income Daily, Base Model, best configuration, sampling frequency=10 min

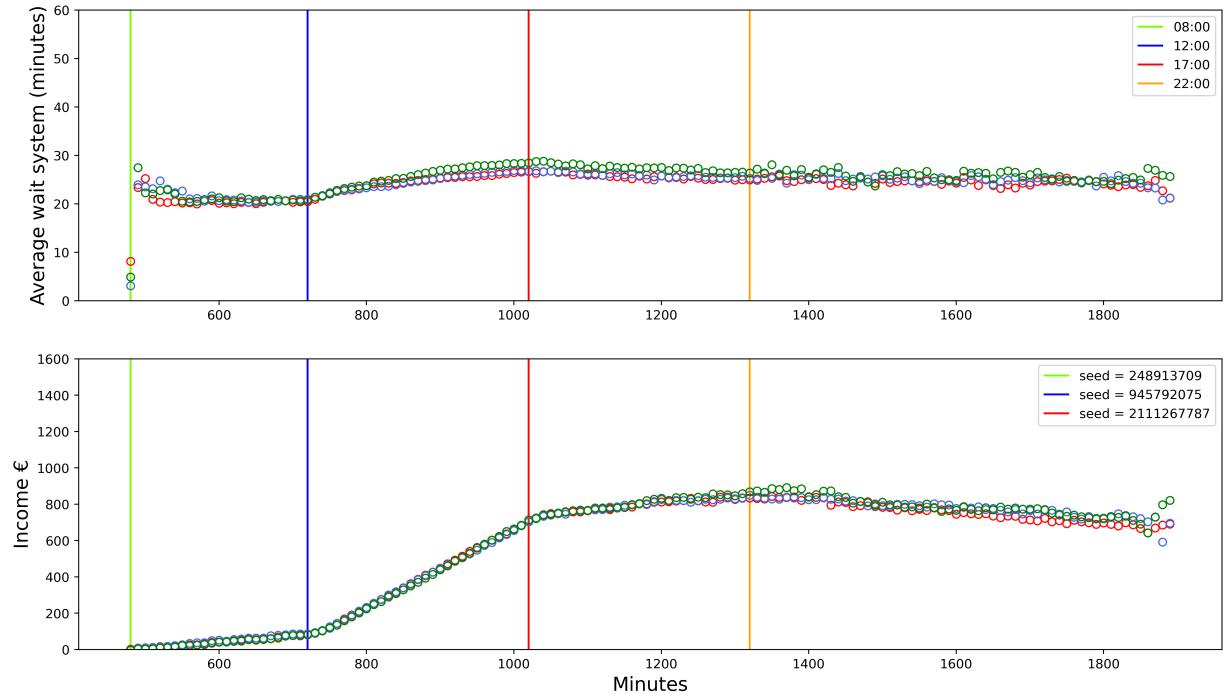


Figure 8: Average Wait System & Income Daily, Base Model, maximum configuration, sampling frequency=10 min

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 avere la precedenza rispetto agli utenti che sono in possesso di un biglietto standard.

In figura 9 è possibile osservare lo schema ad alto livello del modello avanzato.

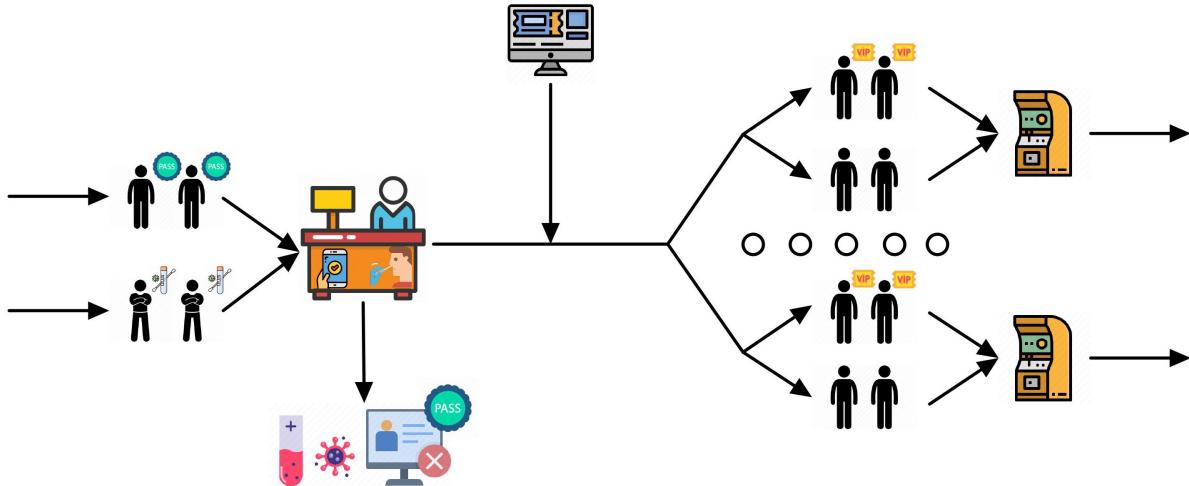


Figure 9: Advanced Model, High-level view

2.1 Obiettivi

Il primo obiettivo dello studio avanzato è quello di migliorare i profitti della sala giochi, rispetto al modello base. A tale scopo è stata definita una nuova funzione guadagno:

$$G = \#giocatori\ standard * costo\ biglietto\ standard + \#giocatori\ premium * costo\ biglietto\ premium - \#giocatori\ standard * rimborso\ biglietto(\text{attesa}\ media\ coda\ standard) * costo\ biglietto\ standard - \#giocatori\ premium * rimborso\ biglietto(\text{attesa}\ media\ coda\ premium) * costo\ biglietto\ premium - \#arcade\ attivi * costo\ elettricità$$

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 nella coda Covid-19, inferiore a quello del modello base, per gli utenti che possiedono il green-pass.

2.2 Modello Concettuale

Nel modello avanzato la sala giochi è modellata attraverso una rete aperta composta da N nodi rappresentanti i videogiochi (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 inferiore a chi esegue il test antigenico rapido.

Non sono state scelte code di priorità con prelazione, che avrebbero avuto prestazioni migliori per gli obiettivi prefissati, poiché 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.

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 nel 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 un cliente in servizio, questo continuerà a processarlo, 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.

In figura 10 è possibile osservare lo schema a basso livello del modello avanzato.

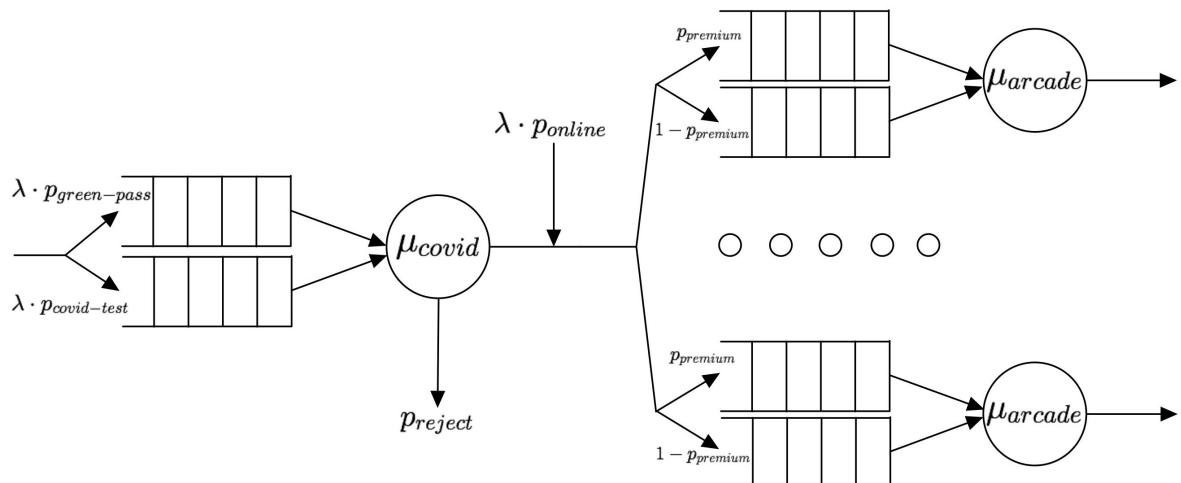


Figure 10: Advanced Model, Low-level view

2.3 Modello delle Specifiche

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 clienti/minuto (14 minuti/cliente), la media per la seconda fascia oraria è di 0.2 clienti/minuto (5 minuti/cliente), la media per la terza fascia oraria è di 0.07143 clienti/minuto (14 minuti/cliente), la media per la quarta fascia oraria è di 0.02857 clienti/minuto (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 con una distribuzione Normale con media 2 minuti (0.5 clienti/minuto) e varianza 2.25 minuti, troncata tra 1 e 3 minuti.

La distribuzione del tempo di servizio per il tampone è stata modellata con una distribuzione Normale con media 10 minuti (0.1 clienti/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 con una distribuzione Normale con media 15 minuti (0.06 clienti/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.

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*, facendo particolare riferimento alle librerie 3 implementate dall'autore del libro di testo, e forniteci nel corso delle lezioni. 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 Multiqueue Priority - Rete Aperta

```

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

```

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 stati eseguiti 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, $\lambda_{system} = \frac{1}{14} \frac{job}{min}$, #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, $\lambda_{system} = \frac{1}{14} \frac{job}{min}$, #jobs = 517028.

Per quanto riguarda la tabella 5, si può notare come la relazione di Little, applicabile in quanto il modello soddisfa le condizioni sufficienti, sia soddisfatta e come l'utilizzazione e il tempo di interarrivo medio, rispettivamente, decresca ed aumenti al crescere 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.

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

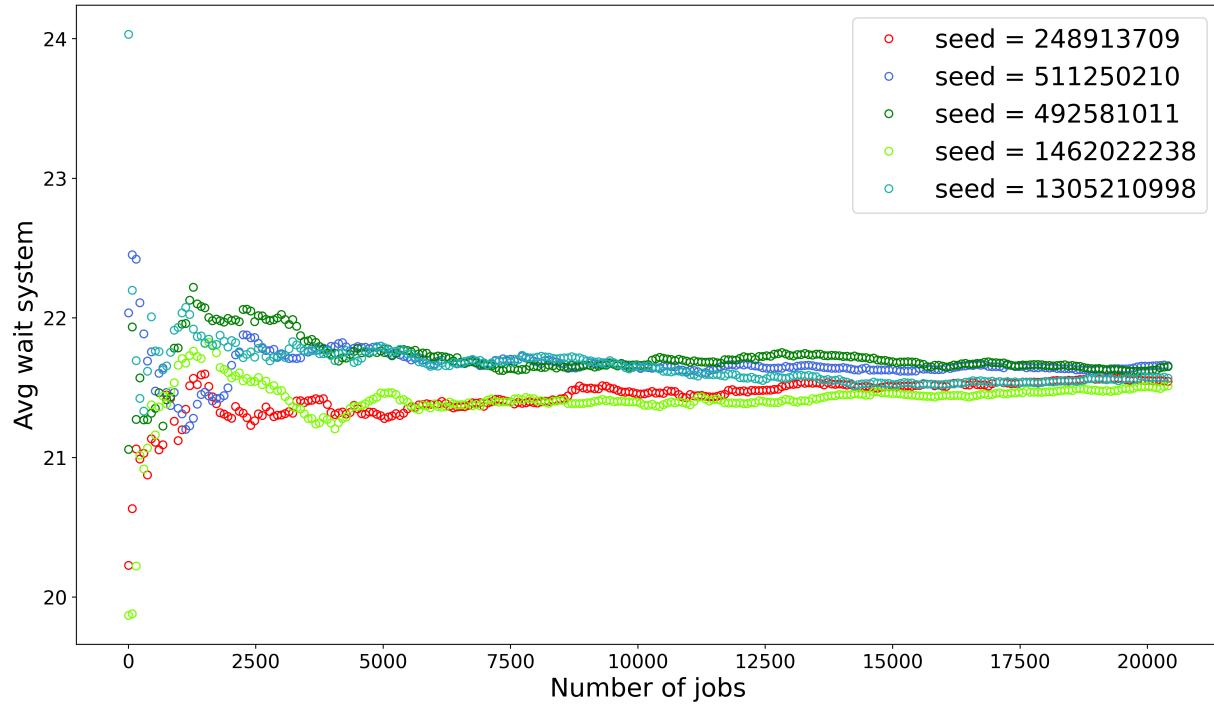
Attraverso l'analisi del comportamento transiente viene studiato il tempo medio di risposta del sistema, per ogni fascia oraria, utilizzando il metodo delle repliche. In particolare, vengono effettuate run di simulazione consecutive, utilizzando ogni volta come punto di partenza lo stato del sistema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

Di seguito vengono riportati i risultati dell'analisi in forma tabellare:

| Fascia oraria | Media tempo di risposta | Deviazione standard | Intervallo di confidenza |
|---------------|-------------------------|---------------------|----------------------------|
| 08:00-12:00 | 21.631579666207035 | 0.0913023716394926 | ± 0.022986946418864715 |
| 12:00-17:00 | 30.8211009210939 | 0.8122768001631648 | ± 0.204504690813104 |
| 17:00-22:00 | 21.631579666207035 | 0.0913023716394926 | ± 0.022986946418864715 |
| 22:00-8:00 | 20.846428788388543 | 0.07370313807357468 | ± 0.018556035898925353 |

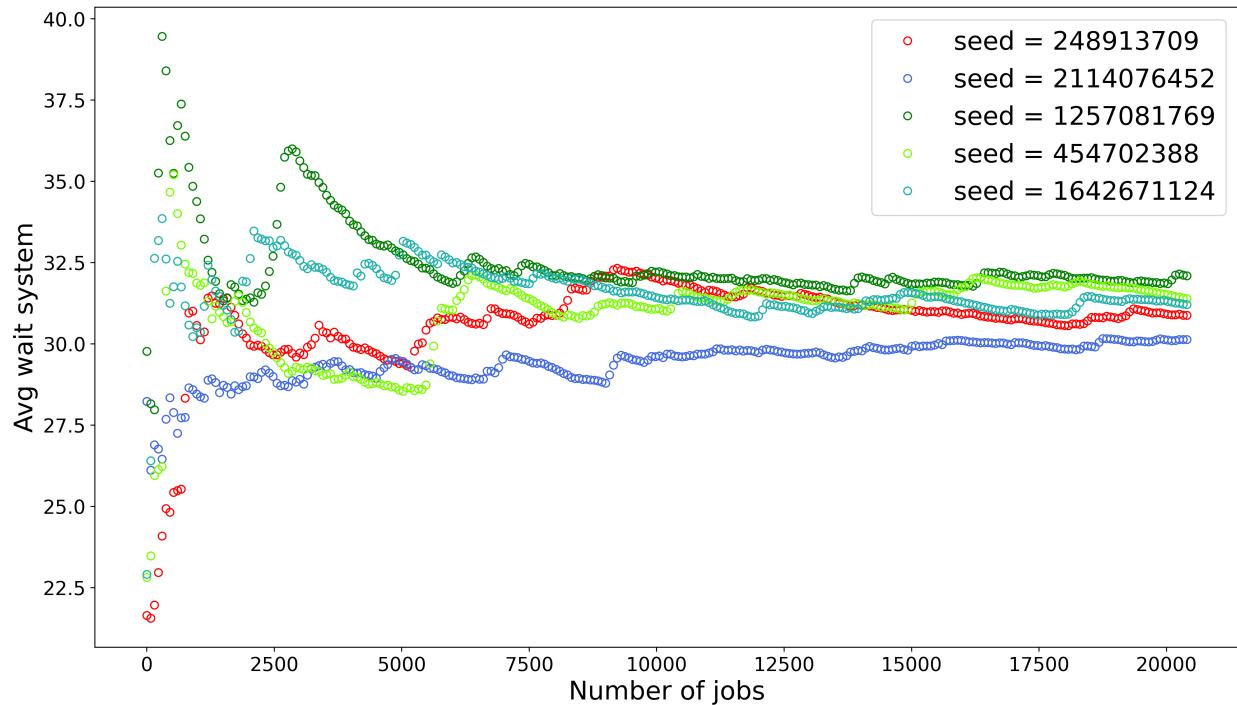
Tali risultati sono stati ottenuti utilizzando la procedura relativa al metodo delle repliche, descritta precedentemente, utilizzando la configurazione minima di nodi Arcade, per ogni fascia oraria, necessaria per raggiungere la stazionarietà.

Di seguito vengono inoltre mostrati i grafici relativi a 5 diverse repliche, per ogni fascia oraria. 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 nella seguente figura. Tali fasce, infatti, hanno lo stesso $\lambda_{arrival}$ da modellazione.



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

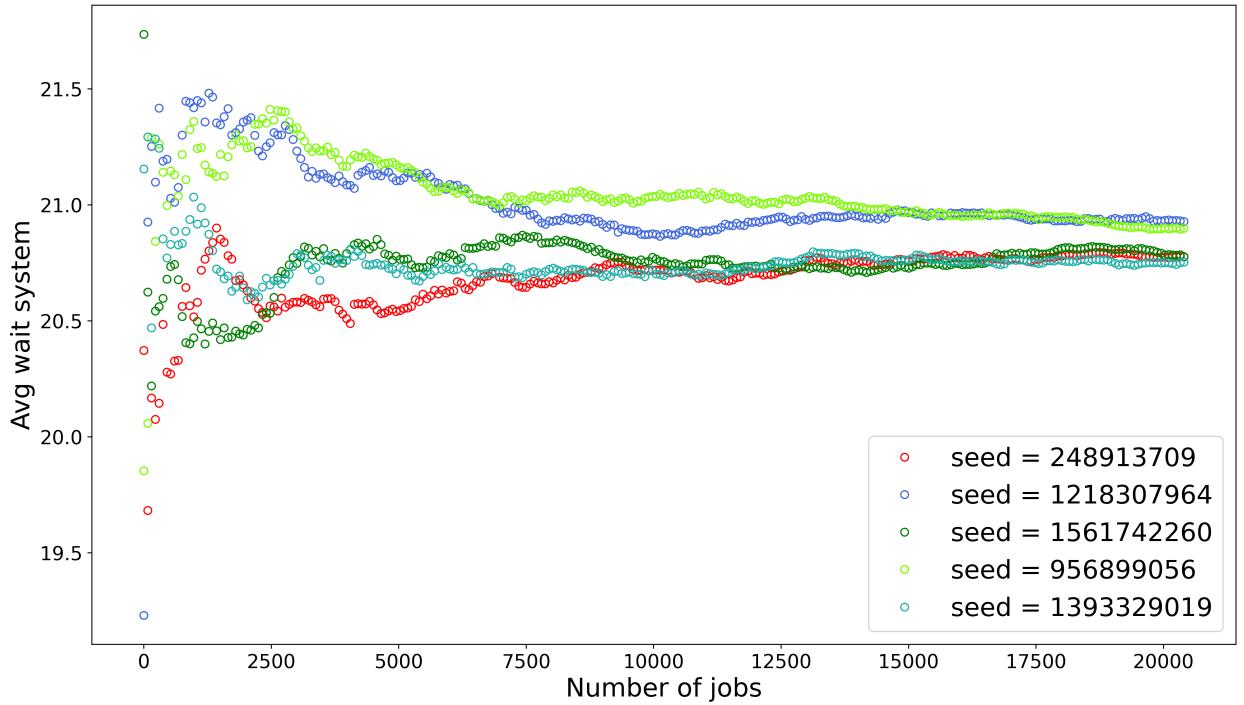
Per quanto riguarda la seconda fascia oraria (12:00-17:00), è possibile osservare i risultati dell'analisi nella successiva figura.



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

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

nella seguente figura.



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

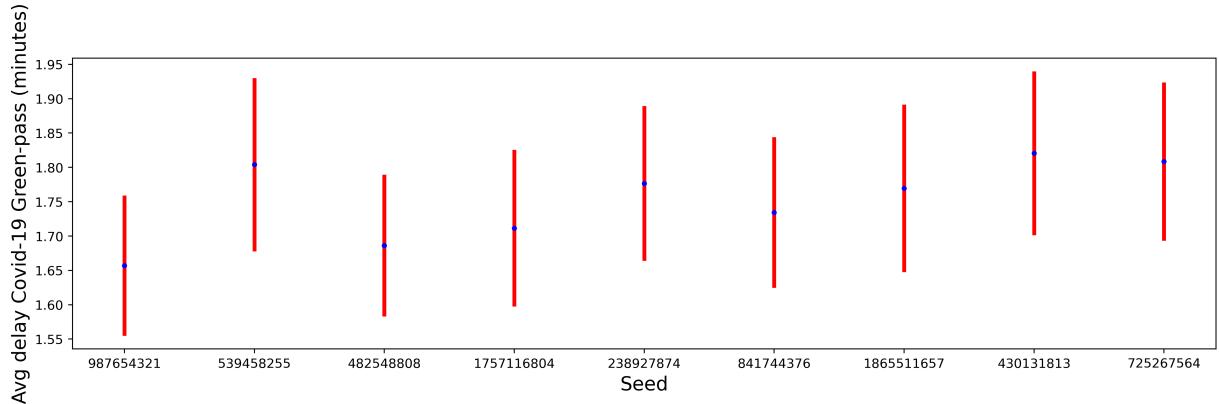
Come si evince dai risultati dell'analisi, il modello avanzato, come previsto, raggiunge la stazionarietà utilizzando la configurazione ottimale ottenuta nel modello base.

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.

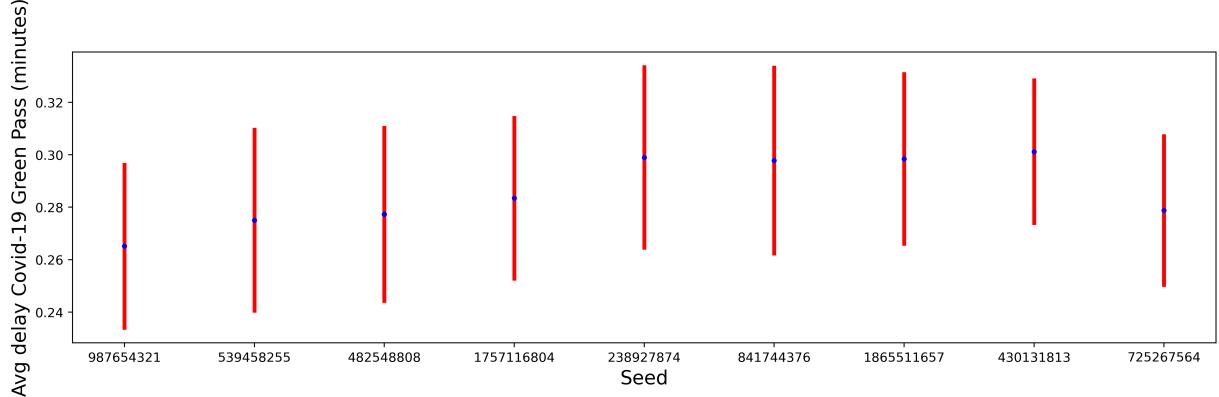
In questa fase sono state effettuate simulazioni con tempi d'osservazione molto lunghi, utilizzando la tecnica batch-means per ottenere delle stime degli indici di prestazione in regime stazionario, evitando la problematica del bias dovuto alle condizioni iniziali del sistema. Cruciale in questa fase 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 per i clienti dotati di green-pass, per ogni seed, per il modello base e il modello avanzato, con un intervallo di confidenza pari al 95%.



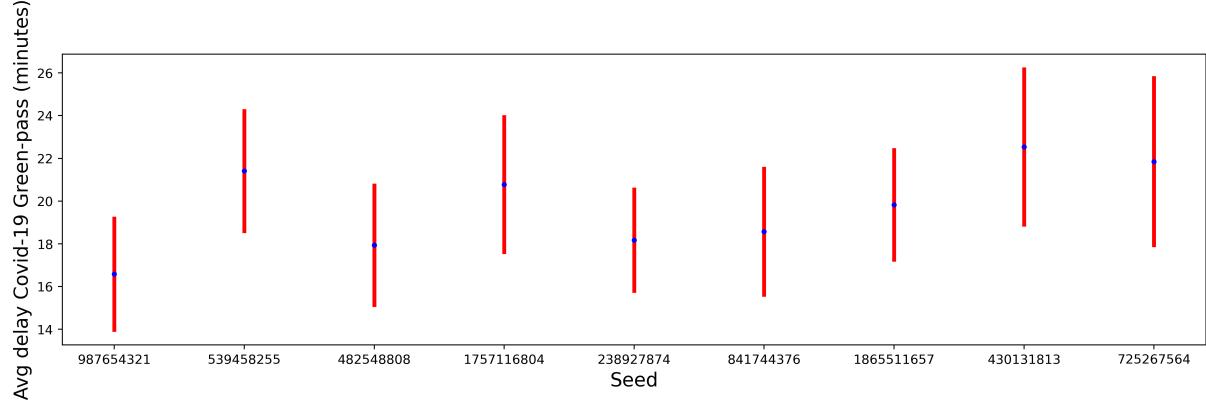
| SEED | MEAN VALUE | STD | CONFIDENCE INTERVAL | CONFIDENCE LEVEL |
|------------|--------------------|----------------------|---------------------------|------------------|
| 987654321 | 1.657018427230146 | 0.3919550835253442 | ± 0.1021072033123597 | 95% |
| 539458255 | 1.8039738942805255 | 0.4840676232809395 | ± 0.1261032023943117 | 95% |
| 482548808 | 1.686246769873599 | 0.3955807785380034 | ± 0.10305172372647185 | 95% |
| 1757116804 | 1.711487892489119 | 0.432773518366248093 | ± 0.11403325858970863 | 95% |
| 238927874 | 1.776710078911322 | 0.4325849592608041 | ± 0.11269158697428673 | 95% |
| 841744376 | 1.7343391918196984 | 0.4208117032744516 | ± 0.10962456656002241 | 95% |
| 1865511657 | 1.7695524880930216 | 0.4677824699371919 | ± 0.12186079928912168 | 95% |
| 482548808 | 1.686246769873599 | 0.3955807785380034 | ± 0.10305172372647185 | 95% |
| 430131813 | 1.8204593383438676 | 0.45782138818595 | ± 0.11926586368977439 | 95% |
| 725267564 | 1.8085013862657082 | 0.44195527177589905 | ± 0.11513262281051986 | 95% |

Figure 11: Base Model, Average delay Covid-19 Green-Pass, 08:00-12:00 & 17:00-22:00, $\lambda_{arrival} = \frac{1}{14} \text{ job/min}$, b=256, k=64



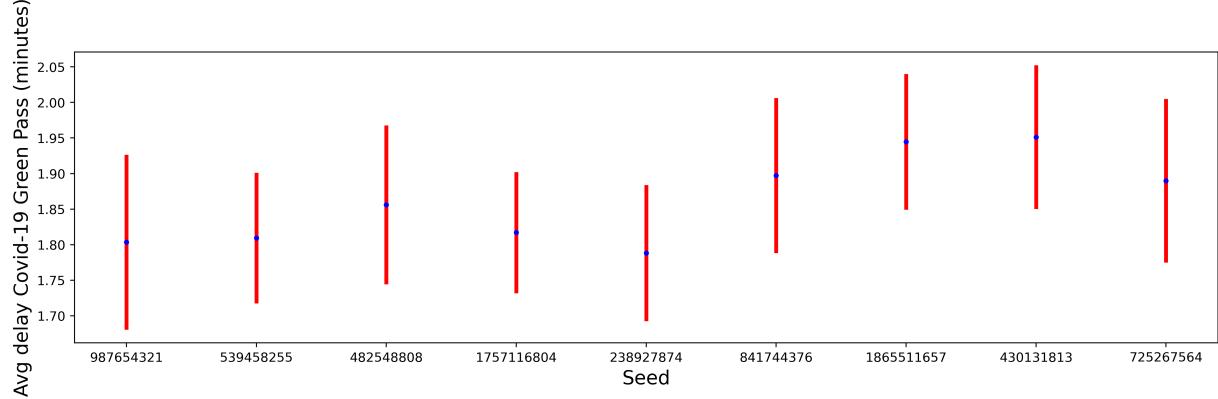
| SEED | MEAN VALUE | STD | CONFIDENCE INTERVAL | CONFIDENCE LEVEL |
|------------|---------------------|----------------------|----------------------------|------------------|
| 987654321 | 0.2651574725493838 | 0.02217031946752235 | ± 0.031826273399511024 | 95% |
| 539458255 | 0.2750603147245544 | 0.03541923538702763 | ± 0.035277714159747425 | 95% |
| 482548808 | 0.27731700699764233 | 0.02950154138068184 | ± 0.0373611102601185 | 95% |
| 1757116804 | 0.28345604955486087 | 0.02051481035056412 | ± 0.03139500100862805 | 95% |
| 238927874 | 0.2898919860265338 | 0.03494951156739 | ± 0.03516718497417174 | 95% |
| 841744376 | 0.29782725559551576 | 0.03883149510258191 | ± 0.03616663309759336 | 95% |
| 1865511657 | 0.2984097446397857 | 0.02708981315955176 | ± 0.033107837955551334 | 95% |
| 482548808 | 0.27731700699764233 | 0.02950154138068184 | ± 0.0373611102601185 | 95% |
| 430131813 | 0.301206712608697 | 0.010733001305826144 | ± 0.027960263625842453 | 95% |
| 725267564 | 0.2787716806661602 | 0.01167149861255289 | ± 0.029091252779453035 | 95% |

Figure 12: Advanced Model, Average delay Covid-19 Green-Pass, 08:00-12:00 & 17:00-22:00, $\lambda_{arrival} = \frac{1}{14} \text{ job/min}$, b=256, k=64



| SEED | MEAN VALUE | STD | CONFIDENCE INTERVAL | CONFIDENCE LEVEL |
|------------|---------------------|--------------------|--------------------------|------------------|
| 987654321 | 16.578580550630903 | 10.340424122234236 | ± 2.6937570975953165 | 95% |
| 539458255 | 21.407738790082796 | 11.123453820573319 | ± 2.8977421355971305 | 95% |
| 482548808 | 17.932942527942615 | 11.060285119938108 | ± 2.8812862210552805 | 95% |
| 1757116804 | 20.7705051224049072 | 12.484255713497168 | ± 3.252241111089127 | 95% |
| 238927874 | 18.168705982472815 | 9.45458010190495 | ± 2.462988166948323 | 95% |
| 841744376 | 18.566191283355685 | 11.664418328554277 | ± 3.0386673980133496 | 95% |
| 1865511657 | 19.825435921071426 | 10.204153425073331 | ± 2.658257571334866 | 95% |
| 482548808 | 17.932942527942615 | 11.060285119938108 | ± 2.8812862210552805 | 95% |
| 430131813 | 22.55319996338946 | 14.296622537077328 | ± 3.724376096729394 | 95% |
| 725267564 | 21.848367622807526 | 15.34816708403181 | ± 3.9983112422623077 | 95% |

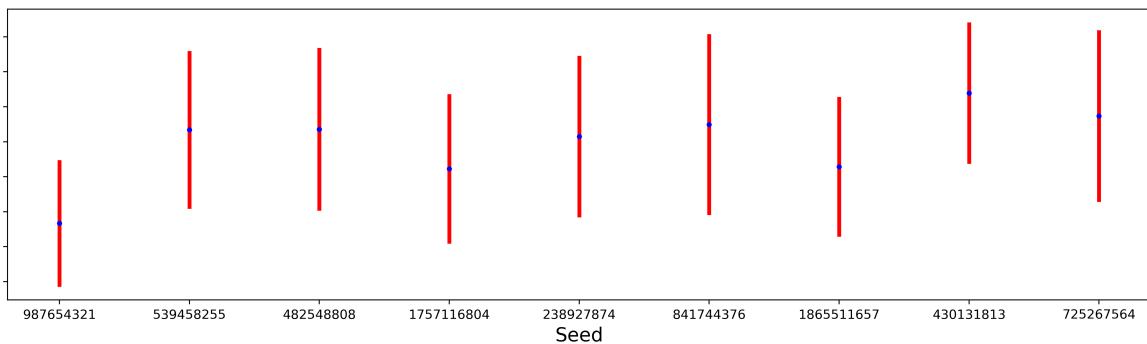
Figure 13: Base Model, Average delay Covid-19 Green-Pass, 12:00-17:00,
 $\lambda_{arrival} = \frac{1}{5} \frac{job}{min}$, b=256, k=64



| SEED | MEAN VALUE | STD | CONFIDENCE INTERVAL | CONFIDENCE LEVEL |
|------------|--------------------|---------------------|---------------------------|------------------|
| 987654321 | 1.8034880175227537 | 0.47198074955617925 | ± 0.129544822355129 | 95% |
| 539458255 | 1.8091756139848176 | 0.3527244438920505 | ± 0.09188733102218964 | 95% |
| 482548808 | 1.8560161814868013 | 0.4280255767992163 | ± 0.1150383406189458 | 95% |
| 1757116804 | 1.81702156618597 | 0.3272186452784387 | ± 0.08524288652166885 | 95% |
| 238927874 | 1.7882593521381844 | 0.3674339241589149 | ± 0.09571925961645342 | 95% |
| 841744376 | 1.8971495526408553 | 0.41836913891639527 | ± 0.10898626044742305 | 95% |
| 1865511657 | 1.9446076702122521 | 0.36659595003384093 | ± 0.09550096114819566 | 95% |
| 482548808 | 1.8560161814868013 | 0.4280255767992163 | ± 0.1150383406189458 | 95% |
| 430131813 | 1.9511742534920518 | 0.3881784978402056 | ± 0.10112337476008756 | 95% |
| 725267564 | 1.8898890165264743 | 0.44189141355215256 | ± 0.11511598727009874 | 95% |

Figure 14: Advanced Model, Average delay Covid-19 Green-Pass, 12:00-17:00,
 $\lambda_{arrival} = \frac{1}{5} \frac{job}{min}$, b=256, k=64

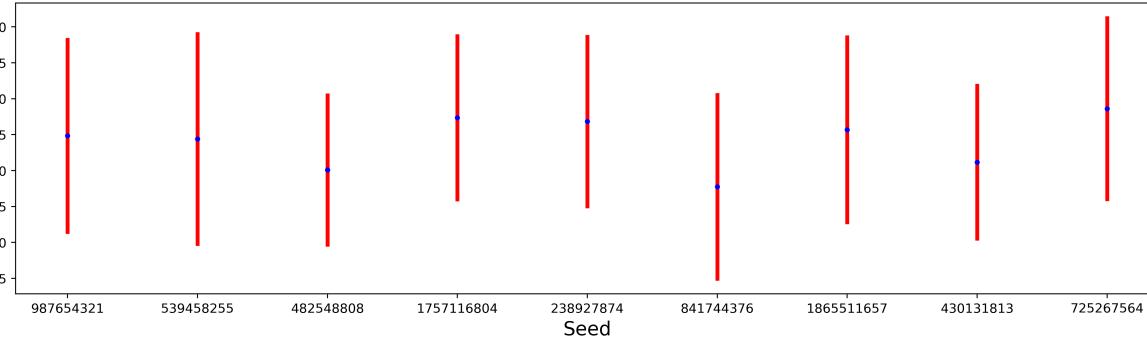
Avg delay Covid-19 Green-pass (minutes)



| SEED | MEAN VALUE | STD | CONFIDENCE INTERVAL | CONFIDENCE LEVEL |
|------------|--------------------|---------------------|----------------------------|------------------|
| 987654321 | 0.513337755964413 | 0.1389768632666143 | ± 0.03620450258137832 | 95% |
| 539458255 | 0.5668592708571076 | 0.17320402378595803 | ± 0.045120931490833446 | 95% |
| 482548808 | 0.5670787667601349 | 0.1784899723734353 | ± 0.04649796026225721 | 95% |
| 1757116804 | 0.544480465812209 | 0.16419449241216016 | ± 0.04277388181498979 | 95% |
| 238927874 | 0.5629645019579377 | 0.17740935193746357 | ± 0.04621645063220767 | 95% |
| 841744376 | 0.5698849315806727 | 0.19848823601376447 | ± 0.05170765611069784 | 95% |
| 1865511657 | 0.545664712269986 | 0.15348755174963197 | ± 0.03998464444307188 | 95% |
| 482548808 | 0.5670787667601349 | 0.1784899723734353 | ± 0.04649796026225721 | 95% |
| 430131813 | 0.5878378446236471 | 0.15512438713645538 | ± 0.04041105218889827 | 95% |
| 725267564 | 0.5746980243411038 | 0.1884268190984402 | ± 0.049086582457708805 | 95% |

Figure 15: Base Model, Average delay Covid-19 Green-Pass, 22:00-08:00,
 $\lambda_{arrival} = \frac{1}{35} \frac{job}{min}$, b=256, k=64

Avg delay Covid-19 Green Pass (minutes)



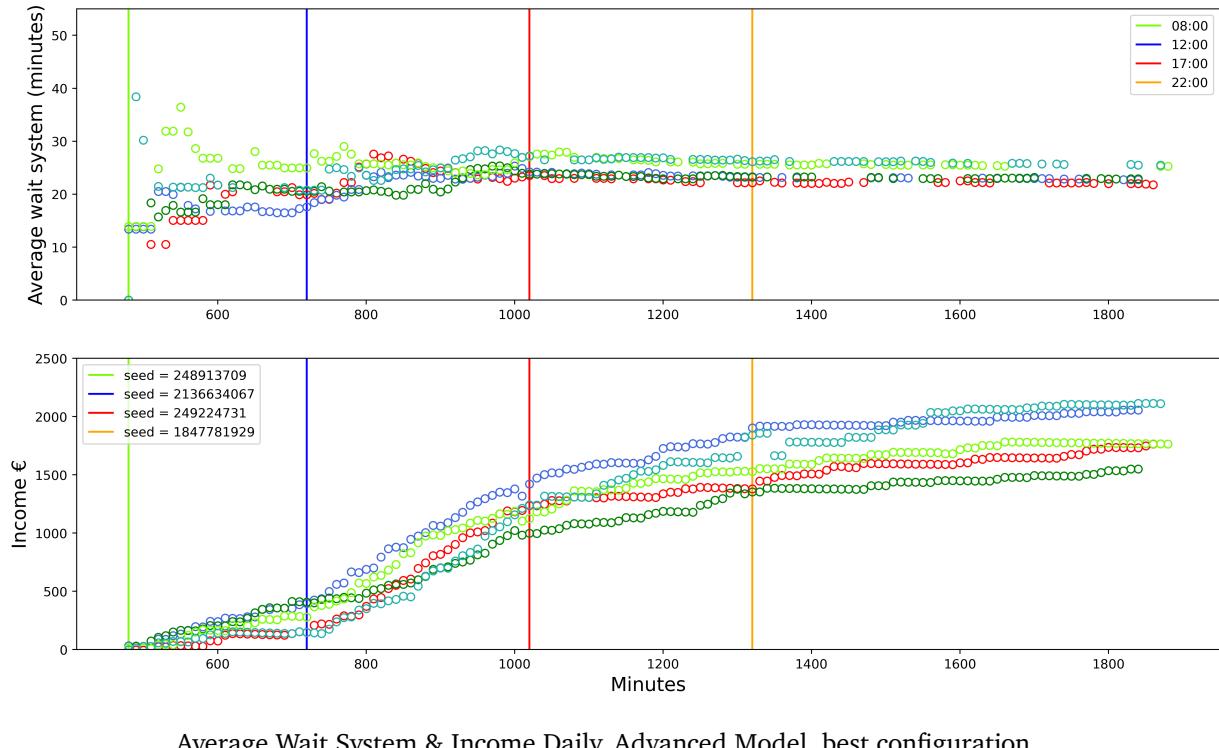
| SEED | MEAN VALUE | STD | CONFIDENCE INTERVAL | CONFIDENCE LEVEL |
|------------|----------------------|----------------------|----------------------------|------------------|
| 987654321 | 0.06484014683463561 | 0.05235408393665761 | ± 0.013638626764759866 | 95% |
| 539458255 | 0.06439144084013518 | 0.05711860261048901 | ± 0.014879819180326439 | 95% |
| 482548808 | 0.06008617668833253 | 0.04087314377753175 | ± 0.010647756789357579 | 95% |
| 1757116804 | 0.06735423961675877 | 0.0446622027822353 | ± 0.011634834603477959 | 95% |
| 238927874 | 0.06682874160319838 | 0.046311543366728325 | ± 0.012064500176203873 | 95% |
| 841744376 | 0.057742149418870105 | 0.050120237816276655 | ± 0.013056693299499662 | 95% |
| 1865511657 | 0.06567901270092515 | 0.050457586744239495 | ± 0.013144575194702789 | 95% |
| 482548808 | 0.06008617668833253 | 0.04087314377753175 | ± 0.010647756789357579 | 95% |
| 430131813 | 0.061163732445865586 | 0.04185184940469543 | ± 0.010902716856552825 | 95% |
| 725267564 | 0.06861758389201747 | 0.04938956064319112 | ± 0.01286634648221411 | 95% |

Figure 16: Advanced Model, Average delay Covid-19 Green-Pass, 22:00-08:00,
 $\lambda_{arrival} = \frac{1}{35} \frac{job}{min}$, b=256, k=64

Come si può osservare dai grafici e dalle tabelle riportate, il tempo di attesa medio nella coda Covid-19, per i clienti che possiedono il Green-Pass, diminuisce sensibilmente utilizzando il modello avanzato, a parità di configurazione e fascia oraria. Tale risultato permette di raggiungere il secondo obiettivo prefissato relativo al modello avanzato.

2.7.3 Analisi guadagno giornaliero

Attraverso questa analisi viene effettuata una simulazione concreta di una tipica giornata lavorativa, ricavata nello studio precedente. Di seguito viene mostrato, per il modello avanzato, il grafico ottenuto per 4 repliche indipendenti:



Average Wait System & Income Daily, Advanced Model, best configuration,
sampling frequency=10 min

Successivamente, è stato svolto un ulteriore studio in cui: i valori ottenuti per 64 repliche vengono mediati incrementalmente, utilizzando l'algoritmo di Welford. Viene inoltre mostrato il confronto tra 3 diversi seed, i quali, come descritto precedentemente, vengono selezionati tramite la funzione `getSeed()`, al fine di garantire indipendenza tra le diverse simulazioni. Il risultato di tale procedura è mostrato nelle seguenti figure.

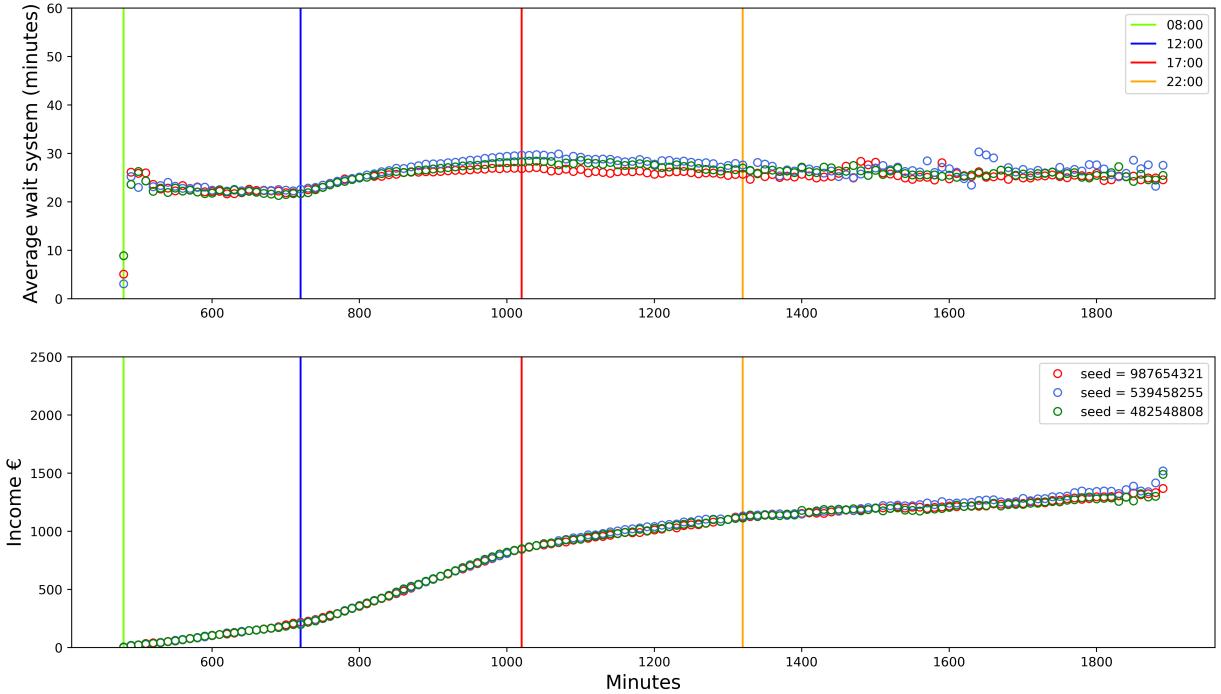


Figure 17: Average Wait System & Income Daily, Base Model, best configuration, sampling frequency=10 min

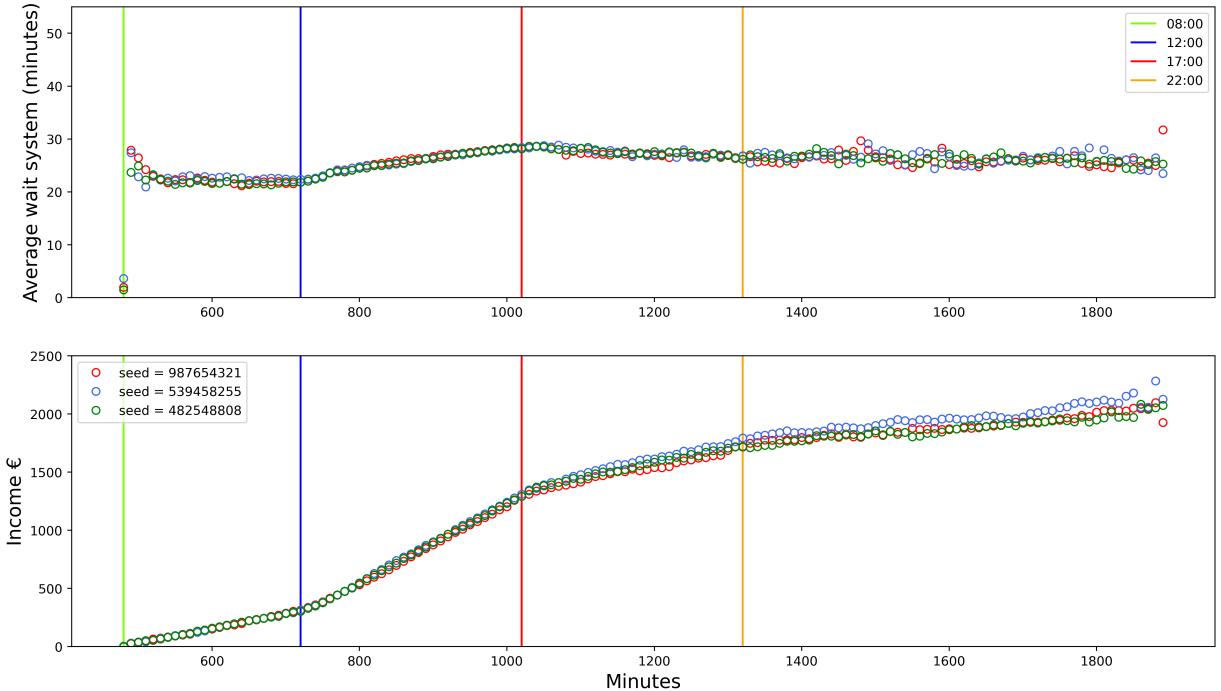


Figure 18: Average Wait System & Income Daily, Advanced Model, best configuration, sampling frequency=10 min

Nelle figure 17 e 18 si può notare il confronto tra il modello base e quello avanzato in corrispondenza della configurazione ottimale. In particolare si osserva un aumento del guadagno, nel caso avanzato, semplicemente introducendo code di priorità, senza incrementare il numero di nodi o i tempi di servizio. Inoltre, si ottiene, mediamente, un tempo di risposta medio del sistema pari o minore a quello del mod-

ello base, per via dell'introduzione della priorità basata sulla size per quanto riguarda il nodo Covid-19, nonostante quest'ultima impatti poco sul sistema stesso. Questa analisi conduce al raggiungimento del primo obiettivo.

3 Conclusioni

L'intento principale dell'analisi è quello di massimizzare i profitti di una sala giochi, individuando la configurazione ottimale del sistema con lo studio del modello base. Successivamente, attraverso lo studio del modello avanzato, si è agito sulla struttura della rete, in modo da evitare ulteriori spese dovute all'introduzione di centri aggiuntivi. Inoltre, viene adottata una politica più fair nei confronti dei clienti provvisti di green-pass; infatti, questi sperimentano un tempo di attesa in coda proporzionale al tempo di servizio.

References

- [1] Lawrence M. Leemis, Stephen K. Park, Discrete-Event Simulation - A first course, Pearson Education Prentice Hall, 2006.
- [2] Stuart Kurtowski, Tracy Camp, Michael Colagrosso - MANET Simulation Studies: The Incredibles.
- [3] <https://github.com/pdsteale/DES-Python>