UNIVERSITÀ DI PISA

Computer Engineering
Performance Evaluation of Computer Systems and Networks

Project Specification

Academic Year: 2022/2023

Biondi Matteo

Cristofani Federico

Guidotti Matteo

Contents

1 INTRODUCTION

1.1 Problem Description

From the group project assignment:

A merry-go-round (MRG) has N seats, and children occupying a seat pay one coin for a ride. A ride takes T units of time, during which the MRG does not take anyone else on board. Children arrive and queue up at the MRG with some interarrival times (see later), and the owner may decide to run a ride even when the MRG is not full. Children may drop out of a queue with an increasing probability after a threshold time Q.
Evaluate at least the earnings per unit of time of the MRG owner depending on the threshold he/she chooses to start a ride on the MRG and the threshold Q. More in detail, at least the following scenarios must be evaluated:
- Exponential distribution of the interarrival times.
- Burst arrivals: interarrival times are still exponential, but the number of children in one arrival is geometrically distributed.

1.2 Objectives

The aim of our work is to evaluate the earnings per unit of time of the MGR owner and optimize it, in addition to evaluate the utilization time of the MGR, in order to detect the costs involved

1.3 Performance Indexes

In order to define a metric of performance of the objectives, we defined the following indexes of performance:

- **Coins Per Unit of Time**: It is the number of earned coins by the MGR owner per unit of time. Let N_c be the number of earned coins during a period of time T_s, the coins per unit of time C_t measured in that period will be computed as:

$$C_t = \frac{N_c}{T_s} \text{ [coin/second]}$$

- **Utilization**: It is defined as the percentage of time in which the MGR is performing a ride within a certain period of time. Let T_r be the time for which the MGR has been performing rides during a period of time T_s, the utilization U can be computed as:

$$U = \frac{T_r}{T_s}$$

2 MODELING

2.1 Introduction

The system is composed by a queue that receives children with a given distribution and by the owner, who will handle it by deciding to start new rides or to wait additional children to come

2.2 General Assumptions

In order to model the system, we made the following assumptions:

- The time that is required to place the children in the MGR seats and to let them exit from it is incorporated into the quantity of time T, that measures the duration of a ride
- The threshold thanks to which the owner decides or not to start a ride when there aren't sufficient children to fill the MGR seats depends only on a fraction of the total number of seats that compose the Merry Go Round
- We assume to carry on the analysis in a contest of a MGR placed in a fair of medium size, so every range of values assigned to our factors has been assigned to consider it realistic and coherent with this assumption
- If a child has lost his/her patience in the same moment in which he/she could enjoy a ride, then he/she will enjoy the ride instead of leaving the queue
- The earnings per unit of time, in our model, is gross of costs
- The increasing probability of leaving the queue is modelled with a quantity that, for every child who enters in the queue, indicates the maximum amount of time that he/she would accept to remain in the queue. The quantity is taken from an exponential distribution and if after such quantity the child is still in the queue, he/she will leave it

2.3 Preliminary Validation

Before the implementation, let's analyse some of our assumptions in order to understand if they are reasonable or not

- We can incorporate the time to sit down the children and to let them exit with the time duration of a ride T because they are three consecutive events. It is also reasonable that the time to sit down and the time to exit are almost constant for every different ride
- It is reasonable that an owner decides or not to start a ride according to the number of children in the queue, given that if he/she starts a ride with a portion of occupied seats that is too small, he/she will have more costs than benefits
- In a realistic world, it's impossible to leave a queue when it's our time to enjoy the thing for which we are in it, so it is reasonable that if the moment to enjoy a ride comes when a child has lost his/her patience, he/she will enjoy it instead of leaving away
- We can see the event "child remains in the queue" as the complementary of the event "child quits from the queue". So if the latter must have an increasing probability, the probability of remaining in the queue has to be decreasing with the increasing of the time. That's the case if we model the time for which a child will remain in the queue with an exponential distribution of probability

2.4 Factors

The factors that can affect the performance of the system that we identified are:

- **Q**: Threshold of time expressed in seconds after which the children that are still in the queue start to lose their patience
- **vFraction**: Minimum number of children that must be in the queue in order to start a ride. It is expressed as a fraction of the total number of seats
- **T**: Time duration of a MGR ride, expressed in seconds
- **N**: Number of seats of which the MGR is composed
- **C**: Number of coins that each child pays to the owner in order to be able to enjoy a ride
- **λ**: Exponential distribution mean inter-arrival time of a child or a group of children
- **P**: Probability of success of a trial assigned to the geometric distribution that regulates the number of children that compose a just arrived group in the burst arrival scenario
- **Δ**: Mean of the exponential distribution used to assign to each child a quantity of time after which he/she will decide to leave the queue

3 IMPLEMENTATION

3.1 Modules

The following modules were defined and implemented in this project:

- **ChildPool**: This module is designed to represent the arrival of new children at the merry-go-round. Arrivals can be of one person at a time or of a group of people.
- **ChildQueue**: This module is designed to represent the queue of the merry-go-round. When children arrive, they join the queue. When the merry-go-round frees up, the children are pulled from the head of the queue under specific conditions.
- **Owner**: This module is designed to represent the merry-go-round owner. The merry-go-round owner is in charge of deciding when to start the merry-go-round and how many people to bring on board.
- **MerryGoRound**: This module represents the physical merry-go-round. Once it is filled (in full or not), it simply must run for a certain time 'T'.

3.2 Network

The modules are connected to each other by connections without delay.

- The **"ChildPool" module** is connected to the module "ChildQueue" by a single connection incoming to the latter and outgoing from the former module. Via this connection, "ChildPool" notifies "ChildQueue" of the arrival of new children in the queue.
- The **"ChildQueue" module** is also connected to the "Owner" module to notify it of a change in the status of the queue if one or more children enter the queue. The "ChildQueue" module can also receive messages from the "Owner" requesting information on the number of children in the queue or requesting for a certain number of children to be removed from the queue so that they can enter the merry-go-round.
- The **"Owner" module** is also connected with the "MerryGoRound" module. In addition to what has just been described, the 'Owner' module receives messages from 'MerryGoRound' informing it that the merry-go-round is free again. In response to this, the "Owner" module sends a message to the "ChildQueue" module about the number of people in the queue. If their number is above a certain threshold, then the module sends a new message to "ChildQueue" to extract a certain number of children from the queue to let them ride on the merry-go-round that is now free. It then sends the module "MerryGoRound" a message to start the service.
- The **"MerryGoRound" module** is connected with the "Owner" module only, from which it receives messages to start the service and to which it sends messages to notify the termination of a ride.

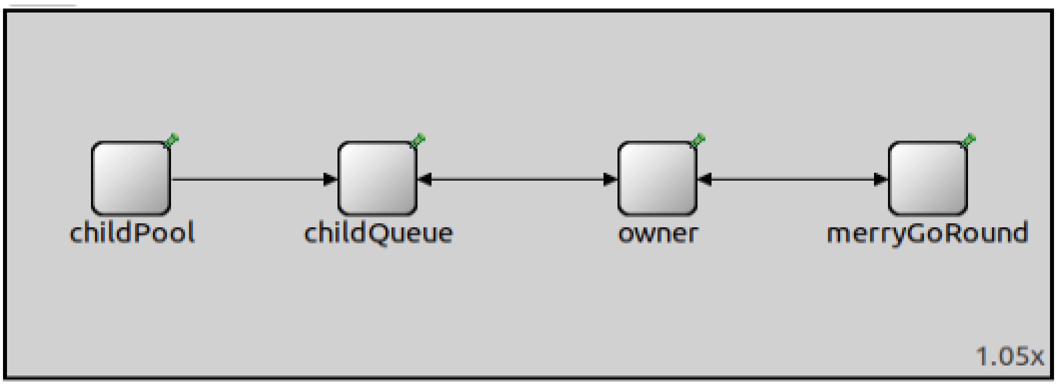

Figure 1: Modules network in OMNeT++ IDE

3.3 Messages

To enable proper communication and data exchange between modules, new message formats have been defined:

- **Name:** ChildArrivalMsg
  Class: ChildArrivalMsg
  Source: ChildPool
  Recipient: ChildQueue
  Purpose: To notify the arrival of a certain number of children in the queue
  Content: The message consists of a vector, '_children', of 'Child' elements representing arriving children. Each 'Child' object has its own '_id' and its maximum waiting time in the queue, '_quitTime'

- **Name:** HowmanyChildren
  Class: QueueHowManyMsg
  Source: ChildQueue
  Recipient: Owner
  Purpose: To notify the number of children in the queue.
  Content: The message consists of a single integer field 'howMany' indicating the number of children in the queue. This message is sent when new children are added to the queue or under a specific request of the "Owner" with a "HowmanyRequest" message

- **Name:** QuitChild
  Class: QueueQuitMsg
  Source: ChildQueue
  Recipient: ChildQueue
  Purpose: To notify the exit of a child from the queue because it has exceeded its maximum waiting time
  Content: The message consists of a single field of type 'child' through which to identify the element to be removed from the child queue.

- **Name:** RemoveFromQueue
  Class: RemoveFromQueueMsg
  Source: Owner
  Recipient: ChildQueue
  Purpose: To notify the queue that the first "howMany" children have been removed from the queue so that they can be served.
  Content: The message consists of a single integer field "howMany" indicating the number of children to be pulled from the queue so they can be served

The messages have different priorities between them to ensure proper interaction. The priorities are of three different levels:

- HIGH_PRIORITY = 0
- MEDIUM_PRIORITY = 1
- LOW_PRIORITY = 2

The priorities associated with each message are shown below:

- "ChildArrivalMsg" = HIGH_PRIORITY
- "CreateChildren" = HIGH_PRIORITY
- "HowmanyChildren" = HIGH_PRIORITY
- "RemoveFromQueue" = HIGH_PRIORITY
- "StartMgr" = HIGH_PRIORITY
- "HowmanyRequest" = MEDIUM_PRIORITY
- "MgrIsFree" = MEDIUM_PRIORITY
- "RideFinished" = MEDIUM_PRIORITY
- "QuitChild" = LOW_PRIORITY

3.4 Modules’ Behaviour

3.4.1 Behaviour of the ChildPool module in implementation

i. After a certain interarrival time has elapsed, it creates a new arrival consisting of one or more children (geometric distribution with mean equal to 1/“P”), each with its own "quitTime" (dependent on "Q" and a value extracted from an exponential distribution with mean " **Δ** ").
ii. Create and then send the message to "ChildQueue" containing the newly arrived children
iii. Schedules the next arrival at a future time with exponential distribution with mean defined in the factor "**λ**".

3.4.2 Behaviour of the ChildQueue module in implementation

The module receives a message:

i. **IF** the message is a "ChildArrivalMsg": It inserts new arrivals to the queue, schedules their possible future queue exit events (due to the expiry of their associated "quitTime"), and informs "Owner" of the number of people in the queue
ii. **OTHERWISE IF** the message received is of type "HowmanyRequest", it sends "Owner" a "HowmanyChildren" message containing the number of people in the queue
iii. **OTHERWISE IF** the message received is of type "RemoveFromQueue", remove as many children as requested (by "Owner") from the queue so that they can be served
iv. **OTHERWISE IF** the message received is of type "QuitChild", remove if still present the child associated with that message from the queue (its maximum waiting time has expired)

3.4.3 Behaviour of the Owner module in implementation

The module receives a message:

i. **IF** the received message is of type "MgrIsFree", it sends a message to "ChildQueue" of type "HowmanyRequest" to find out the number of children in the queue (answer in step 2).
ii. **OTHERWISE IF** the message received is of type "HowmanyChildren",
    a. **IF** the carousel is busy **OR** there is not enough time left for a ride, then ignore
    b. **OTHERWISE** identifies the maximum number of people who can ride the merry-go-round, sends a message to "ChildQueue" of type "RemoveFromQueue" to remove that number of children from the queue, and sends to "MerryGoRound" a message "StartMgr" to start the merry-go-round

3.4.4 Behaviour of the MerryGoRound module in implementation

The module receives a message:

i. **IF** the received message is of type "StartMgr", it schedules an event after "T" time to stop the merry-go-round
ii. **OTHERWISE IF** the received message is of type "RideFinished", it sends a message of type "MgrIsFree" to "Owner".

3.5 Statistics

"ChildPool" module statistics:

- "**interArrivalStat**": Statistics to collect vector-type and scalar-type data related to the interarrival time between one arrival and the next one of children.
- "**bulkStat**": Statistic to collect vector-type and scalar-type data relating to the number of children in each new arrival

"ChildQueue" module statistics:

- "**queueTotalStat**": Statistic that allows tracking the arrival of each child in the queue. The statistics collected are vector-type and scalar-type data, concerning the total number of arrivals.
- "**queueQuitStat**": Statistics to keep track of the exit from the queue of each child who has not yet been served but for whom the maximum waiting period has expired. The statistics collected are of a vector-type and scalar-type data relating to the number of children leaving the queue without being served
- "**queueServedStat**": Statistics to keep track of children being served. The statistics collected are vector-type and scalar-type data related to the number of children enjoy a ride
- "**queueLengthStat**": Statistics to keep track of the queue size trend. For each change to the queue size (due to arrival, exit from the queue, or removal to provide service) a signal is emitted. The statistics collected are both vector-type and scalar-type data related to the time average of the queue size.

"Owner" module statistics:

- "**coinsPerUnitOfTime**": As the name suggests, this statistic tracks the number of coins obtained per unit of time. For each new ride, a signal is emitted containing the number of coins earned. This information is each time added to the previously obtained coins and the sum is divided by the current simulation duration. The statistics collected are both vector-type (to understand the trend) and scalar-type data relative to the last calculated value.
- "**avgPeoplePerRide**": Statistics to keep an average of the people who get on the ride each time it starts. The statistics collected are scalar-type data and concern the number of rides completed and the average number of people.

“MerryGoRound” module statistics:

- “**MGR_Utilization**”: Statistics to determine the degree of utilization of the merry-go-round itself. In particular, every time the merry-go-round stops at the end of one of its rides, a signal with a value of T is emitted. This value is then added to others previously emitted and then divided by the duration of the simulation at the instant of emission. The statistics collected are both vector-type (to understand the trend) and scalar-type data relative to the last calculated value.

To avoid an unmanageable amount of data, for each configuration executed in our simulations, from the above-mentioned statistics, we select only the statistics needed for analysis.

4 VERIFICATION

Once the simulator has been implemented, we performed some verification tests to assess if the implementation respects the model that we have defined.

These tests consisted in running simulations under some specific conditions for which we knew in advance the results to expect. This allowed us to compare results with the ones we expected in order to spot some possible problems in case of unexpected behaviour.

4.1 Deterministic test

We started testing the simulator removing the aleatory component of our model, given by the random inter-arrivals times of children in the systems. To do so we have replaced the first module, *childPool*, with a full deterministic one, named *dummyChildPool* in which the children arrive in the system at a constant rate. Moreover, we set the factor *Q* in such a way that we do not have any quit from the queue to simplify the calculus and we have performed the test varying the value of the factor *vFraction*

We have manually computed the number of children served in during the simulation and confirmed that the results were the same.

Note that all the other tests have been carried out using the original modules of the simulator, to guarantee the verification of the entire system.

The configuration is the following:

- **Simulation duration** = 3720 s
- **Q** = 3720 [s] (no quit from queue)
- **Δ** = 300 [s] (no impact because Q)
- **vFraction** = [0.5, 1]
- **T** = 120 [s]
- **N** = 24
- **C** = 1
- **λ** = 10 [s]
- **P** = 1

Given these factors the results must be:

Statistics	vFraction = 1	vFraction = 0.5
# Children arrivals	372	372
# Served children	360	360
# Quit children	0	0
# Rides	15	30
Utilization	0.483871	0.967742
Coins per unit of time	0.096774	0.096774

Note that 12 children are neither served nor exit from queue because end of simulation before (or at the same time) reaching the threshold needed to start a ride.

4.2 Degeneracy test

In the following we have tested the behaviour of the simulator in borderline conditions, setting one (or two if needed) factor at a time to 0.

For each configuration the other factors have been set to a realistic value to better understand the effect of the factor under analysis.

- **Q, Δ ->** As expected no child has been served because children in queue have no patience to wait for the arrival of other children until reach of the minimum number necessary to start a ride. Children join queue and at the same time decide to leave it.
- **N** -> The merry go round hasn't any seat, so is impossible to start a ride with children on board. All children join queue and leave it after some time.
- **λ** -> Inter-arrival times are all scheduled at simulation time 0, so the simulation can't go on and continues to infinite.
- **T** -> Merry go round has a ride duration set to 0, the service time is infinite. However, in the base configuration some children quit from queue because the ride can start only when children threshold has been reached and this requires some time. Moreover, the utilization stays to 0 because each ride has 0-time duration.

4.3 Consistency test

The inter-arrivals of children, as described above, is characterized by two factors: **λ** and **p**. It's possible to estimate the total number of arrivals with the following formula:

$$\#Children \approx \frac{SimulationDuration}{\lambda} \cdot \frac{1}{p}$$

We have defined two configurations, each long 700 hours:

1. **λ**=10 [s], **p**=1
2. **λ**=20 [s], **p**=1/2

The test proved that the total amount of children arrived in the system is approximately the same number in both configurations.

	Mean Children Arrived	**0.95 Confidence Interval**
λ=10, p=1	252014	[251 874, 252 154]
λ=20, p=1/2	252 215	[251 879, 252 552]

Note that above numbers are rounded to integer.

4.4 Continuity test

We have ensured the continuity of the output of our simulator slightly changing one factor at a time, evaluating also that the results agreed with the expected ones. Similarly at the consistency test the factors that are not under test have been set to realistic values in each configuration.

- **Δ** -> Increasing this value means increasing the patience of children waiting for a ride, so the mean number of quit from queue decrease.
- **λ** -> Higher values of mean inter-arrivals times results in less frequent arrives in the system, reducing the total number of children.
- **N** -> Increasing the number of seats available, keeping the *vFraction* factor constant, is pretty like modifying the latter factor keeping constant the number of seats. At this point we hadn't enough information to evaluate in advance the trend of served children varying the factor, however we could assess that the output results slightly change for each different value of the factor.
- **P** -> If the mean number of children in a group increase and the frequency of arrivals keep constant, then the total amount of children increases in accordance with the increase of group size.

- **T** -> Again we hadn't enough knowledge of the system to understand the importance of T factor, however we notice that slightly changing values of T didn't impact on the number of rides, so higher values increase the utilization of the Merry Go round because each ride has a longer duration.

4.5 M/D/1 model

Our complete system can't be easily fitted in a well-known model given by theory, because we have bulk arrivals, bulk services depending by job in queue, determinist service time and quit from queues after a random time interval. However, removing some features the system can be perfectly fitted in a M/D/1 queue:

- **N** = 1, **vFraction**=1 -> Removal of bulk services
- **Q** = simulation duration -> Removal of quit from queue
- **P** = 1 -> Removal of bulk arrivals

Other factors were set to:

- **λ** = [80, 100, 120] [s]
- **Δ** = 300 [s] (no impact because Q)
- **T** = 60 [s]
- **C** = 1
- **Simulation duration** = 3000 hours
- **Warmup time** = 150 hours

The mean performance indexes can be manually computed leveraging Pollaczek–Khinchine formula, valid for M/G/1 queues, therefore M/D/1 included.

For simplicity we report just the results for **λ** = 100 [s].

	Formula	Theoretical value	Simulation value	0.95 Confidence Interval
Utilization	$\frac{\lambda}{\mu}$	0.6	0.5997	[0.5991, 0.6004]
Throughput	$\rho \cdot \mu$	0.01	0.0099	[0.009984, 0.010006]
Mean jobs in the queue	$\frac{1}{2} \cdot (\frac{\rho^2}{1-\rho})$	0.45	0.4504	[0.4479, 0.4529]
Mean waiting time	$\frac{\rho}{2\mu \cdot (1-\rho)}$	45	45.0574	[44.8309 45.2838]

Note that $\rho = \frac{\lambda}{\mu}$ represents the utilization of the server i.e., the merry-go round.

5 SIMULATION EXPERIMENTS

5.1 Study of interarrival time and scenarios identification

To identify the extremes of the interarrival times, the utilization has been changed varying the values of interarrival time for different configuration of the ride duration and vFraction factors. Utilization has been used to make a comparison looking at the load of the system. Three different areas of interest have been identified.

a) Low load scenario: Mean interarrival time ϵ [80; 160] [s]
b) Medium load scenario: Mean interarrival time ϵ [40; 80) [s]
c) High load scenario: Mean interarrival time ϵ (0;40) [s]

For each of them, one representative lambda has been chosen, about in the middle of the range, as the interarrival time of that scenario in case of single arrival (just one child per arrival). Then a different lambda has been chosen for each scenario as representative of the interarrival time in case of bulk arrival (one child or more per arrival). In the following the table that summarizes the chosen values.

Scenario	**Low Load**		**Medium Load**		**High Load**	
Arrival type	Single arrival	Bulk arrival	Single arrival	Bulk arrival	Single arrival	Bulk arrival
λ	120 sec	480 sec	60 sec	240 sec	20 sec	80 sec
P	1	¼	1	¼	1	¼

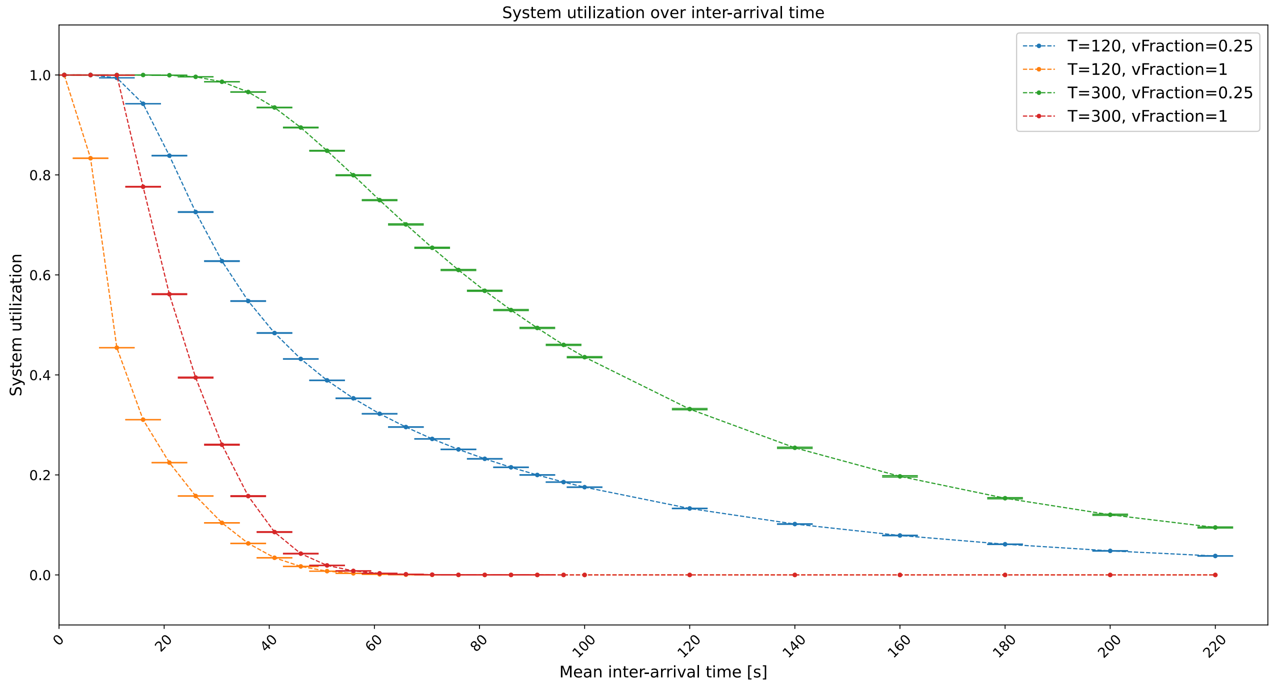

Figure 2: The Utilization varying interarrival time, vFraction and duration of a ride

The confidence intervals associated with the points in the graph are computed with 95% of confidence. These confidence intervals are very small and appear to be as strict horizontal lines. This is due to the fact that the sample standard deviations are small.

5.2 Calibration of Warm-Up period

To properly calibrate the warm-up period of our simulation, many simulations were performed. Looking at the achieved results, we noticed that utilization was the index that impact heavily in the warm-up time. In the following figure, the worst case is shown.

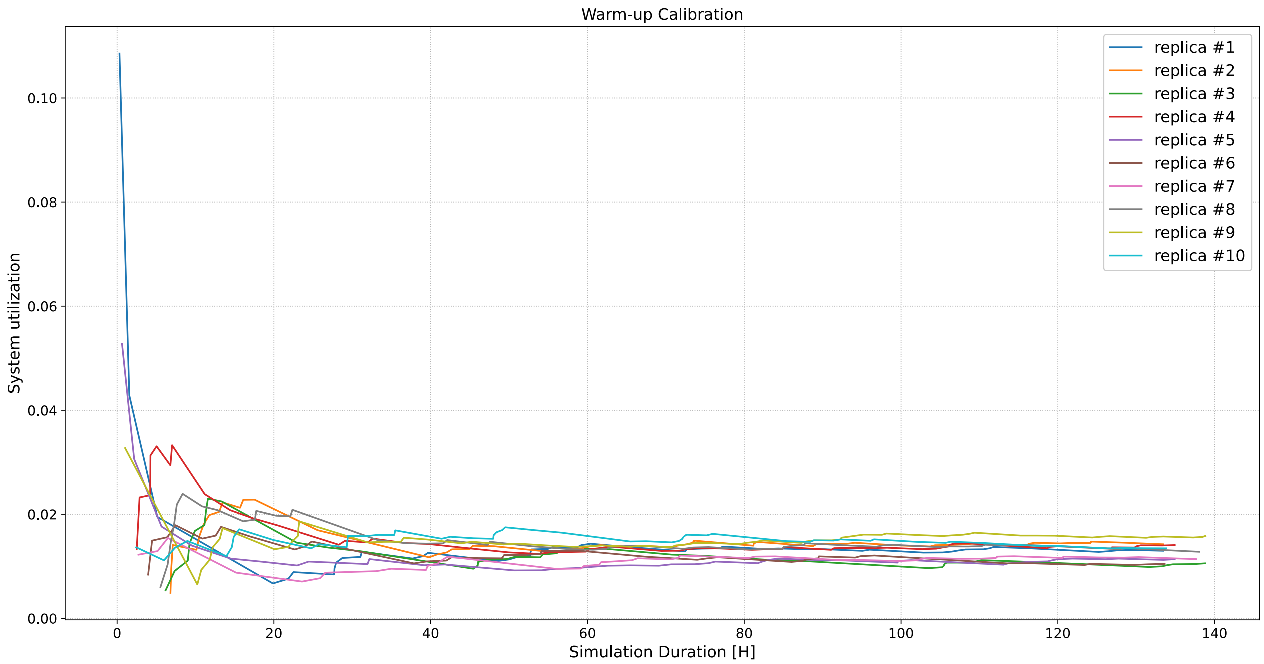

Figure 3: Change in utilization as the simulation evolves, zoomed in approximately in the range [0; 140] hours.

The configuration shown is as follows:

- **vFraction:** 1
- **λ:** 480 [s]
- **N:** 24
- **T:** 120 [s]
- **P:** 1/4
- **Q:** 900 [s]
- **Δ:** 300 [s]

After analysing the above, a warm-up period of 15 hours was chosen.

5.3 Calibration of Simulation Time

To properly calibrate the simulation duration, a trade-off was made between the memory for storing data and the time consumption for performing the simulations but letting our KPI to converge. Obviously, the duration must be greater than the warm-up duration. All things considered, a simulation-duration of 650 hours was chosen.

5.4 Factorial Analysis

To analyse the contribution of the factors on our KPI (Coins per unit of time, Utilization), we perform a $2^k r$ analysis with r = 10 and k = 3 (so we perform $10 \cdot 2^3 = 80$ experiments). We consider the following factors, with the same values in all the considered scenarios and type of arrival:

- N: 24
- C: 1
- Δ: 300 [s]

The values of the factors λ and P are different for the considered scenarios and types of arrivals, but still constant and not part of the factorial analysis. The values of these two factors are the same as the shown in the table at the beginning of the "Simulation Experiments" chapter. The factors taken into consideration in this factorial analysis are the following:

- T: {120, 300} [s]
- Q: {180, 900} [s]
- vFraction:
  - {3/24, 15/24} for low load scenarios
  - {6/24, 1} for both medium and high load scenarios

The first step is to check the hypothesis. We must control, for each scenario and arrival types, that the residuals are normal and independent and that its standard deviation is constant (a.k.a. homoskedasticity). For what concerns the normal hypothesis, each QQ Plot shows a linear tendency, so the hypothesis is verified. It's possible to see (Figure 3) an example of QQ plot.

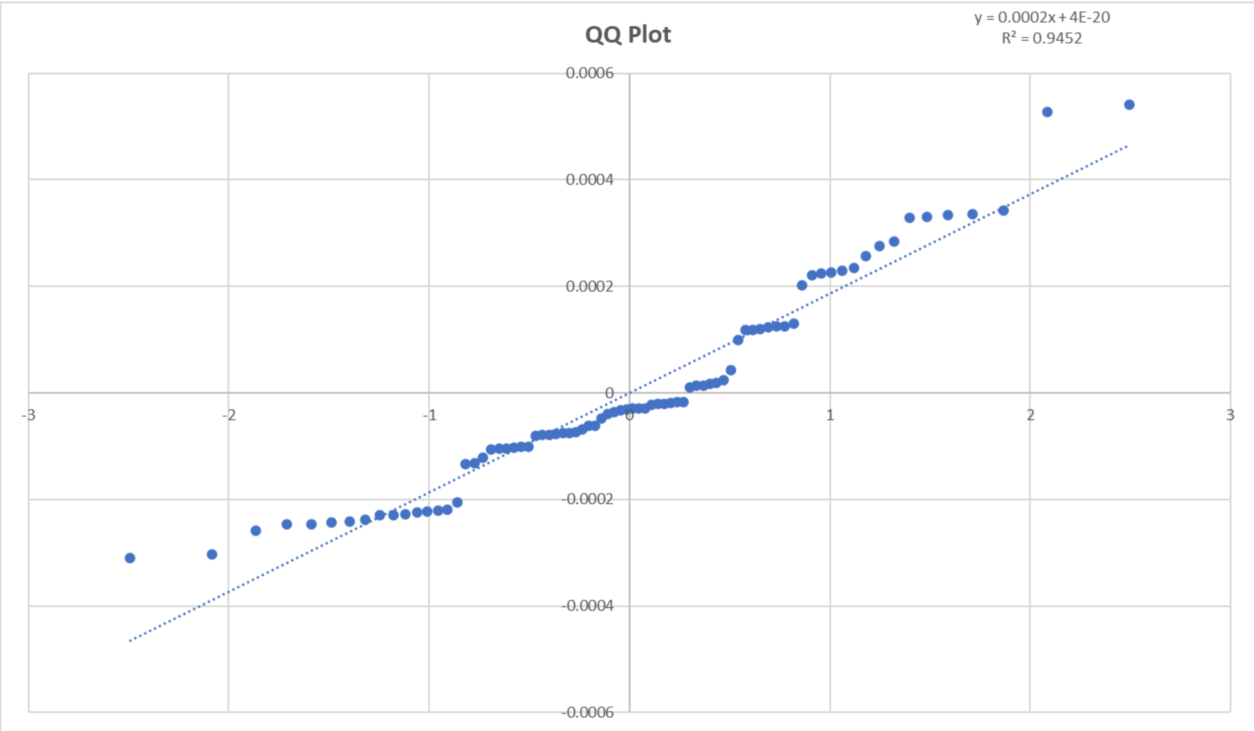

Figure 4: QQ plot for high load single arrival configuration. Visible linear trend.

For independence of the residuals, we have a "Residuals vs Experiment Number" plot for each scenario and arrival types (Figure 4 is an example). In these plots we can't see any significant trend, so the hypothesis is respected.

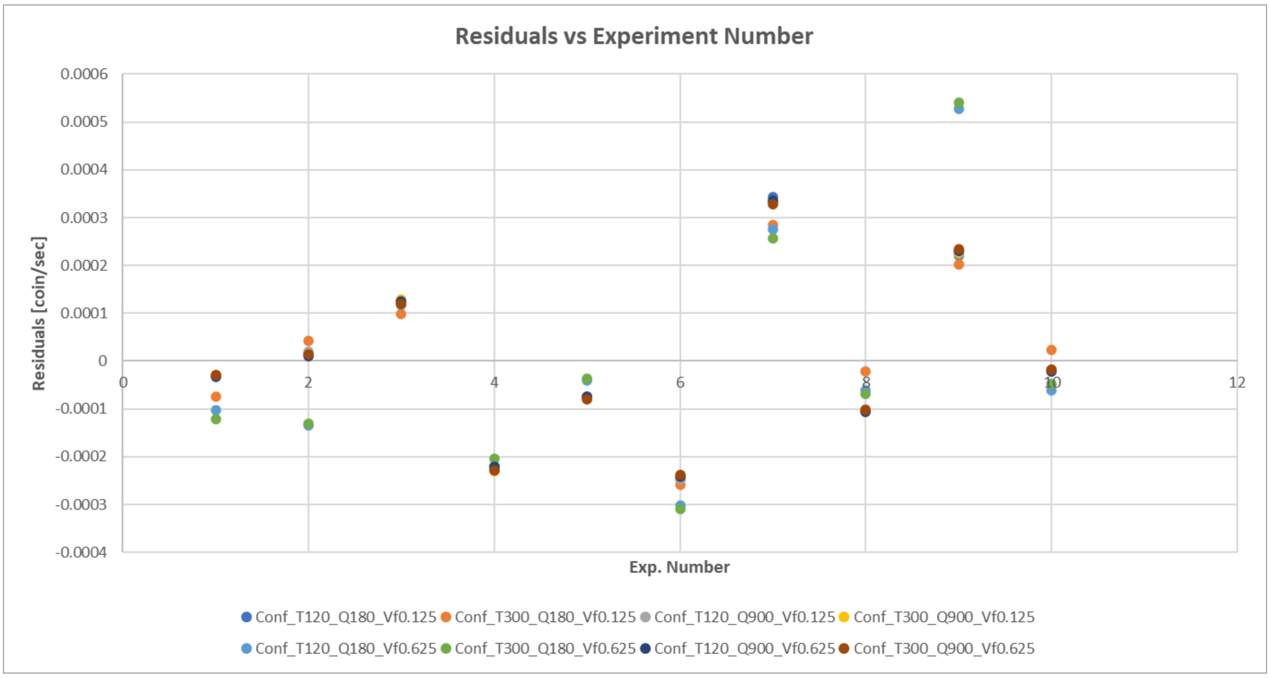

Figure 5: Residuals at different experiment numbers for high load single arrival configuration. No visible accumulation trend.

For the homoskedasticity, we have a “Residuals vs Average Predicted Response” plot for each scenario and arrival types (Figure 5 is an example). In these plots we can’t notice any significant trend or otherwise, if a trend can be identified, the errors (y axis) are at least one order of magnitude below the average predicted response (x axis) and so we can ignore trend. So the homoskedasticity hypothesis is respected.

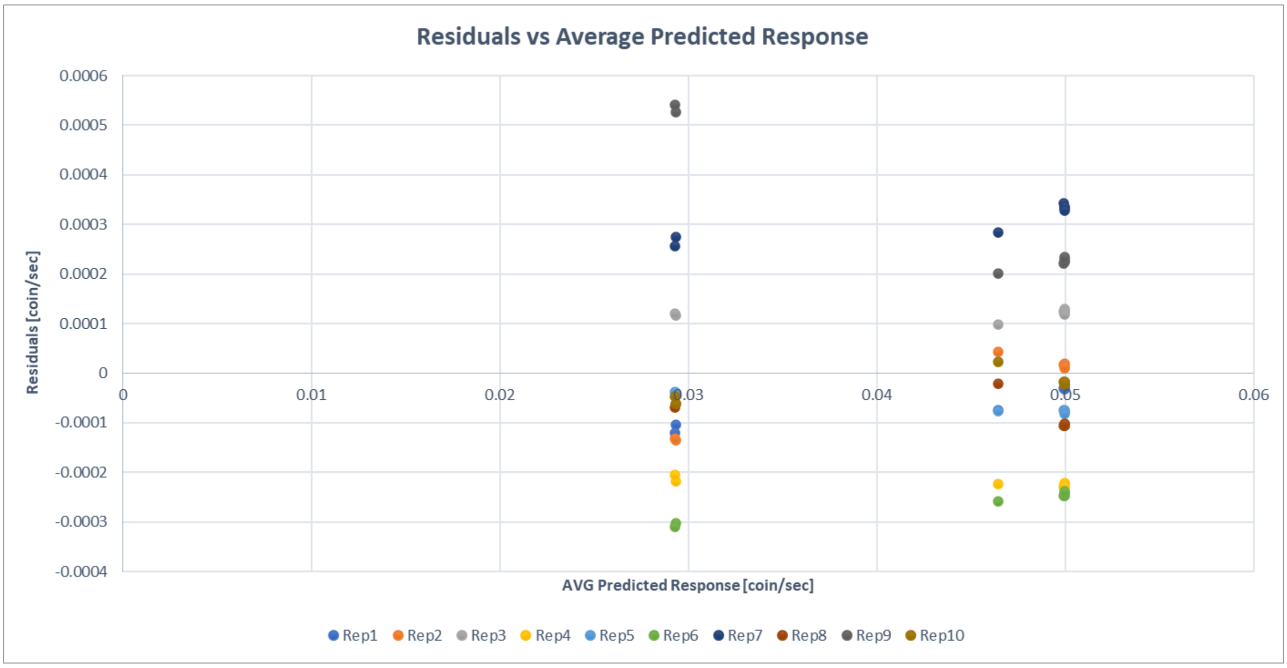

Figure 6: Residuals for different average predicted response for high load single arrival configuration. No visible trend.

After performing all these steps for any of the following cases, in the following we concentrate only on the results of the $2^k r$ analysis.

5.4.1 Low load factorial analysis

Coins per unit of time – Single Arrival

Here a table that resume all the analysed factors and their fraction of variation. The factors that are the most relevant one for the "coinsPerUnitOfTime" metric are the one coloured in red.

	Frac. Variation(%)	qi Confidence Interval (95%)	
		Lower Extreme	Upper Extreme
T	0.004	-3.803E-05	-6.649E-06
Q	7.077	8.929E-04	9.242E-04
Vfraction	91.304	-3.279E-03	-3.248E-03
T_Q	0.004	6.780E-06	3.816E-05
T_Vfraction	0.004	6.649E-06	3.803E-05
Q_Vfraction	1.564	4.114E-04	4.428E-04
T_Q_Vfraction	0.004	-3.816E-05	-6.780E-06

Utilization – Single Arrival

Here a table that resume all the analysed factors and their fraction of variation. The factors that are the most relevant one for the "Utilization" metric are the one coloured in red.

	Frac. Variation(%)	qi Confidence Interval (95%)	
		Lower Extreme	Upper Extreme
T	12.643	9.850E-02	9.975E-02
Q	0.706	2.279E-02	2.404E-02
Vfraction	75.820	-2.434E-01	-2.421E-01
T_Q	0.115	8.811E-03	1.006E-02
T_Vfraction	10.676	-9.171E-02	-9.046E-02
Q_Vfraction	0.029	-5.342E-03	-4.091E-03
T_Q_Vfraction	0.003	-2.048E-03	-7.967E-04

Coins per unit of time – Bulk Arrival

Here a table that resume all the analysed factors and their fraction of variation. The factors that are the most relevant one for the "coinsPerUnitOfTime" metric are the one coloured in red.

	Frac. Variation(%)	qi Confidence Interval (95%)	
		Lower Extreme	Upper Extreme
T	0.013	-7.088E-05	2.028E-05
Q	13.368	7.729E-04	8.640E-04
Vfraction	79.260	-2.038E-03	-1.947E-03
T_Q	0.014	-1.878E-05	7.238E-05
T_Vfraction	0.010	-2.300E-05	6.815E-05
Q_Vfraction	6.575	5.283E-04	6.195E-04
T_Q_Vfraction	0.011	-6.928E-05	2.187E-05

Utilization – Bulk Arrival

Here a table that resume all the analysed factors and their fraction of variation. The factors that are the most relevant one for the "Utilization" metric are the one coloured in red.

	Frac. Variation(%)	qi Confidence Interval (95%)	
		Lower Extreme	Upper Extreme
T	22.464	6.316E-02	6.530E-02
Q	0.781	1.091E-02	1.305E-02
Vfraction	65.872	-1.111E-01	-1.089E-01
T_Q	0.155	4.262E-03	6.399E-03
T_Vfraction	10.483	-4.495E-02	-4.281E-02
Q_Vfraction	0.115	3.518E-03	5.654E-03
T_Q_Vfraction	0.017	7.171E-04	2.854E-03

5.4.2 Medium load factorial analysis

Coins per unit of time – Single Arrival

Here a table that resume all the analysed factors and their fraction of variation. The factors that are the most relevant one for the "coinsPerUnitOfTime" metric are the one coloured in red.

	Frac. Variation(%)	qi Confidence Interval (95%)	
		Lower Extreme	Upper Extreme
T	0.002	-6.050E-05	2.732E-06
Q	17.842	2.676E-03	2.739E-03
Vfraction	77.104	-5.660E-03	-5.597E-03
T_Q	0.002	-3.465E-06	5.976E-05
T_Vfraction	0.002	-2.994E-06	6.023E-05
Q_Vfraction	5.002	1.402E-03	1.465E-03
T_Q_Vfraction	0.002	-6.003E-05	3.202E-06

Utilization – Single Arrival

Here a table that resume all the analysed factors and their fraction of variation. The factors that are the most relevant one for the "Utilization" metric are the one coloured in red.

	Frac. Variation(%)	qi Confidence Interval (95%)	
		Lower Extreme	Upper Extreme
T	15.113	1.086E-01	1.099E-01
Q	1.830	3.734E-02	3.867E-02
Vfraction	71.619	-2.384E-01	-2.371E-01
T_Q	0.304	1.482E-02	1.616E-02
T_Vfraction	11.119	-9.435E-02	-9.302E-02
Q_Vfraction	0.004	-2.438E-03	-1.104E-03
T_Q_Vfraction	0.000	-6.279E-04	7.053E-04

Coins per unit of time – Bulk Arrival

Here a table that resume all the analysed factors and their fraction of variation. The factors that are the most relevant one for the “coinsPerUnitOfTime” metric are the one coloured in red.

	Frac. Variation(%)	qi Confidence Interval (95%)	
		Lower Extreme	Upper Extreme
T	0.011	-1.089E-04	4.918E-06
Q	22.227	2.313E-03	2.426E-03
Vfraction	69.485	-4.246E-03	-4.133E-03
T_Q	0.011	-3.147E-06	1.106E-04
T_Vfraction	0.007	-1.384E-05	9.995E-05
Q_Vfraction	8.018	1.366E-03	1.480E-03
T_Q_Vfraction	0.009	-1.033E-04	1.050E-05

Utilization – Bulk Arrival

Here a table that resume all the analysed factors and their fraction of variation. The factors that are the most relevant one for the “Utilization” metric are the one coloured in red.

	Frac. Variation(%)	qi Confidence Interval (95%)	
		Lower Extreme	Upper Extreme
T	22.1788	8.074E-02	8.264E-02
Q	2.3234	2.549E-02	2.739E-02
Vfraction	64.9830	-1.408E-01	-1.389E-01
T_Q	0.4327	1.046E-02	1.236E-02
T_Vfraction	9.8479	-5.539E-02	-5.349E-02
Q_Vfraction	0.1524	5.823E-03	7.723E-03
T_Q_Vfraction	0.0275	1.926E-03	3.826E-03

5.4.3 High load factorial analysis

Coins per unit of time – Single Arrival

Here a table that resume all the analysed factors and their fraction of variation. The factors that are the most relevant one for the “coinsPerUnitOfTime” metric are the one coloured in red.

	Frac. Variation(%)	qi Confidence Interval (95%)	
		Lower Extreme	Upper Extreme
T	0.251	-4.833E-04	-3.939E-04
Q	41.059	5.570E-03	5.660E-03
Vfraction	28.955	-4.760E-03	-4.670E-03
T_Q	0.248	3.921E-04	4.815E-04
T_Vfraction	0.245	3.894E-04	4.788E-04
Q_Vfraction	28.948	4.670E-03	4.759E-03
T_Q_Vfraction	0.246	-4.791E-04	-3.897E-04

Utilization – Single Arrival

Here a table that resume all the analysed factors and their fraction of variation. The factors that are the most relevant one for the "Utilization" metric are the one coloured in red.

	Frac. Variation(%)	qi Confidence Interval (95%)	
		Lower Extreme	Upper Extreme
T	11.508	1.087E-01	1.094E-01
Q	1.992	4.500E-02	4.573E-02
Vfraction	82.294	-2.919E-01	-2.912E-01
T_Q	0.360	1.893E-02	1.966E-02
T_Vfraction	1.514	3.918E-02	3.992E-02
Q_Vfraction	1.962	4.466E-02	4.539E-02
T_Q_Vfraction	0.366	1.909E-02	1.983E-02

Coins per unit of time – Bulk Arrival

Here a table that resume all the analysed factors and their fraction of variation. The factors that are the most relevant one for the "coinsPerUnitOfTime" metric are the one coloured in red.

	Frac. Variation(%)	qi Confidence Interval (95%)	
		Lower Extreme	Upper Extreme
T	1.137	-8.667E-04	-6.622E-04
Q	48.515	4.892E-03	5.096E-03
Vfraction	25.170	-3.699E-03	-3.495E-03
T_Q	1.114	6.546E-04	8.592E-04
T_Vfraction	0.326	3.073E-04	5.119E-04
Q_Vfraction	23.041	3.339E-03	3.544E-03
T_Q_Vfraction	0.328	-5.127E-04	-3.081E-04

Utilization – Bulk Arrival

Here a table that resume all the analysed factors and their fraction of variation. The factors that are the most relevant one for the "Utilization" metric are the one coloured in red.

	Frac. Variation(%)	qi Confidence Interval (95%)	
		Lower Extreme	Upper Extreme
T	38.167	1.628E-01	1.646E-01
Q	2.403	4.018E-02	4.199E-02
Vfraction	56.731	-2.005E-01	-1.987E-01
T_Q	0.494	1.772E-02	1.953E-02
T_Vfraction	0.168	-1.177E-02	-9.965E-03
Q_Vfraction	1.649	3.312E-02	3.493E-02
T_Q_Vfraction	0.366	1.513E-02	1.694E-02

5.4.4 Results and notes about the factorial analysis

- One important result to be highlighted is that factor T is not relevant for the "CoinsPerUnitOfTime" metrics. The most relevant factors for this metric are "Q" and "vFraction" (and their combination)
- The most relevant factors for "Utilization" metric are "T" and "vFraction" (and their combination). The "Q" factor is instead not relevant.
- The factors that have a relevant fraction of variation are all characterised by having a qi that does not include zero in its confidence interval. There are some factors that have zero in the qi confidence interval, but their impact is negligible.

5.5 Experiments

We performed different types of experiments according to the results of the $2^k r$ factorial analysis. For each experiment we performed 35 replicas (with different seed-set in order to get uncorrelated results). In this project, we set ourselves the following objectives:

- Analyse the trend for the "CoinsPerUnitOfTime" ("C.U.T." for short) varying "Q" and "vFraction" factors both in single and bulk arrivals, trying to identify the optimum configuration(s). Analyse and compare the results for different scenarios
- Analyse the trend for the "Utilization" varying "T" and "vFraction" factors based on the results of the previous point, looking for the configuration(s) with minimum "Utilization".
- Analyse and compare the results for single and bulk arrivals for the same scenario

5.5.1 Earnings Optimization and Scenarios Comparison

For this experiment, the used configuration is the following:

- Δ = 300 [s]
- N = 24
- C = 1
- T = 210 [s]
- Q = [180, 900] [s] with step = 60 [s]

The configuration is based on the $2^k r$ factorial analysis, so the "T" factor is not changed in the simulation because of the negligible relevance for this metric. The factors that are not included in the previous list (λ, P and vFraction) are factors that are strictly correlated to the selected scenario. Their values are specified in the dedicated section.

For each graph in the following paragraphs (Figure 7 – 12) there will be a dashed line on the top that indicates the upper extreme of the 95% confidence interval of maximum "C.U.T." values reachable in that configuration. The maximum "C.U.T" is computed as the ratio between the total number of children arrived and the simulation time. This represents the "C.U.T." if we serve any arrived child in the queue. Each point in the graph has a confidence interval of 95% associated to it.

Low Load Scenario

The specific values for the factors not yet specified are:

- **λ** = 120 [s] for single arrival, 480[s] for bulk arrival
- **P** = 1 for single arrival, 1/4 for bulk arrival
- **vFraction** = [3/24, 15/24] with step = 1/24

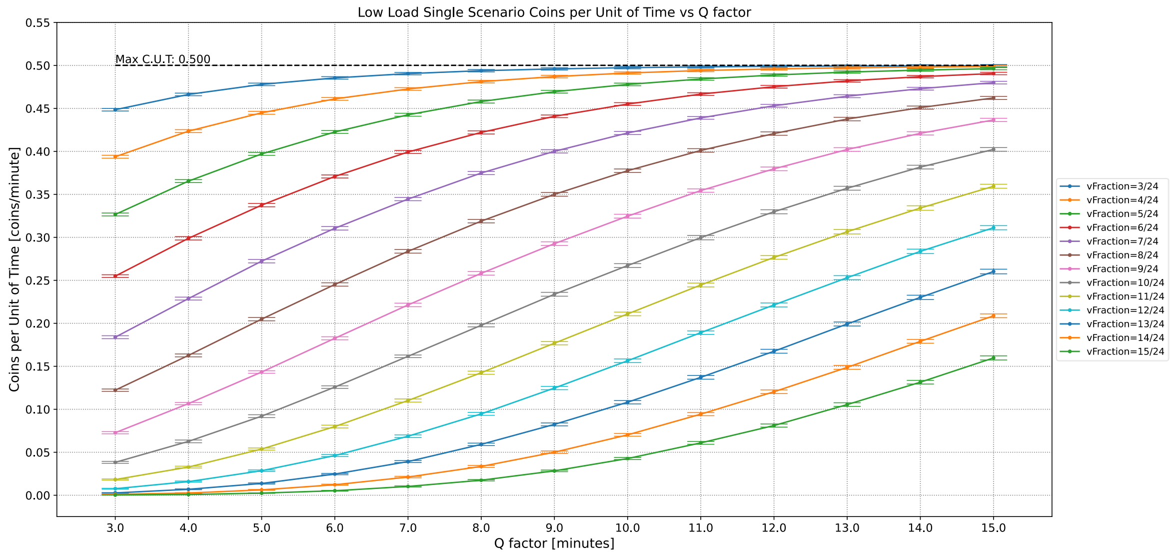

Figure 7: Variation of the "CoinsPerUnitOfTime" metric for different "Q" values (x-axis) and "vFraction" values (different curves) in case of single arrival in low load scenario

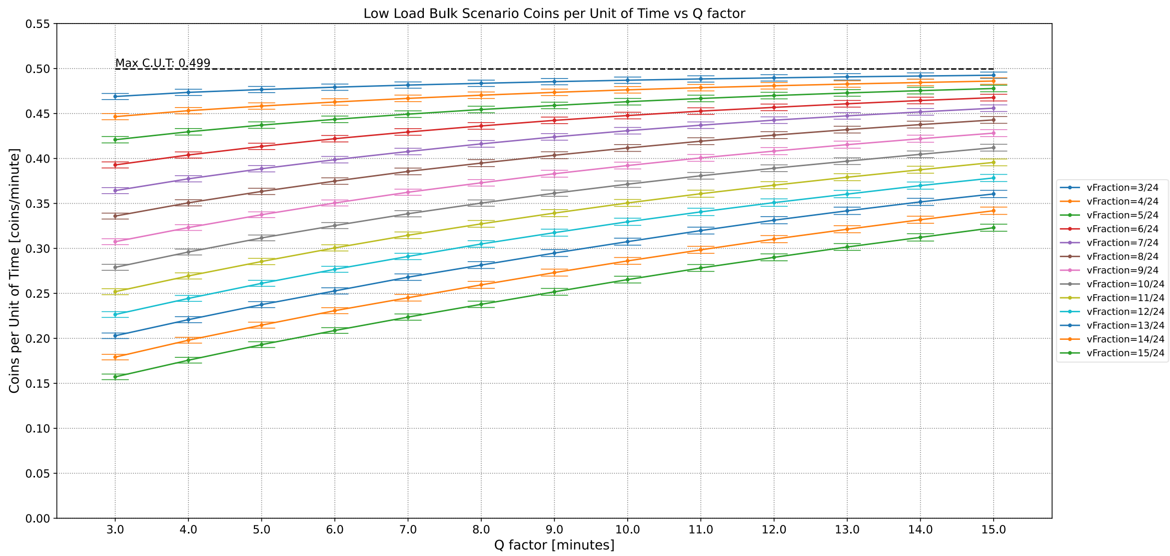

Figure 8: Variation of the "CoinsPerUnitOfTime" metric for different "Q" values (x-axis) and "vFraction" values (different curves) in case of bulk arrival in low load scenario

In these graphs (both single and bulk arrival case) we can notice that "C.U.T." grows with increasing "Q" values and decreasing "vFraction" values. This makes perfectly sense because:

- Increasing values of "Q" factor means that children in the queue have more "patience" so there are less children that will leave the queue before enjoying a ride.
- Decreasing values of "vFraction" factor means that the owner lets the merry-go-round start with fewer children, so the time to wait for the missing people to arrive at the start of the merry-go-round is gradually reduced.

By analysing the single arrival case, going towards smaller values of 'vFraction', the associated curves have different trends. This is due to the fact that moving on the x-axis these curves tend to reach the practical maximum value, so they change their concavity and tend to collapse all in the same value. With regard to the bulk arrival case, the curves still change their trend (less visibly than before) and tend to reach a practical maximum value and collapse on it.

After this graphs analysis step, we can conclude that, for any values of “Q” in our simulations, the best configuration that let optimize the “C.U.T.” is the one with “vFraction” {3/24, 4/24}.

Medium Load Scenario

The specific values for the factors not yet specified are:

- **λ** = 60 [s] for single arrival, 240[s] for bulk arrival
- **P** = 1 for single arrival, 1/4 for bulk arrival
- **vFraction** = [6/24, 1] with step = 2/24

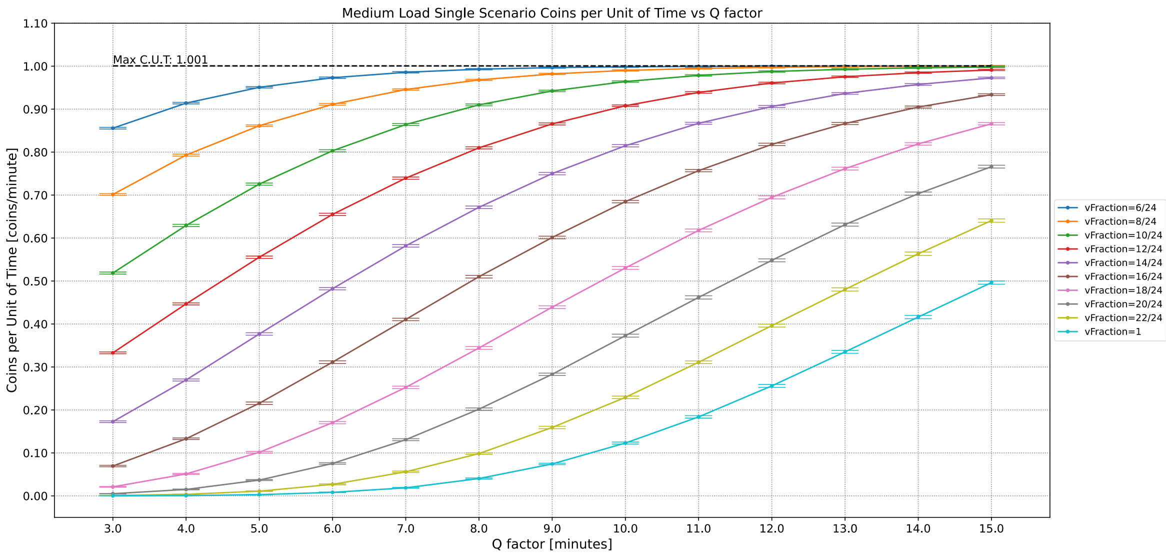

Figure 9: Variation of the "CoinsPerUnitOfTime" metric for different "Q" values (x-axis) and "vFraction" values (different curves) in case of single arrival in medium load scenario

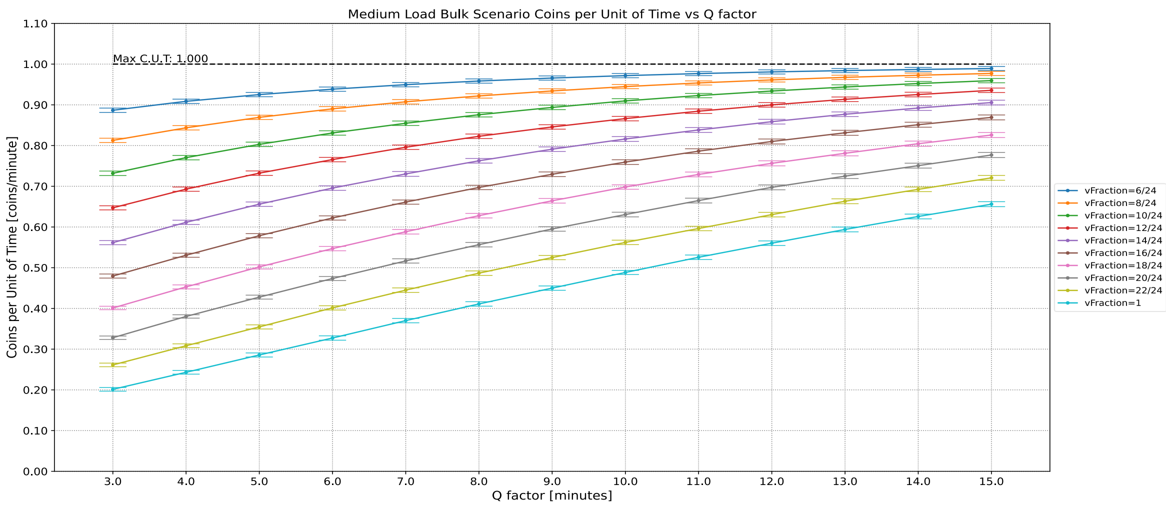

Figure 10: Variation of the "CoinsPerUnitOfTime" metric for different "Q" values (x-axis) and "vFraction" values (different curves) in case of bulk arrival in medium load scenario

From now on, in this scenario and also in the high load one, the “vFraction” factor assume different values with respect to the previous scenario (see at the beginning of this chapter). In general, the values start from a bigger minimum and arrive at the theoretical maximum value of 1 because of the high number of arrivals. The usage of different values for different scenarios is due to the fact that, in the low load one, the “vFraction” values used from now on did not let us achieve interesting results.

Comparing the medium load graphs with the ones for the previous scenario, we can notice that:

- Curves in both types of arrivals are very similar for the two different scenarios. The curves still tend to a practical maximum “C.U.T.” and change their trend over “Q” and for different “vFraction” values. The same claims as in low load scenario still hold as regard this aspect.
- Because of the higher load than before, here we can appreciate higher values of “C.U.T.” reached for all the analysed configurations. In particular, if we compare the common configurations for the two scenarios, we can see that in the medium one the reached values are anytime bigger than the ones in the low load scenario. Moreover, the maximum practical “C.U.T.” is bigger than the one in the low load scenario

High Load Scenario

The specific values for the factors not yet specified are:

- **λ** = 20 [s] for single arrival, 80[s] for bulk arrival
- **P** = 1 for single arrival, 1/4 for bulk arrival
- **vFraction** = [6/24, 1] with step = 2/24

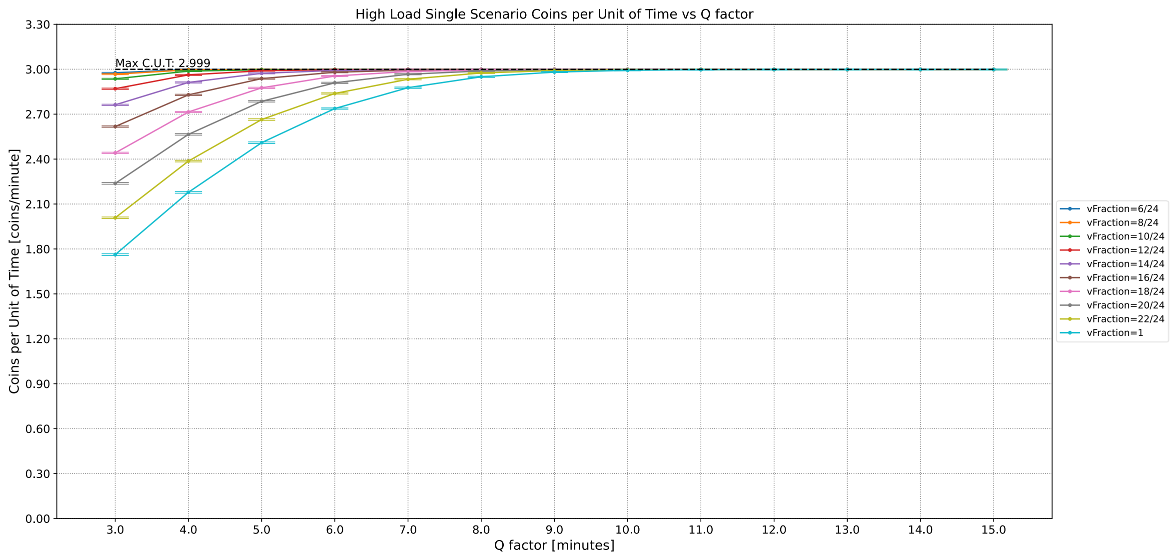

Figure 11: Variation of the "CoinsPerUnitOfTime" metric for different "Q" values (x-axis) and "vFraction" values (different curves) in case of single arrival in high load scenario

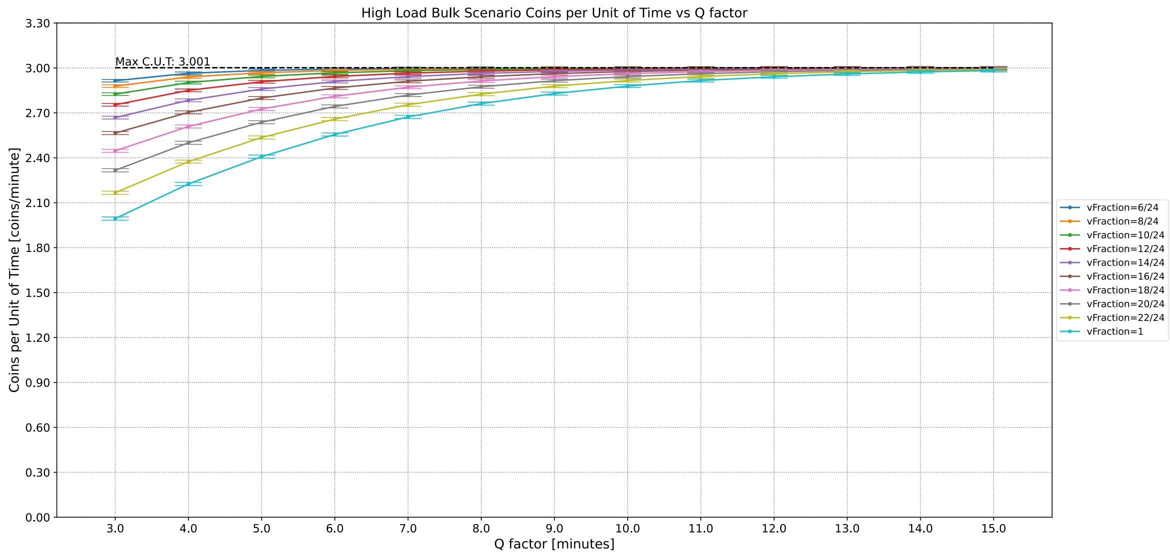

Figure 12: Variation of the "CoinsPerUnitOfTime" metric for different "Q" values (x-axis) and "vFraction" values (different curves) in case of bulk arrival in high load scenario

In this scenario, the claims associated with the medium load one still hold. Due to the very high load, all the “vFraction” reach higher value of “C.U.T.” also for small values of “Q”, that because even in case of “not patience” children we cover the high number of quit from the queue with a high number of arriving children. We can also notice that the high load scenario has a greater number of curves that reach the practical maximum values of “C.U.T.”.

5.5.2 Maximum Earning and Minimum Utilization

After analysing the configuration that maximize the “CoinsPerUnitOfTime” metric, now the focus is on the “Utilization” metric. As we can see from the 2^kr factorial analysis, the “T” factor doesn’t change significantly the “C.U.T.” metric but has a heavy impact on the “Utilization”. The graphs that support this claim are not added to this documentation because are not significant for the treatment. The values are equals in all the six different cases (three scenarios and two types of arrival) and just looking at the results we can not appreciate any significant differences. In this experiment the objective is to make some considerations regarding the earning and the cost for the owner. In particular, associating to a higher utilization of the merry-go-round, a higher cost for its maintainability (as reasonable), the focus here is trying to find a trade-off between the earning and the cost for the owner. That type of association can be transposed to a trade-off between “CoinsPerUnitOfTime” metric (to be maximized) and the “Utilization” metric (to be minimized).

Based on the results achieved in the previous section, we decided to extend this study not only to the optimal configuration at all for each scenario and type of arrival. The selected “candidates” are any configurations that reach, in a certain scenario and type of arrival, that are at least greater than 90% of maximum “C.U.T.”. The following analysis is based considering the results obtained at a fixed “Q” factor as 15 minutes. Then we will discuss shortly about different values of “Q”.

In the following table the selected “vFraction” values for each scenario and type of arrival:

	Low Load scenario	**Medium Load scenario**	**High Load scenario**
Single arrival	[3/24, 8/24] step 1/24	[6/24, 16/24] step 2/24	Any
Bulk arrival	[3/24, 7/24] step 1/24	[6/24, 14/24] step 2/24	Any

Taking the medium load single arrival as an example, the graph below shows the “Utilization” metric for the selected “vFraction” configurations and different values of “T” factor. The points in the graph have an associated confidence interval of 95%. These confidence intervals are very small and appear to be as strict horizontal lines. This is due to the fact that the sample standard deviations are small.

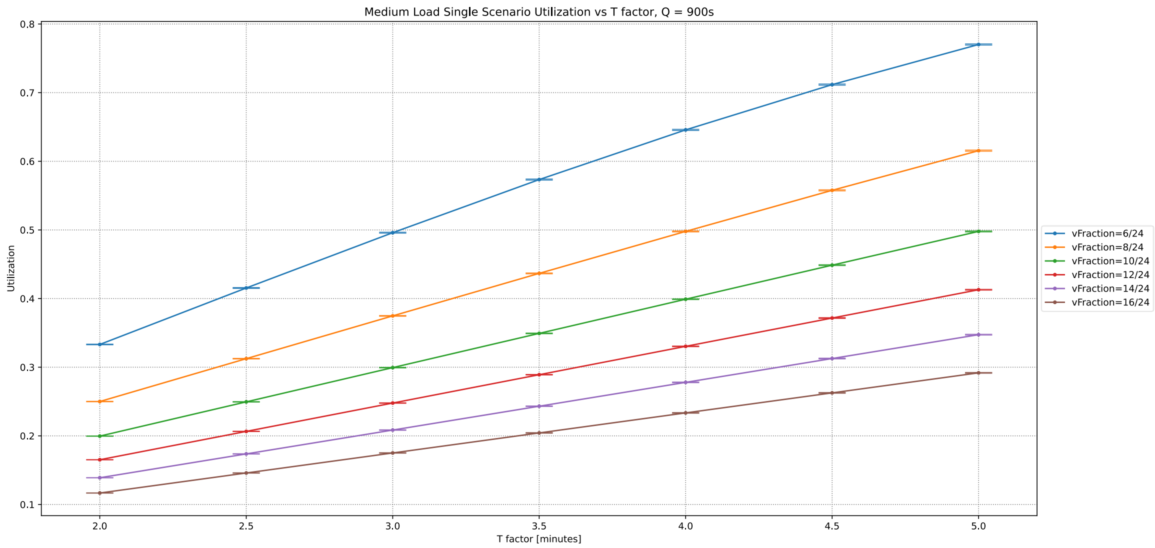

Figure 13: Utilization graph for medium load single arrival case, for different values of "T" (x-axis) and "vFraction" (different curves)

As we can deduce from the graph above, given a certain value for the “T” factor, we get the minimum “Utilization” value for the maximum “vFraction” value (16/24 in this case). Taking a certain value for the “vFraction” factor, we get the minimum “Utilization” value for the minimum “T” value (2 minutes in this case). So, we can conclude that, to achieve a minimum “Utilization” given the above selected configurations, we need to choose the one that has the highest “vFraction” value and apply to it the smallest possible “T” value (2 minutes). This let us minimize the cost associated with the “Utilization” maintaining an acceptable “C.U.T.” that is at least 90% of the maximum.

For different scenarios, we get different values but not different trends. The claims that we have just presented are still valid and hold in any case. So, for the ease of treatment, we decided to not put graphs in this documentation that don’t add any information to the ones that we have just presented.

By changing the value of fixed “Q” factor, there are some important considerations that must be highlighted. For different “Q” factor values, the number of curves that reach the maximum tends to decrease. So based on that value we can have fewer or zero curves that reach maximum “C.U.T.”. In the first case we must compare less curves (or none in case we have just one) but the claims above still hold. Otherwise, if no curve reaches the maximum value, we need to take as the best solution the curve with highest reached “CoinsPerUnitOfTime” metric. In any case the minimum “Utilization” is still reachable by selecting the minimum “T” possible value.

5.5.3 Single arrivals vs Bulk arrivals

Analysing the results that come from the graphs that represents the “C.U.T.” trends, previously described in 5.5.1, we noticed that there are some differences between single and bulk arrivals in the same configuration. Because of that, we decided to deepen this. Taking into consideration the low load scenario, the following graph shows the trend for the “C.U.T.” metrics for three different values of “vFraction” varying “Q” and for both types of arrivals (single and bulk curves of a same configuration have same colour in the graph). Each point in the graph has associated a 95% confidence interval.

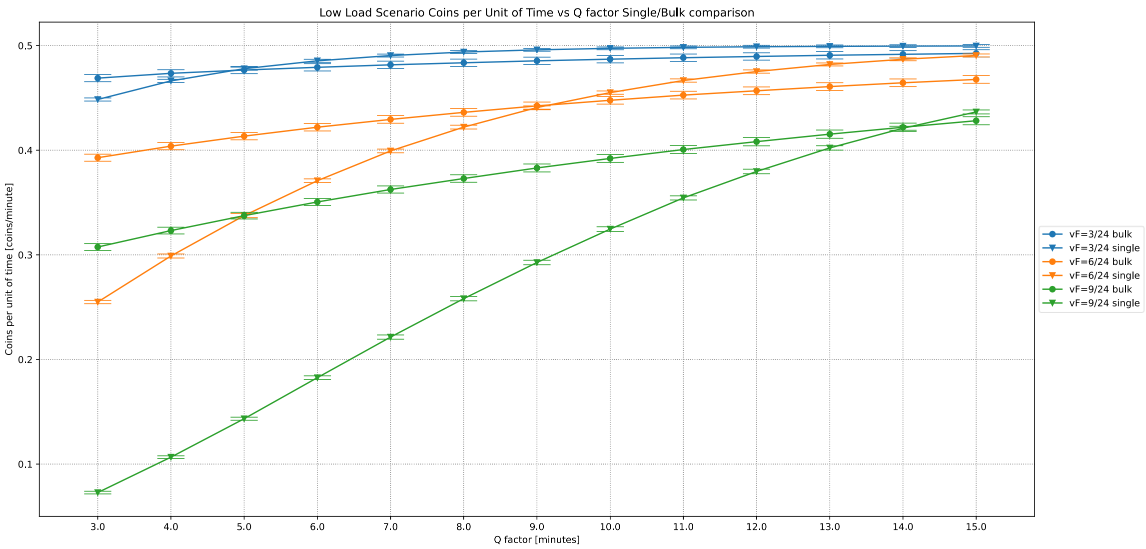

Figure 14: "CoinsPerUnitOfTime" comparison between single and bulk arrivals in low load scenario for different configurations (varying "vFraction" factor and "Q" factor).

By looking at the graph above, we can notice that, for each couple of configurations that differ only for the type of arrival, there is a point in which the two curves intersect each other. The associated value of "Q" to this point is different in different configurations. Calling "Q_i" this incident point, we can analyse the trend of the curves by looking them in three different intervals:

- In the points in which "Q" < "Q_i" the bulk arrival curve is above the single arrival one. So the bulk arrival configuration has a higher "C.U.T." values with respect to single arrival one. This is due to the fact that, for small values of "Q", the children have little "patience" so they tend to quit from the queue before reaching the minimum number of children to start a ride. The single arrival type, in this case, is the most damaged configuration because for each arrival just a child joins the queue. So, in general, a child needs to wait several arrivals until reaching the threshold. The bulk arrival type instead can cover the whole threshold also with just one arrival. Even if a child in the queue need to wait, in general, more than in single case the next arrival, that next arrival can by its own let the merry-go-round start because it could be composed of enough children to exceed the threshold
- In the points in which "Q" is close to "Q_i", the results of the single and bulk arrivals configurations are comparable
- In the points in which "Q" > "Q_i", the single arrival curve is above the bulk arrival one. So the single arrival configuration has a higher "C.U.T." values with respect to bulk arrival one. This is due to the fact that, having high values of "patience", before a child quits from the queue, several other children will arrive. The bulk arrival type, in this case, is the most damaged configuration because waiting for a bigger amount of time could lead to add to the queue a number of children that is not sufficient to exceed the threshold. In this case, the children that are waiting in the queue should wait a lot of time and this could let their "patience" to exceed, deciding so to quit.

Both the above results are due to the different trends of the arrivals. In fact, in the single arrival case it is more "regular" given that it is derived by a single source of randomicity (the exponential distribution for the inter-arrival time). Instead, the bulk arrival type is characterized by a less "regular" trend, due to the fact that it is derived by two sources of randomicity (the exponential distribution for the inter-arrival time and the geometric distribution for the number of children that compose each group). This fact is demonstrated by the mean variances of the number of children in the queue, that is smaller in the single arrival case, showed in the following table:

	Mean Variance	**95% Confidence Interval**
Single Arrival	3.9282	[3.9255, 3.9308]
Bulk Arrival	15.6543	[15.5010, 15.8076]

In order to show in a clearer way this result, we decided to draw a box plot of the number of children in the queue of the first replica of our experiment.

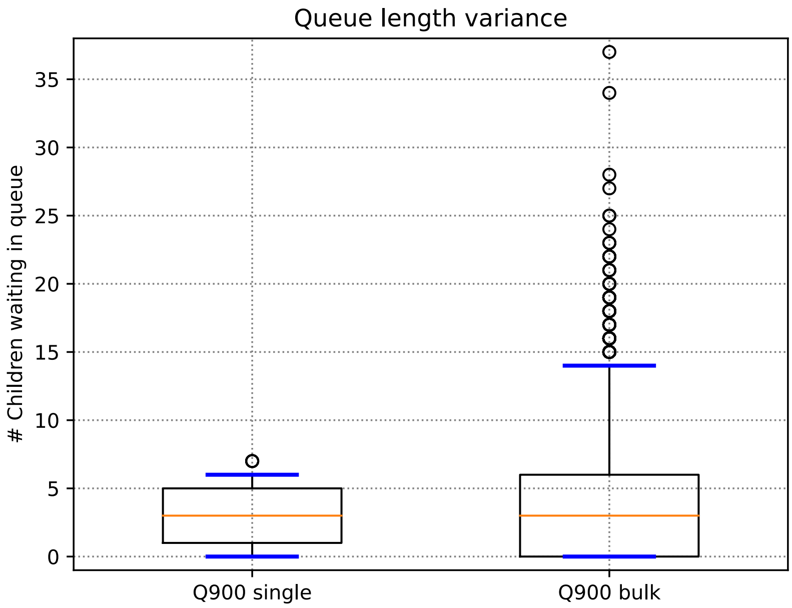

Figure 15: Box Plot of the number of children in the queue in single and bulk arrivals configurations. "vFraction" = 6/24

The value of "Q_i" depends on the value of "vFraction" and the type of load we are analyzing. In fact, by increasing "vFraction" we can see that the intersection point moves towards higher "Q" values. This is because with a higher "vFraction" the children must wait more, so higher values of "patience" are required in order to exceed the minimum threshold to enjoy a ride. This condition emphasizes the convenience of bulk arrivals with respect to the single one, for the same reasons described before.

If, instead, we take into account different scenarios with different values of "λ" factor, "Q_i" moves towards smaller "Q" values for the same "vFraction" values when for increasing load. This is due to the fact that having a higher load, the time that is needed to exceed the threshold is generally smaller, requiring lower values of "patience", so the number of quits decrease. Obviously, all the claims we made before are still valid for different "vFraction" values in different load scenarios.

6 CONCLUSIONS

We can conclude our work affirming that the optimal configurations from the point of view of the earned coins per unit of time are the ones with the smallest “vFraction” and highest “Q” values, taking into account the range of values we considered in our experiments. This is because smaller “vFraction” means less time to wait for the children and higher “Q” values mean that they have more “patience”. However, the owner can tune the time needed to perform a ride in order to reduce the “Utilization” (and so the costs) of the MGR, without affecting the earning. In particular, the time to perform a ride has to be as small as possible in order to get the minimum “Utilization”. In the same way, the owner can also change “vFraction” values in order to decrease the “Utilization”, but in this case it would also affect its earning. So, the owner must take into account the trade-off between the “Utilization” and the earning to select the “vFraction” that is more suitable depending on the costs associated with the “Utilization”.

We also suggest to take in consideration the possibility to increase the time needed for performing a ride and the requested coins per child for each ride. This could lead to a significant increasing of earnings without having heavily cost effects. For example, with same configurations and scenarios, doubling the requested coins, the owner would earn twice as much as before. However, this solution would require additional research to study how long a ride should last in order to get the same amount of people willing to pay twice the price we analysed in our project. In fact, the risk is to increase the duration of a ride too much so that the costs exceed the benefits.