PERFORMANCE EVALUATION

The Carrefour

Overview

Analysis of the performance of two policies related to the management of queuing at tills.
Project assigned during 1° academic year in master degree of Computer Engineering.
February 2020, Pisa.

Leonardo Bacciottini
Federico Pacini
Marco Pettorali
A.A 2019/2020

Index

1. Introduction

1.1 Project description

The project is about the analysis of the behaviour of a supermarket in different scenarios. In particular two are the policies, appliable to the management of the queue at the checkout, that we are going to compare in terms of performance. From now on we will call them P1 and P2 for the sake of brevity. Hence:

> **P1.** Implement a *single* queue in the supermarket and send the *head-of-line* as soon as a till is idle, considering that if the j^{th} till is set free a customer needs a Δ*j unit of time to reach that one and obviously it will be idle throughout the duration of the walk. (Fig1)
>
> **P2.** Implement a *queue for each till* managed in *FIFO* order, allowing customers to enqueue to whatever till they want but expecting that they will enqueue in that one whose till is the shortest one. (Fig2)

Moreover, we know that in the supermarket there are N tills which are spaced by a certain distance, and a customer needs Δ seconds to move from the x^{th} to the $(x+1)^{th}$ one, and this interests only P1. These are factors that can be varied to represent different scenarios that we encounter in real life but however they remain constant during a simulation.

1.2 Objectives

The global objective of the project is to understand in which cases "P1 is better than P2". This is achievable only if we set correctly the scenarios in which the two policies will work: then we will analyse some typical metrics of queueing at supermarkets to see which policy minimizes (or maximizes) them.

We focused on a single customer: he wants his waiting time in the queue to be minimized in terms of mean value. Moreover, he would like to know that in this supermarket he has a high chance (say 98%) to stay in the queue not more than a certain amount of time.

Then we focused on the point of view of the manager of the supermarket: a policy is good if it can grant a less crowding in the supermarket, and he wants to know if it's better to increase the number of tills in the supermarket or improve the capacity of the single tills. Moreover, he is interested in knowing the fairness of the policies: the fairer is the supermarket, the more customers will be satisfied.

1.3 Performance indexes

In order to get the aims previously described, we are going to extract the following indexes:

- Extracting the mean value of the response time distribution and its median, 98^{th} percentile and Lorenz curve gap.
- Extracting the mean value of customers in the system.
- Extracting the mean waiting time, its median, 98^{th} percentile and Lorenz curve gap.

1.4 Scenarios

The inter-arrival time of the customers and their service time are IID RV whose distributions are:

> **Inter-arrival time.** Exponential
>
> **Service time.** Exponential, Lognormal, Normal

Knowing that, we are going to analyse the following scenarios:

- Exponential inter-arrival time and exponential service time.

- Exponential inter-arrival time and lognormal service time.
- Exponential inter-arrival time and normal service time.

2. Model

2.1 Description

Since the two policies have different organization but share some factors, we will show them one after the other and then discuss about the appearing symbols.

A scheme representing P1 is the following:

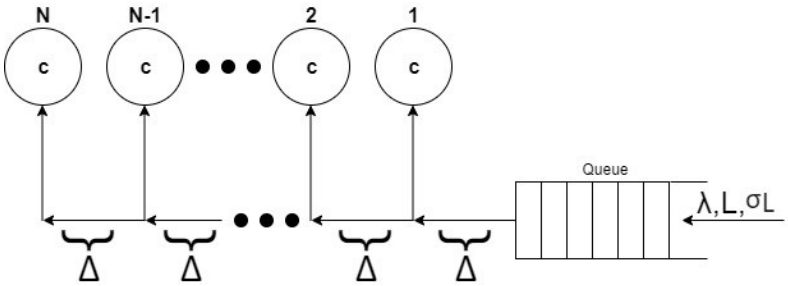


Figure 1, Policy P1

While P2's one is the following:

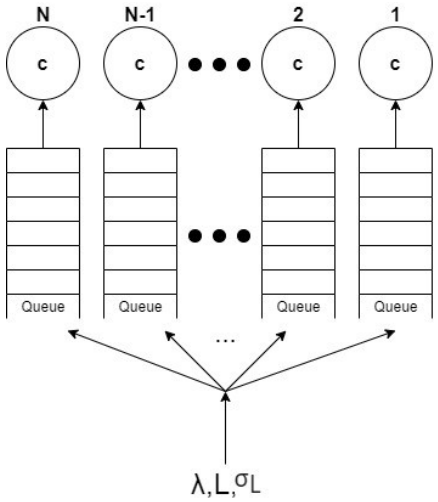


Figure 2, Policy P2

In the two figures we can see the following symbols:

- N is the number of tills.
- c is the speed ratio of a till, i.e. the number of items checked per second.
- λ is the inter-arrival rate of a customer.
- L is the cart length of an arriving customer (as a real number)
- σ_L is the variance of the distribution of the items in a cart of an arriving customer (for lognormal and normal distributions of the number of items in the cart)

In figure 1 we can see the following one:

- Δ is the time needed to reach the j^{th} till from the $(j+1)^{th}$

2.2 Assumptions

Modelling the system, we have assumed the following:

- Infinite queuing space: no one can be rejected.
- Inter-arrival time and number of items in a cart are independent RVs.
- All the tills have the same performance, i.e. *c* is constant.
- All the queues follow FIFO policy.
- Each arrived customer must be checked out: no afterthoughts are admitted.
- The number of tills remains constant during a simulation.
- The spacing Δ between two neighbour tills, in P1, is the same for each couple.

2.3 Factors

The factors that influence the system are the following:

- Number of tills: *N*
- capacity of a till: *c* [items processed/second]
- Inter-arrival rate: *λ* [average arrival each λ seconds]
- Cart length: *L*
- Variance of the number of items in a cart: σ_L
- Distance space between tills: *Δ*
- Policy: P_i , *i*= 1,2

2.4 Approximated model for theory validation

In this section we are going to formulate a model for P1 and one for P2, according to queueing theory, that we will use to validate ours: obviously we will make assumption that lead us to approximate them with the models seen during course lessons. Since our knowledge is stronger on exponential distributions, we will deal with them at least to validate the scenario in case of exponential inter-arrival and service time.

Starting from P1 we can observe that if we assume Δ=0 we can bring back ours to a M/M/C where C=N. From QT we know that:

$$\rho = \frac{\lambda}{N\ \mu} \text{ assuming that } E[t_s] = \frac{1}{\mu} = \frac{L}{c}$$

Actually, we should add the contribution of *Δ * j* to the mean service time, but in this case we couldn't use the model above. So, we can compute the mean waiting time:

$$E[W] = P_0 \frac{(\lambda/\mu)^N\ (1/N\mu)}{N!\ (1-\rho)^2}$$

And from steady state probability (making sure that $\rho < 1$):

$$P_0 = \frac{1}{\sum_{n=0}^{N-1} \frac{(\lambda/\mu)^n}{n!} + \frac{(\lambda/\mu)^N}{N!} \frac{1}{1-\rho}}$$

And the mean response time:

$$E[R] = E[W] + E[t_s]$$

And the mean enqueued customers is:

$$E[N_q] = P_0 \frac{(\lambda/\mu)^N}{N!} \frac{1/N\mu}{(1-\rho)^2}$$

For P2 we consider only one till for the sake of simplicity since we don't have strong knowledge of load balancing: assuming that we can use the M/M/1 model. Hence:

$\rho = \frac{\lambda}{\mu}$ assuming that $E[t_s] = \frac{1}{\mu} = \frac{L}{c}$

So, we can compute the mean response time:

$$E[R] = \frac{1}{\mu(1-\rho)}$$

The mean waiting time is:

$$E[W] = E[R] - E[t_s]$$

The mean enqueued customers is:

$$E[N_q] = \frac{1}{\mu - \lambda} - \rho$$

3. Implementation

3.1 Omnet++ model

The following picture is the network that represent our simulation model in *Omnet++.*

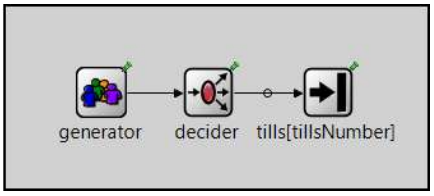


Figure 3, Omnet++ model

The components used are:

- **generator**: its aim is to generate arrivals of customers and related number of items in their cart, according to the chosen distribution.
- **decider:** according to the chosen policy it lets the incoming customer be enqueued at his queue (P1) and send it to the correct queue as soon as it reaches the head-of-line and a till is free or assigns the incoming customer to the till which has the shortest queue (P2).
- **tills[tillsNumber]:** is a vector of tills; each till can have or not a queue (according to the policy) and process the customers' cart at rate *c*.

3.2 Verification

In order to verify that the implementation behaves as our model we have performed the following tests.

1. **Deterministic inter-arrival and service time with high load**: setting very fast arrival and very high service time we have visually checked that the decider chose the correct till every time, both P1 and P2.
2. **Exponential inter-arrival and service time with Δ=0 for P1:** according to the theoretical model of section 2.4 for P1, we have collected statistics about mean response and waiting time with a well known ρ: checking the theoretical results with empirical ones more or less were the same.

3 **Exponential inter-arrival and service time with N=1 for P2:** according to the theoretical model of section 2.4 for P2, we have collected statistics about mean response and waiting time with a well known ρ: checking the theoretical results with empirical ones more or less were the same.

4 **Continuity test incrementing N, upgrading c and raising λ:** obviously changing only one of the three at time, we have collected statistics about mean response and waiting time: As expected we have observed that incrementing N we would experience decreasing waiting and response time and so as for upgrading c; instead for raising λ we would observe incremental response and waiting time until they diverge to infinity when the system become instable.

5 **Exponential inter-arrival and service time with N>1 for P2:** adding more than one till in P2 avoids us using a model composed by N M/M/1 in parallel due to the arrivals at each till that are not exponential, since the customer coming after another is more likely to go to a different till. A confirmation of it can be found in *figure 4*, where one can observe that the distribution of arrivals at a random till isn't distributed as an exponential, especially for low values.

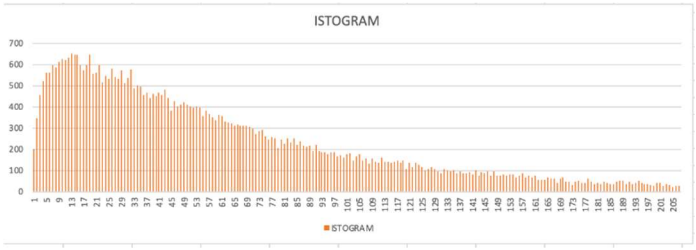


Figure 4, Distribution of arrivals at a till in P2

3.3 Range of factors

The following list shows what is the range chosen for each factor that we have considered meaningful:

- $N \in [\,5\,,\,20\,]$
- $c \in [\,0.3\,,\,1\,]$
- $\lambda \in [\,0.3\,,\,1\,]$
- $L \in [\,4.5\,,\,16\,]$
- $\sigma_L \in [\,0.64\,,\,1\,]$
- $\Delta \in [\,0.5\,,\,1.5\,]$

4.Simulation

4.1 Estimating warm-up and simulation time

In order to set the warm-up time, we have decided to run 10 repetitions of the worst scenario that we had in the simulation (and quite certainly it is even in real conditions), taking the response time as the output of the simulations.

Setting N=5, λ=0.1, c=0.3, Δ=1s, we have plotted the mean of the of each output, getting only one trajectory: observing it we realise find out that after t=50000s the trajectory is stable. (Fig5)

Once set warm-up time correctly, we have concentrated on the simulation time: observing different replicas of the worst scenario we have observed that it's enough to simulate t=216000s to get a large number of info, and a right trade-off between time required to simulate and smaller CI with 95% confidence.

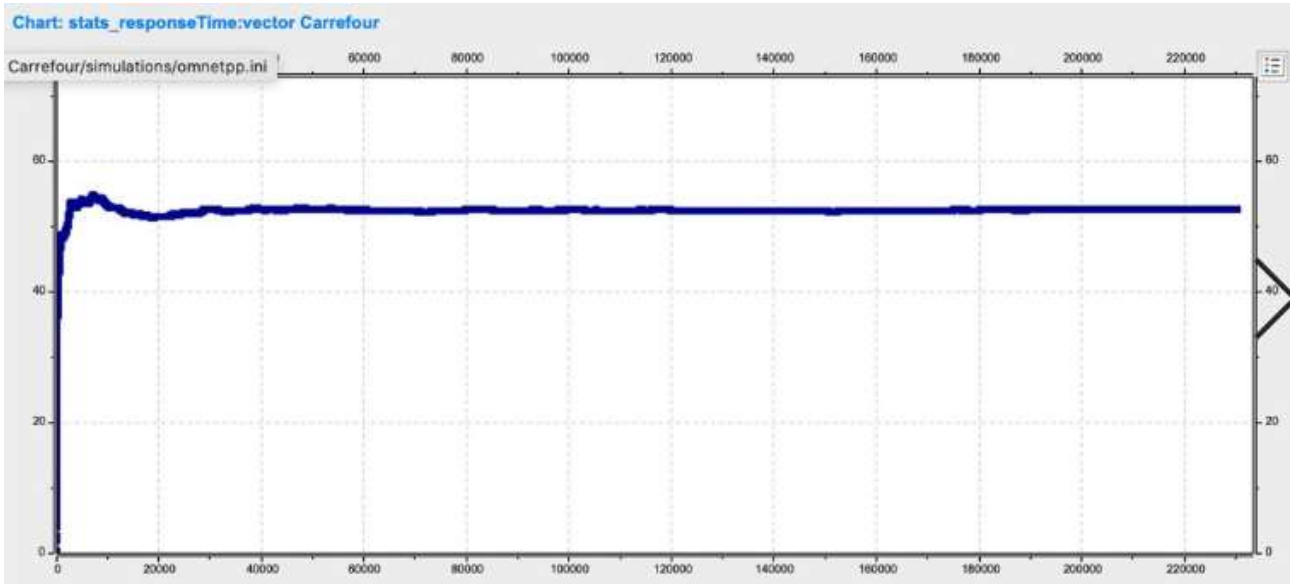


Figure 5, Mean value of 10 aggregated repetitions

4.2 IID samples

To ensure the IIDness of the samples we have decided to run 35 repetitions (not to get overlapping CI for metrics of the scenarios that we want to compare later) of the same scenario using 35 different initial seeds for the RNGs, obviously. Then, for each repetition, we have extracted the mean value of the metrics in order to get 35 samples that obviously would be independent.

Thanks to the high number of IID samples, we can compute the CI of the aggregated metric applying CLT since samples are more than 30 in number.

4.3 $2^k r$ analysis

In order to discern what are the factors that mostly impact on the metrics we have decided to realise a $2^k r$ analysis where k=4 and r=35.

The factors that have been analysed are:

- P = policy P_i [1 , 2]
- D = delta Δ [0.5 , 2]
- C = capacity c [0.3 , 1]
- T = till N [5 , 10]

Following the straightforward procedure considering once exponential inter-arrival and service time and once lognormal service time, we have got the output of the runs. Then we have checked the hypothesis on residuals of each metric of interest using QQ-plots to ensure Normal distributed and lag plots to be sure of IIDness and constant standard deviation.

Hence the most important results for waiting time are:

- N, c are the predominant factors (with 32,5% and 28% respectively) and also their interplay is important (27,9%). This indicates that these factors affect waiting time performances in quite the same way.

In both distributions the explained variability is over an acceptable threshold: 99% for exponential ones and 97,5% for exponential-lognormal ones.

Other results has been get analysing the response time:

- The capacity of a till c has a high impact on it (76,9% on *exponential scenario*; 68,9% on *lognormal*).
- The number of tills N has less impact but still considerable among the others (9,6 % on *exponential scenario*; 11,9% on *lognormal*).
- The interplay between N and c plays a considerable impact on it (9,5 % on *exponential scenario*; 11,7% on *lognormal*).

In both distributions the explained variability is over an acceptable threshold: 99% for exponential ones and 97,5% for exponential-lognormal ones. Of course, it could be expected that the capacity and number of tills would be much more influent than the policy or other factors, but it's not always possible to increase these factors, and that's why policies are important.

4.4 Results

In this section we will show the results according to the objectives previously described in [1.2].

In order to understand what parameters do define the workload of the simulation, let's see the following table:

	High load	Medium load	Low load
Load ratio	4/5	3/4	3/5

The load ratio is the M/M/C utilization:

$$\rho = \frac{\lambda}{N\mu} = \frac{\lambda L}{Nc}$$

We verified that also in our case this is a good index to have an idea of the system stress. Indeed, when it is greater than 1, the system is unstable. It's important to note that in the case of P1, μ should be computed as (c/L)*((N+1)/2)*Δ. This is the reason why in the case of high load and 20 tills, P1 (Δ=1,5s) is wildly unstable.

4.4.1 Mean waiting time in the supermarket

The mean waiting time experienced is important from the point of view of the customers: people do not like to wait.
Before getting into details, we have to know that:

- For the following statistics, waiting time includes the time taken by a customer to reach the till. This because this time is valuable for comparisons between P1 and P2, since it's not present in the latter, and because it's reasonable that, from a customer point of view, this time is seen as part of the total amount of time spent waiting.
- All the comparisons made between indexes and performances use 95% confidence intervals: this means that we have at least 95% confidence that what we have assumed is true.
- In some graphs we will see P1 (Close): it indicates the policy P1 where the tills are disposed such that it's faster to move from one till to another (delta = 0.5s) instead of in the other case when delta is 1.5s. These values have been chosen thinking about possible real implementations for P1 that may be more or less optimized.
- Second is the unit of time.

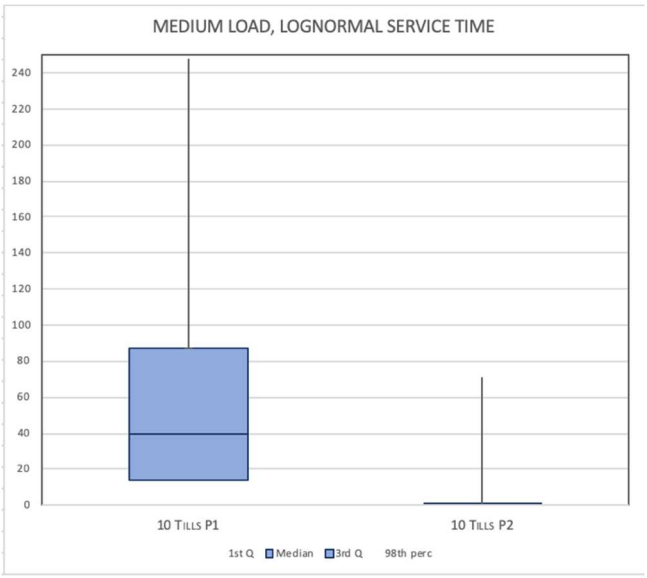
Let's see some of the main results.

4.4.1.1 Mean value, exponential service time, high workload.

With a high workload and big market, P1 does not have a desirable behavior: when the supermarket has more tills, the mean waiting time is very high compared to P2. This is due to the fact that the Δ parameter has more importance in a big supermarket, because the last till is really far and it will take a lot of time to reach it. This obviously affects the whole performance of the system. In each sub-scenario P2 always wins: P1 performances get closer to P2 when tills are closer, i.e. when the parameter Δ is smaller and less dominant.

4.4.1.2 Mean value, *normal service time, medium workload*

For the normal service time, a medium workload represents well a real situation; limit cases of workloads, indeed, are not so different from a medium workload, due to the fact that normal distribution is a light-tailed one. P1 still has the same behavior as seen in the previous paragraph due to the big impact of parameter Δ. In this graph, we can see that P2 always wins and P1 has a similar behavior only when Δ is small.



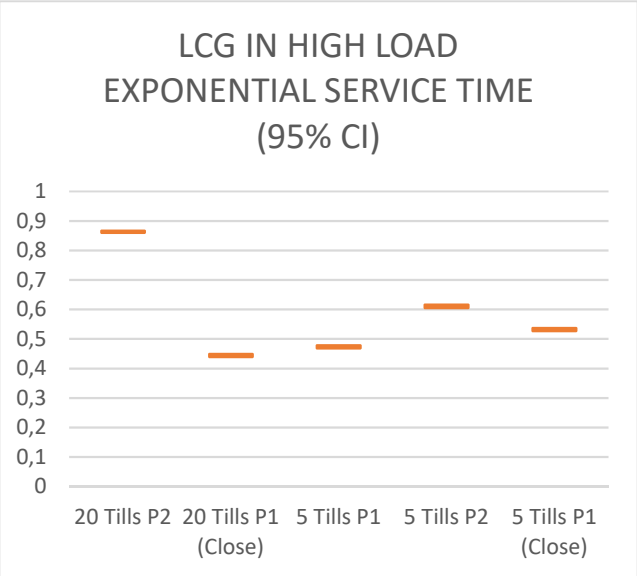
4.4.1.3 *Median, lognormal service time, big market, high workload*

A lognormal, due to the fact that its variance is huge, introduces more queueing in the system, and so its waiting times would suffer a lot this situation. Since lognormal distribution is a heavy tailed one, we decided to pick the median as a key performance indicator, since it's more robust to the outliers that this distribution sometimes introduces into the system.

An important aspect that we have observed is in the beside graph: we can see that median for P2 in a big supermarket, with medium workload is 0 (this is true even for high workload). This means that statistically more than 50% of the population isn't queued in these conditions, so P2 is a really robust policy even if service times are lognormal. The interesting fact is that under the same conditions, P1 is really bad if Δ = 1,5s, as we may see in the picture. We'll be back on this topic.

4.4.2 LCG for waiting time

This parameter is interesting to understand how waiting time is distributed among the population: the more LCG is closer to 1, the more the system is unfair, i.e. customers will experience very different waiting time spent in queue; vice versa if it approaches 0.

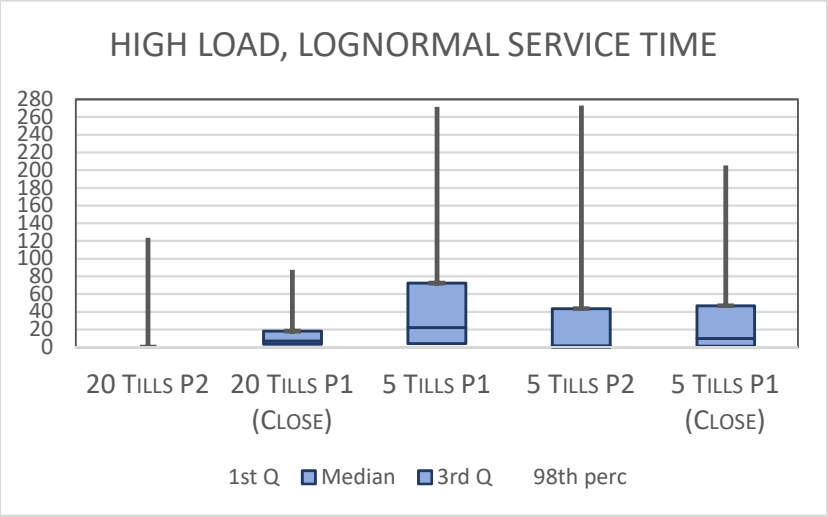


We can notice that P2 has in general a higher value of the LCG, even with different distributions (which for the sake of brevity we won't show); this means that P2 is a less fair policy. This is not surprising indeed because, as better seen with high number of tills, P2 can grant less queueing than P1: in particular P2 loses more fairness as the number of tills is higher. This happens because with many tills there will be many customers who find at least an empty till when they arrive, and others who get there in critical moments who will wait for a long time. Instead for P1, even if they find, for instance, the last one free, they have to walk for "Δ * N" seconds to get there and so the advantage (or unfairness, looked from another point of view) is lost. Being unfair could be an advantage or not: a customer could be lucky and experience a very little waiting time but he could be hapless and so wait a lot more than the first one. Usually if mean value is quite the same, fairness is appreciated.

4.4.2 Box Plots for waiting time

This graph is useful to have a glance on what are the percentiles of the policies: customers will be gladdened knowing that there is a high percentage not to wait more than a certain threshold.

Since quite all scenarios are similar, for the sake of brevity we will show the following: it differs from the exponential only for the scale of ordinates. The first thing caught by observing these plots is that P1 (delta = 1.5s) performances are very bad when there is a high number of open tills (10-20 in our case). In this situations P1 should be discarded a priori. In the high load plot P1 (delta = 1.5s) has no 20 tills entry. This because in that scenario P1 is not even stable! The reason is due to the increasing weight of the delay in the mean service time



In general, P2 has a lower median. This could be expected, since P1 introduces the delay of reaching the destination till. What is noticeable is that at high load, P2's 98th percentiles are consistently higher than P1 (delta=0,5s)'s. This means that when the system is stressed enough (e.g. in busy days), P1 gives more confidence to a customer in not waiting more than a reasonable amount of time, as we have seen for the LCG. Moreover, another confirmation of what we have seen in with LGC is that when there's a high number of open tills, P2 gives at least 75% probability of having zero waiting time even in the case of high load.

4.4.3 Mean number of customers

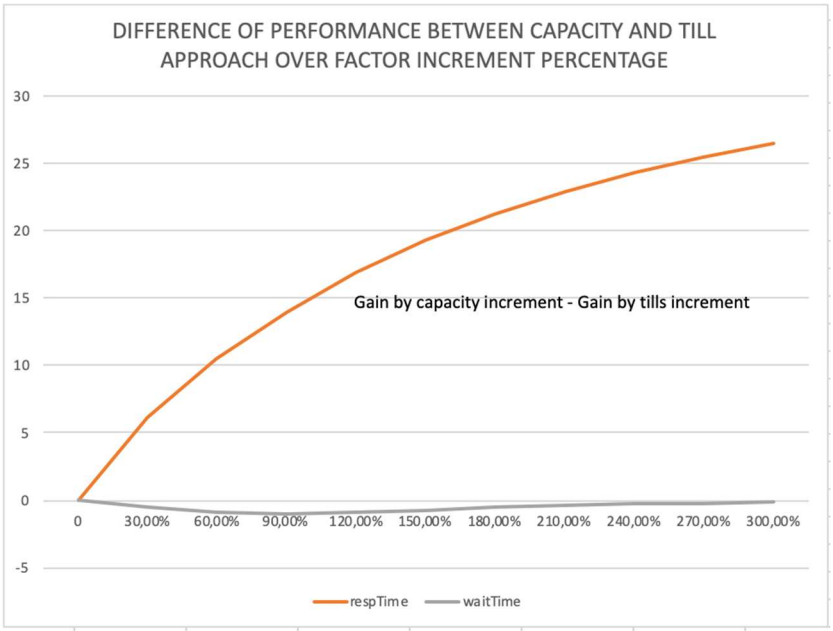
The number of customers in the supermarket is an important metric to keep an eye on, since even big supermarkets can't let a huge amount of people to occupy the tills' area. To find out which policy fits better different situations, we can take a look at the following graph: its left part tells us that with exponential service time, in a big supermarket in a high workload condition (the most critical one), P1 is really bad if the tills are not close to each other. P1 is also bad with a normal distribution and, under the same conditions, it doesn't matter which distribution we are going to pick for service times, as we can see from the central part of the graph. The situation changes a little when comparing P1 and P2 under a low workload and a lognormal distribution: P1 almost approximates P2, but the latter is always the best choice, if we want to minimize the number of customers in the supermarket tills' region.

4.4.3.1 A case when P1 is competitive

The only case in which P1 and P2 are really close in terms of number of customers, is when the service time is exponential, and we apply a low workload on a small market. Indeed,

in this case, the impact of Δ is really small, and P1 and P2 policies are almost interchangeable.

4.4.4 Comparison between increasing number of tills and upgrading capacity



This graph shows on the x axis the percentage of increment of capacity and number of tills, while on y axis the difference of increment of the same performance index due to the increase of the parameters. Let's make an example: looking at the graph we pick the point (60%,10): this means that increasing the percentage of number of tills and capacity by a 60% once at time we would get a gain of y for the first on and y+10 for the second one: computing the difference of them we get 10.

From this graph it's clear that increasing the capacity of some percentage affects the response time more than increasing the number of tills of the same percentage. This may be easily explained by observing that response time includes service time and increasing the tills won't reduce the response time under the service time. Indeed, waiting time, that doesn't take into account the service time, decreases practically the same if we increase N or C of the same percentage.

5. Conclusion

What is the best policy? As it always happens with statistics the answer is: "it depends". The first thing to note is the following:

P1 (delta = 1.5s) performances are always worse than P2's. This means that if tills aren't close, or simply the structure of the supermarket makes moving from a till to another difficult, then P1 should be avoided in any case.

Given this, what we need to compare is a well optimized P1 (e.g Δ=0.5s) and P2. These two scenarios are very close in terms of performances, meaning that customers will wait a similar amount of time on average. What is important is that P1 tends to be slightly better in the case of small supermarkets (as small we mean up to 5 tills), while P2 scales very well with the number of tills.

Talking about fairness, this assumes an important meaning in the high load scenario. As seen in the box plot, it turns out that if Δ is low, then P1 behaves quite well even with a high number of tills and guarantees (with 95% confidence) a much lower 98^{th} percentile. This means that with an optimized P1 customers may wait for a longer time in average, but no customer will experience a waiting time longer than a reasonable time. LCG results indeed confirm this theory, asserting that with P2, especially with a high number of tills, almost all the customers will find no queue at all, and a few others will wait for a very long time. These are the results and it's not our job to say what will make the costumer happier. The manager of the supermarket will take these results and decide, depending on his/her specific situation, what is the best policy to apply.