Delft University of Technology

FACULTY MECHANICAL, MARITIME AND MATERIALS ENGINEERING

Department Marine and Transport Technology

Mekelweg 2
2628 CD Delft
the Netherlands
Phone +31 (0)15-2782889
Fax +31 (0)15-2781397
www.mtt.tudelft.nl

Specialization: Production Engineering and Logistics

Report number: 2012.PEL.7660

Title: **Optimal occupation of the service spots in a fast food restaurant**

Author: J.J. Meijer

Title (in Dutch) Optimale bezetting van de service plekken in een fastfood restaurant

Delft University of Technology

FACULTY MECHANICAL, MARITIME AND MATERIALS ENGINEERING

Department Marine and Transport Technology

Mekelweg 2
2628 CD Delft
the Netherlands
Phone +31 (0)15-2782889
Fax +31 (0)15-2781397
www.mtt.tudelft.nl

Assignment: computer

Confidential: no

Initiator (university): Delft University of Technology

Supervisor: Dr. ir. H.P.M. Veeke

Date: April 3, 2012

Delft University of Technology

FACULTY OF MECHANICAL, MARITIME AND MATERIALS ENGINEERING

Department of Marine and Transport Technology

Mekelweg 2
2628 CD Delft
the Netherlands
Phone +31 (0)15-2782889
Fax +31 (0)15-2781397
www.mtt.tudelft.nl

Student:	J.J. Meijer	Assignment type:	Computer
Supervisor (TUD):	Dr. ir. H.P.M. Veeke	Creditpoints (EC):	10
		Specialization:	PEL
		Report number:	2012.PEL.7660
		Confidential:	No

Subject: **Optimal occupation of the service spots in a fast food restaurant**

We all know the fast food restaurants in the Netherlands. At those restaurants pace of the service is of great importance. This pace of service can be achieved by installing more cash registers or by using so called runners. The runners back the cashiers up and start collecting the order of the guest while it is still taken. Also a new kind of equipment is about to be installed, order kiosks. At these kiosks the guests can enter their orders themselves.

There is no prior research known, but due to your experience of working at such a fast food restaurant, you should be able to describe the processes well.

Your assignment is to simulate the processes in the restaurant, as well the service part as the kitchen, because this influences the service. With this simulation model you should be able to investigate whether it is beneficial to employ runners instead of extra cashiers. Also the advantages of installing order kiosks should be part of the research.

The report should comply with the guidelines of the section. Details can be found on the website.

The professor,

Dr. ir. H.P.M. Veeke

Summary

As the term fast food restaurant already suggests, pace of the service is of great importance. Besides the benefits of helping as many guests per hour as possible, the guest also expects to be helped quickly. This means that the experience time of the guest, the time between arrival and having the food, should be as low as possible.

There are different ways to lower the experience time. One way to reduce the experience time is to deploy a so called runner. This runner starts collecting the products while the order is still taken by the cashier. Another, upcoming, way to bring the experience time to a minimum is the use of order kiosks. At such a kiosk the guests enter and pay the order themselves, where after they get in a line to get their order that is collected by employees.

Case studies, based on a simulation model written in Delphi, are performed and compared to an initial situation. With the precondition of using the same number of employees, the implementation of runners and order kiosks are compared with the initial situation with only cashiers. The experience time is used to compare the results. In the simulation model the turnover per hour, average spending and number of cash registers, runners and kiosks are input variables. The outputs are the experience times of the guests.

Two cases are performed. The first case with a turnover per hour of €3000.- and an average spending of €10.- per guest, with order and collect times of five seconds per product each. The second case has an hourly turnover of €1500.- and an average spending of €10.- per guest. The collect times of the products are set to ten seconds per product. The cases show great influences from the number of cashiers, runners and order kiosks used.

A first conclusion from the performed cases is that it is beneficial to employ runners, but to a certain level. If the kiosks are excluded, the first case has the best result with two runners and five cashiers, instead of only seven cashiers. An improvement of 24.7% for the average experience time is realized. With more than two runners the results deteriorate again. The second case shows an improvement of 12.5% for the average experience time when one runner and three cashiers are employed instead of just four cashiers.

The next conclusion that can be drawn from the first case is the advantage of using order kiosks. The runners are able to choose between helping a cashier and help collecting products at a pick up point. When again seven employees are working in the service, the best occupation of the service spots is: three cashiers, three runners and one pick up point with two order kiosks. The average experience time then improves with 32.9% compared to the situation with only seven cashiers.

So without using more employees an improvement in average experience time can be made by implementing runners. No investments in equipment are necessary, just the training of these employees should be extended. If there is a will to invest, even further improvements can be made by installing order kiosks.

Summary (in Dutch)

Zoals de term fastfood restaurant al doet vermoeden is de snelheid van de service erg belangrijk. Naast de voordelen van het helpen van zoveel mogelijk gasten per uur als mogelijk, verwacht de gast het ook om snel geholpen te worden. Dit betekent dat de ervaringstijd van de gast, de tijd tussen binnenkomst en het hebben van het eten, zo laag mogelijk zou moeten zijn.

Er zijn verschillende manieren om de ervaringstijd omlaag te brengen. Een manier om de ervaringstijd te verkorten is door een zogenaamde runner in te zetten. Deze runner begint al met het verzamelen van de producten terwijl de order nog opgenomen wordt door de kassier. Een andere, opkomende, manier om de ervaringstijd naar een minimum te brengen is door het gebruik van een orderkiosk. Bij zo'n kiosk voeren de gasten de order zelf in en betalen, waarna ze in de rij gaan staan om hun order, die verzameld is door medewerkers, op te halen.

Case studies, gebaseerd op een simulatiemodel geschreven in Delphi, zijn uitgevoerd en vergeleken met een aanvangssituatie. Met de randvoorwaarde om hetzelfde aantal werknemers te gebruiken, is het gebruikmaken van runners en orderkiosken vergeleken met de aanvangssituatie met alleen maar kassiers. De ervaringstijd is gebruikt om de resultaten te vergelijken. In het simulatiemodel zijn de omzet per uur, de gemiddelde besteding en het aantal kassa's, runners en kiosken invoervariabelen. De ervaringstijden van de gasten zijn de uitkomsten.

Twee casussen zijn uitgevoerd. De eerste met een omzet per uur van €3000,- en een gemiddelde besteding van €10,- per gast, met bestel- en verzameltijden van vijf seconden per product elk. De tweede casus heeft een omzet van €1500,- per uur en een gemiddelde besteding van €10,- per gast. The verzameltijden van de producten zijn op tien seconden per product gezet. De casussen tonen grote invloeden van het gebruikte aantal kassiers, runners en orderkiosken.

Een eerste conclusie van de uitgevoerde casussen is dat het bevorderlijk is om runners in te zetten, maar tot een bepaald niveau. Als de kiosken buiten beschouwing worden gelaten, heeft de eerste casus het beste resultaat met twee runners en vijf kassiers, in plaats van met alleen zeven kassiers. Een verbetering van 24,7% voor de gemiddelde ervaringstijd is gerealiseerd. Met meer dan twee runners verslechteren de resultaten weer. De tweede casus laat een verbetering zien van 12,5% voor de gemiddelde ervaringstijd wanneer één runner en drie kassiers aan het werk zijn in plaats van alleen vier kassiers.

De volgende conclusie die getrokken kan worden uit de eerste casus is het voordeel van het gebruik van orderkiosken. De runners hebben de mogelijkheid te kiezen tussen het helpen van een kassier of het helpen verzamelen van producten bij een pick-up point. Als er weer zeven werknemers werkzaam zijn in de service, is de beste bezetting van de punten in de service: drie kassiers, drie runners en één pick-up point met twee orderkiosken. De gemiddelde ervaringstijd verbetert dan met 32,9% vergeleken met de situatie met alleen zeven kassiers.

Dus zonder het gebruik van meer werknemers kan er een verbetering in gemiddelde ervaringstijd gemaakt worden door gebruik te maken van runners. Er zijn geen investeringen in apparatuur nodig, alleen de training van deze werknemers zal uitgebreid moeten worden. Als er een wil om te investeren is, zijn er nog verdere verbeteringen mogelijk door orderkiosken te installeren.

List of Figures

Contents

1. Introduction

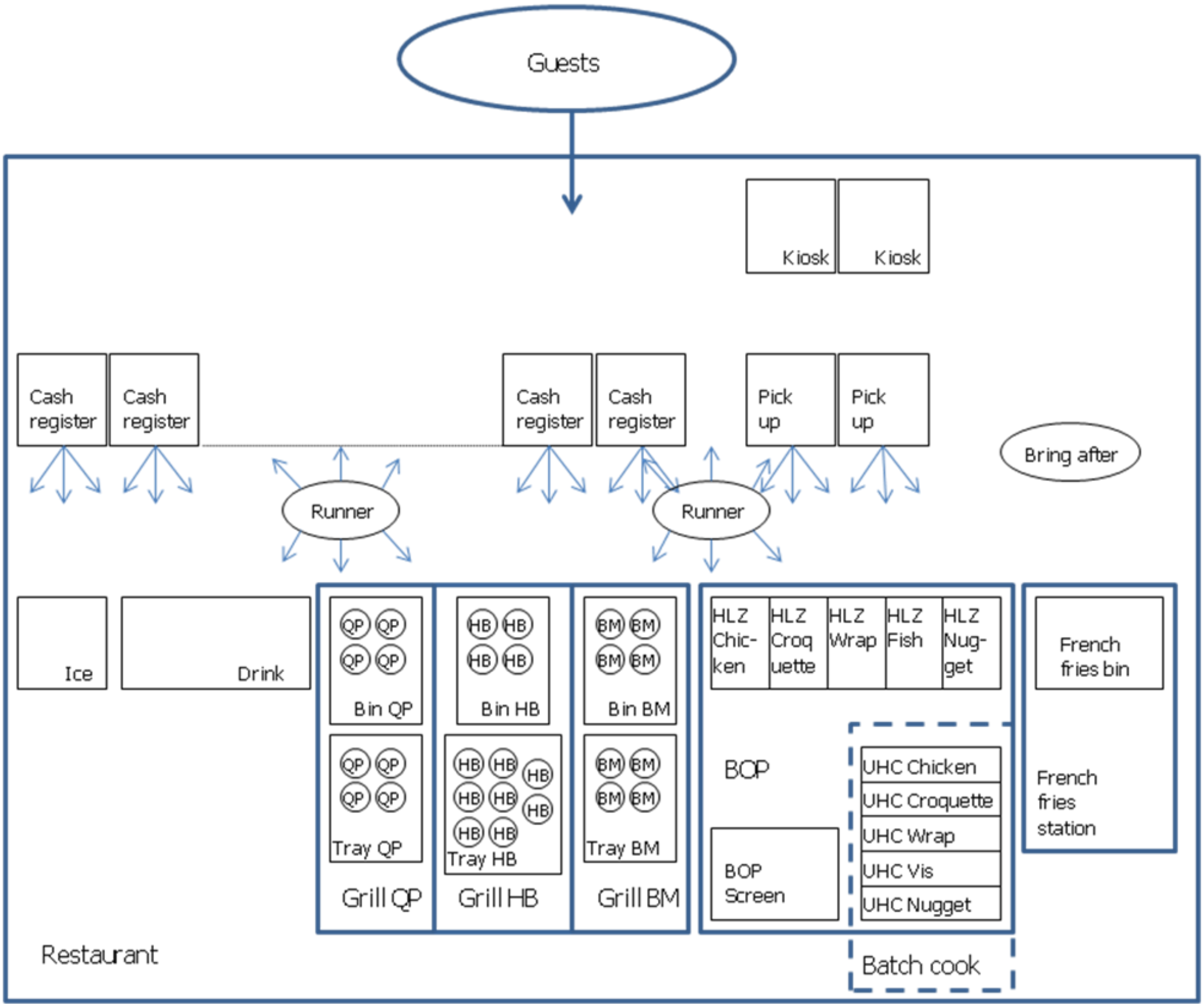
In fast food restaurants as there are in the Netherlands, pace of the service is of great importance. As the term fast food suggests, guests expect a quick meal. Although the slogan of the restaurants changed in the past years from 'serve fast food good' into 'serve good food fast', the speed of helping guests is still a key issue. This means that the experience time of the guest, the time between arrival and having the food, should be as low as possible. From the point of view of the restaurant the aim is to help as many guests as possible, to enlarge the sales volume. From the point of view of the guest it is believed he wants to be helped quickly. There are different ways to lower the experience time. The restaurant could of course decide to open more cash registers, but there is a limit to that in terms of available space and wages, each cash register has to be manned by a cashier. Another way to reduce the experience time is to deploy a so called runner. This runner starts collecting the products while the order still is taken by the cashier. The consideration here lies again in the wages. Yet a third and upcoming way to bring the experience time to a minimum is the use of order kiosks. At such a kiosk the guests enter and pay the order themselves, where after they get in a line to get their ordered products that are collected by employees.
In this report two ways to lower the experience time are compared using case studies. With the precondition of using the same number of employees, the implementation of runners and order kiosks are compared with the initial situation with only cashiers. Also the service time, the time from start helping a guest to greeting the guest goodbye with a completed order, is taken into account. So the performance indicators are the experience time and the service time, which compare to throughput time and handling time respectively. The experience time is used to compare the results, the service time can be used to compare the results with reality, because this is a number that can easily be obtained in the real world, which is already the case. To make the comparison a simulation is built in Delphi. In this simulation the turnover per hour, average spending and number of cash registers, runners and kiosks are input variables. The outputs are the experience times and service times of the guests and averages of those. To create an accurate simulation model also the processes of the kitchen are simulated. These are simulated to enable more realistic behavior of the cashiers and runners, because it takes time to produce the products. The goal of simulating the kitchen is therefore not to create an exact model of the kitchen, but will however give indications of the waste percentage and a percentage of the products that took too long to produce and have been brought to the table of the guest.

The main research question is: given a certain turnover per hour and average spending, how is the experience time influenced by employing more or less cash registers, more or less runners and more or less order kiosks?

In chapter two the processes of the several elements in the restaurant are described, followed in chapter three by a description of the simulation with a process description language. Chapter four and five describe the two performed case studies and the conclusions can be found in chapter six.

2. Description Processes Restaurant

In this chapter the processes of guests entering the restaurant, placing their order and leaving satisfied, are defined. Also the several processes that enable the guests to meet their wishes are described. In 2-1: Schematic representation restaurant, a representation of the different processes in the restaurant is shown.

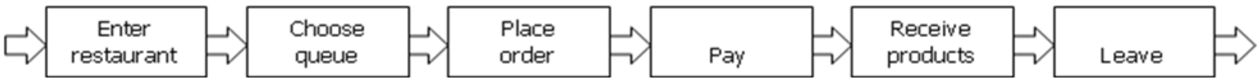


2-1: Schematic representation restaurant

Beneath the processes of the different elements in the above figure are described one by one. Because this report describes a simulation and thus just a model of reality, it is not intended to be complete concerning the number of products. In reality more different products can be chosen from, but it appeared that to get to a satisfying result, the different products could be narrowed down to: ice cream, drink, QP, HB, BM, chicken sandwich, croquette, wrap, fish, nuggets and French fries. Remember that the aim of this research is the service and the division over: number of cashiers, runners and pick up points and how that influences the experience time and service time. More products would not have a significant impact on that, but do have their influence on the waste percentage and the percentage of products that have to be brought to the table of the guest.

Guest

A guest enters the restaurant. He or she will choose the cash register or kiosk with the shortest waiting queue. When it is his or her turn, the guest will place the order of products that are wanted. There are two cases. The first case is if the guest chose a cash register, the second is the case of a kiosk chosen. In the first case the order will be entered by a cashier. After payment the order will be collected by the cashier and when ready, the guest chooses a table to consume his meal. In the second case the guest enters the order himself. After he or she has paid, the guest gets a receipt with a number and gets in the line of a pick up point. At this point the guest picks up his ordered products when they are ready and leaves for a table to enjoy his meal. A schematic representation of this process is shown below.

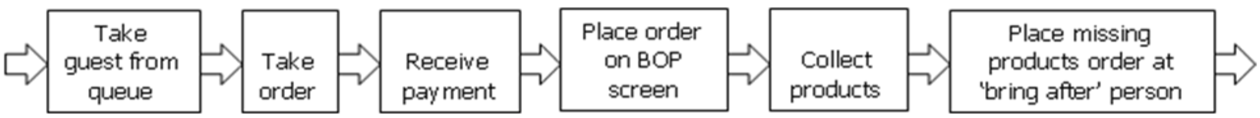


2-2: Function model of a guest

Cash register

Each occupied cash register has a dedicated cashier. It is actually the cashier that runs through a process. The cashier gets the first guest out of the queue of the cash register. He asks the guest what he would like to have and takes his order. If the order contains BOP products (chicken sandwich, croquette, wrap, fish, nuggets), this product order is automatically placed on the BOP screen.

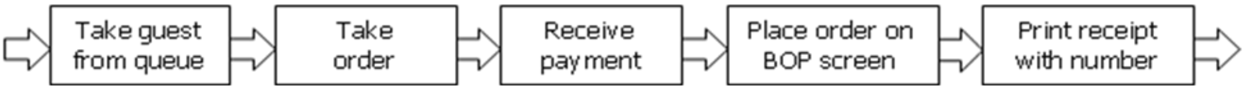
The cashier receives the payment and collects the products the guest asked for. This could be an ice cream, a drink, fries, a QP, HB or BM, or a chicken sandwich, croquette, wrap, fish or nuggets. A drink or ice cream is prepared by the cashier himself, a QP, HB, BM or French fries are collected from a bin. The same for the chicken, croquette, wrap, fish and nuggets, but here the bin is called a heated landing zone, or HLZ. The stations behind the bins are responsible for the supply of the bins and these processes are described below. When it happens that a product is not in the bin, an order is placed at the 'bring after' person. This person brings the missing product to the table of the guest as soon as it is ready. A schematic representation of this process is shown below.



2-3: Function model of a cashier

Order kiosk

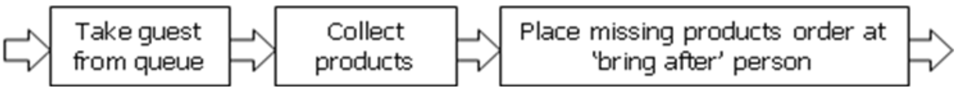
The order kiosk is a new concept that is only implemented in a few fast food restaurants. It is an unmanned cash register where the guest enters his order and pays himself. If the order contains a BOP product it is, after payment, placed on the BOP screen. The kiosk prints a receipt with a unique number for the guest, with which the guest can get his meal from the pick up point. A schematic representation of this process is shown in figure 2-4.



2-4: Function model of a kiosk

Pick up point

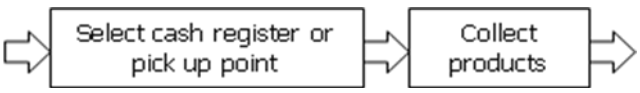
A pick up point is manned by a dedicated employee. However, there have not necessarily to be as many pick up points as kiosks. For instance, there could be two kiosks and only one pick up point. The employee that runs the pick up point gets the number of the first guest in the queue and the order of that guest appears on the screen. From this point on the process of this employee of collecting the products is the same as for the cashier, so preparing a drink or ice cream or collecting the QP, HB, BM, chicken sandwich, croquette, wrap, fish, nuggets or French fries. Again, if a product is not ready a new order is placed at the 'bring after' person. A schematic representation of this process is shown below.



2-5: Function model of a pick up point

Runner

The so called runner has multiple tasks. He helps where most needed. This means that he chooses the cash register with the largest amount of products that still have to be collected. He can start collecting those items while the order is still been taken. There can be as many runners as running cash registers. The runner can also help collecting products for the pick up point. A schematic representation of this process is shown below.



2-6: Function model of a runner

Bring after

The bring after person is in reality not constantly busy with this task. It often is a subtask of a host or lobby person. The orders of missing products are placed in sight and as soon as the product is ready it is brought to the table of the guest. It is an objective to bring the number of products that have to be brought later to a minimum and when it happens to keep the waiting time as low as possible. A schematic representation of this process is shown below.

2-7: Function model of a bring after person

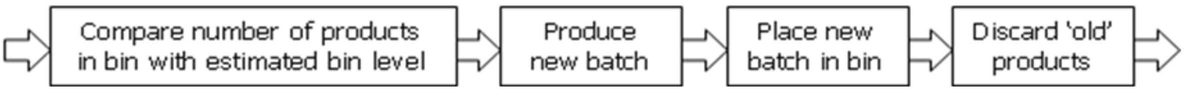
French fries station

The French fries station is responsible for the supply French fries. It has to estimate how many portions are needed and keep its bin filled. The fries are fried in oil and after a set period the fries are put in a heated bin and salted. The French fries are kept hot in the bin, but not for longer than five minutes. After expiration of this period the fries are discarded and put in a waste bin. It takes a bit of experience to get the estimation right, but due to the low cost of the fries, the French fries bin should always be filled. A schematic representation of this process is shown below.

2-8: Function model the French fries station

Grill

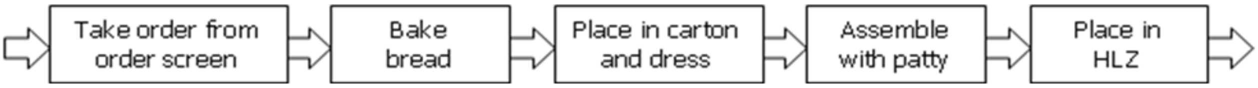
The process of the grill is roughly the same for QP, HB and BM, the main difference is the product that is made. From historical data a certain bin level for each product is derived. The number of products in the bin is compared with the estimated bin level and when this number is lower, a new batch is produced. The sandwiches are kept warm for ten minutes in the bin, after this they are discarded. A schematic representation of this process is shown below.



2-9: Function model of the grill

BOP

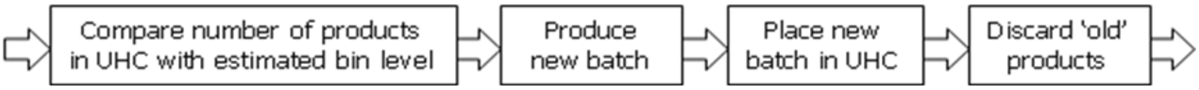
The term BOP comes from the founder of the system, it is short for Bridge Operating Platform. In contrary to the French fries station and the grill it is mostly a pull system instead of a push system and thus the client order entry point lies further up the production line. As described in the processes of the cash register and kiosk, the BOP product orders are placed on the BOP screen. When an order appears on the screen, an employee starts producing the product. If applicable he roasts the bread and assembles the package and the patty. The patty is gained from a holding cabinet called universal holding cabinet or UHC. The finished product is placed in the HLZ to be collected by a cashier, runner or other employee to serve the guest. A schematic representation of this process is shown below.



2-10: Function model of BOP

Batch cook

The batch cook is responsible for the supply of the UHC. From historical data a certain level is determined. A two, three or four bin system is used. When, in the case of a two bin system for example, the first tray is empty, half of the derived UHC level of patties is produced. This way there is always stock. The patties are kept hot in the UHC for twenty minutes and in case of a croquette patty for thirty minutes. After expiring of this period the patties are removed and placed in a waste bin. A schematic representation of this process is shown below.



2-11: Function model of batch cook

3. Simulation

In order to come to an answer of the main research question, a simulation model is built. Recall that the experience and service time dependency of the number of cashiers, kiosks and runners is asked, given a planned turnover and average spending. There are a lot of variables that have to be taken into account and stochastic is important for realism. A first estimation can be made by averages and this can be used to verify the results of the simulation model. In this chapter the processes as described in chapter two are translated to a process description language, or pdl. In appendix A the simulation code can be found.

The process description language is written based on the processes as described in the previous chapter. Terms like 'hold' and 'standby' are printed bold faced, because these are terms that are also used in the actual code and represent a duration of time, a period where in the real world is done something. A guest generator is modeled to simulate the arrival of guests. Here is also decided how much the guest will spend.

Guest generator

The guest generator has to wait until the arrival of the next guest. This waiting time is determined by getting a sample from a uniform distribution. The average of this distribution is 3600 seconds times the average spending of a guest, divided by the planned turnover. The number of products this new guest will order is sampled from another distribution with an average of the average spending, which is an input variable and can in the real world be derived from historical data. So the pdl of the guest generator is:

GuestGenerator.*process*
    Repeat
        Hold (interArrivalTime.*sample*)
        NewGuest = Guest.*create*
        NewGuest.NumberOfProducts = NumberOfProducts.*sample*
        Choose queue //this is a function and returns the shortest waiting queue
        Queue.*AddtoTail*(NewGuest)

The function 'choose queue' decides which cash register queue is shortest. Then it decides which kiosk queue is shortest. Those two are compared and the function returns the one with the shortest length.

Cashier

The cashier start with taking the first arrived guest from the queue. The cashier takes the order from the guest. In this process is determined what kind of product is ordered. This is sampled from a distribution. The division over the sold products can in the real world be derived from historical data.

After the payment, the cashier collects the ordered products. If a certain product is not ready, the order of that product goes to the 'bring after' person, who will bring the product to the table of the guest. It will however take some time for the cashier to find out the product is not ready, so whether a product is ready or not, he has to work the collect time of the product. To enable the runner in the simulation model to help the cashier, the cashier has to enter a virtual 'WantHelp' queue, further explanation is found in the process of the runner.

```
Cashier.process
    Repeat
        While MyQueue.Length = 0 Standby
        MyGuest = MyQueue.FirstElement
        Enterqueue(WantHelp)
        For MyGuest.NumberOfProducts Repeat
            NewProduct = ProductType.Sample  //here the kind of product is decided
            NewProduct = Product.Create
            Hold (NewProduct.OrderTime)
            If NewProduct = BOPproduct then BOPscreen.AddtoTail(NewProduct)
            MyGuest.Order.AddToTail(NewProduct)
        Hold (MyGuest.PayTime)
        Leavequeue(WantHelp)
        For MyGuest.Order.Length Repeat
            MyProduct = MyGuest.Order.FirstElement
            Order.Remove(MyProduct)
            If MyProduct = FriesProduct  then
                If FriesBin.Length = 0 then BringAfter.AddToTail(MyProduct)
                Else FriesBin.Remove(FriesProduct)
            If MyProduct = GrillProduct  then
                If Bin.Length = 0 then BringAfter.AddToTail(MyProduct)
                Else Bin.Remove(GrillProduct)
            If MyProduct = BOPproduct  then
                If HLZ.Length = 0 then BringAfter.AddToTail(MyProduct)
                Else HLZ.Remove(BOPproduct)
            Hold (MyProduct.CollectTime)
        MyQueue.Remove(MyGuest)
```

Kiosk

The kiosk has nearly the same process as the first part of the cashier its process. The orders that are needed from BOP are however placed on the BOP screen after the guest has paid. After this the guest is placed in the queue of a pick up point. The shortest is chosen.

```
Kiosk.process
    Repeat
        While MyQueue.Length = 0 Standby
        MyGuest = MyQueue.FirstElement
        For MyGuest.NumberOfProducts Repeat
            NewProduct = ProductType.Sample  //here the kind of product is decided
            NewProduct = Product.Create
            Hold (NewProduct.OrderTime)
            MyGuest.Order.AddToTail(NewProduct)
        Hold (MyGuest.PayTime)
        MyProduct = MyGuest.Order.FirstElement
        For MyGuest.Order.Length Repeat
            If MyProduct = BOPproduct then BOPscreen.AddtoTail(NewProduct)
            MyProduct = MyGuest.Order.NextElement
        MyQueue.Remove(MyGuest)
        Choose pick up queue  //this is a function and returns the shortest pick up queue
```

Pick up point

The employee manning the pick up point has the process as described below. Again, like in the process of the cashier, to enable the runner in the simulation model to help the employee of the pick up point, the employee has to enter a virtual 'WantHelp' queue and leave it when help is not wanted anymore.

```
PickUp.process
    Repeat
        While MyQueue.Length = 0 Standby
        MyGuest = MyQueue.FirstElement
        Enterqueue(WantHelp)
        For MyGuest.Order.Length Repeat
            MyProduct = MyGuest.Order.FirstElement
            Order.Remove(MyProduct)
            If MyProduct = FriesProduct  then
                If FriesBin.Length = 0 then BringAfter.AddToTail(MyProduct)
                Else FriesBin.Remove(FriesProduct)
            If MyProduct = GrillProduct  then
                If Bin.Length = 0 then BringAfter.AddToTail(MyProduct)
                Else Bin.Remove(GrillProduct)
            If MyProduct = BOPproduct  then
                If HLZ.Length = 0 then BringAfter.AddToTail(MyProduct)
                Else HLZ.Remove(BOPproduct)
            Hold (MyProduct.CollectTime)
```

```
        Leavequeue(WantHelp)
        MyQueue.Remove(MyGuest)
```

Runner

The runner has to choose which cashier or pick up point he will help. He does this by comparing the order sizes. Once he has chosen a cashier to help, he continues helping him until the cashier is ready himself to collect the products, so until the guest has paid. If the runner chose a pick up point to collect products for, the runner continues helping this employee until the order is totally collected.

```
Runner.Process
    Repeat
        While WantHelp.Length = 0 Standby
        Choose cashier or pick up point  //function to choose the greatest order
        While Cashier or PickUp is in WantHelp repeat
            MyGuest = Cashier.MyGuest or PickUp.MyGuest
            MyProduct = MyGuest.Order.FirstElement
            Order.Remove(MyProduct)
            If MyProduct = FriesProduct  then
                If FriesBin.Length = 0 then BringAfter.AddToTail(MyProduct)
                Else FriesBin.Remove(FriesProduct)
            If MyProduct = GrillProduct  then
                If Bin.Length = 0 then BringAfter.AddToTail(MyProduct)
                Else Bin.Remove(GrillProduct)
            If MyProduct = BOPproduct  then
                If HLZ.Length = 0 then BringAfter.AddToTail(MyProduct)
                Else HLZ.Remove(BOPproduct)
            Hold (MyProduct.CollectTime)
```

Bring after

As mentioned in chapter two, the bring after action is actually a subtask of a host or other employee. The reason to model this function in the simulation model is to create a more realistic scenario in the case a product is missing in the bin, HLZ or French fries bin. In the simulation the bring after person has no other tasks, so is always first to get a sandwich from the bin or HLZ or French fries from the fries bin. This way the first product entering the stock cannot be given to another guest. The cashier, pick up point employee or runner places orders in the bring after queue called 'BringAfter'. The process is as follows:

```
BringAfterPerson.Process
    Repeat
        While BringAfter.Length = 0 Standby
        MyProduct = BringAfter.FirstElement
```

```
        If MyProduct = FriesProduct  then
            While FriesBin.Length = 0 Standby
            FriesBin.Remove(FriesProduct)
        If MyProduct = GrillProduct  then
            While Bin.Length = 0 Standby
            Bin.Remove(GrillProduct)
        If MyProduct = BOPproduct  then
            While HLZ.Length = 0 Standby
            HLZ.Remove(BOPproduct)
        Hold (MyProduct.BringAfterTime)
```

French fries station

The French fries station is responsible for keeping the 'FriesBin' to a certain level, here called 'FriesLevel'. In order to model the keeping life, the French fries have a process of their own, which is shown below the process of the French fries station.

```
FrenchFriesStation.Process
    Repeat
        If FriesBin.Length < FriesLevel then
            NewFries = Fries.Create
            Hold(NewFries.CookingTime)
            NewFries.Start
            FriesBin.AddToTail(NewFries)
        Else Standby
```

```
Fries.Process
    Hold(KeepingLife)
    If IsinQueue(FriesBin) then
        LeaveQueue(FriesBin)
        EnterQueue(FriesWasteBin)
    Stop
```

Grill

The grill is responsible for preparing BMs, QPs and HBs. In the simulation model there are three separate processes, each dedicated to their own product. The processes are however rather the same, so below is just one of these processes shown. From historical data a bin level for the separate products is derived. When the number of products in the bin gets below the derived bin level, a new batch of that product is produced. The products are prepared on a tray. To model the keeping life of the product, the product has a process itself, shown below the process of the grill. Again, for similarity, only one of these processes is shown.

```
Grill.Process
    Repeat
        If Bin.Length < BinLevel then
            Tray = TomasQueue.Create
            For I = 1 to BatchSize do
                NewBM = BM.Create  //the same for QP and HB
                Tray.AddToTail(NewBM)
            Hold(CookingTime)
            For I = 1 to Tray.Length do
                MyBM = Tray.FirstElement
                MyBM.Start
                Bin.AddToTail(MyBM)
        Else Standby
```

```
BM/QP/HB.Process
    Hold(KeepingLife)
    If IsinQueue(Bin) then
        LeaveQueue(Bin)
        EnterQueue(WasteBin)
    Stop
```

BOP

BOP prepares the fried products other than the French fries. In this simulation model five products are distinguished: chicken sandwich, croquette, wrap, fish and nuggets. The orders for BOP are placed by the cashier or kiosk on the BOP screen. Once an order is on the screen, the process of preparing the product is initiated. The product then has its own process in this simulation model, therefore the processes of the different products are shown below the process of BOP. So the process of BOP is just the initiator of different processes.

```
BOP.Process
    Repeat
        While BOPscreen.Length = 0 Standby
        MyProduct = BOPscreen.FirstElement
        BOPscreen.Remove(MyProduct)
        If MyProduct = Chicken then    //the same for croquette, wrap, fish and nuggets
            NewChicken = Chicken.Create
            NewChicken.Start
```

The sandwiches themselves make sure they start only when a patty of the right kind is in the UHC and enter the HLZ after the preparation time. The process is shown for chicken, but is also true for croquette, wrap, fish and nuggets.

```
Chicken.Process
    While UHCchicken = 0 Standby
    MyPatty = UHCchicken.FirstElement
    UHCchicken.Remove(MyPatty)
    Hold(PreparationTime)
    EnterQueue(HLZ)
    Stop
```

Batch cook

In the process directly above can be seen that the process will only start when the right patty is in the UHC. Batch cook is responsible for the supply of these patties. Like the bin level of the grill, also from historical data an UHC level is derived. The patties have a keeping life and after expiration of this they enter a waste bin. Therefore the patties have a process of their own, shown below the process of batch cook.

```
BatchCook.Process
    Repeat
        If UHCchicken.Length < UHClevel then  //the same for croquette, wrap, fish and
nuggets
            For I = 1 to (UHClevel – UCHChicken.Length) do
                NewChickenPatty = ChickenPatty.Create
                Pan.AddToTail(NewChickenPatty)
            Hold(CookingTime)
            For I = 1 to Pan.Length do
                MyChickenPatty = Pan.FirstElement
                MyChickenPatty.Start
        Else Standby
```

The process is shown for a chicken patty, but is the same for croquette, wrap, fish and nuggets patties.

```
ChickenPatty.Process
    Hold(KeepingTime)
    If IsInQueue(UHCchicken) then
        LeaveQueue(UHCchicken)
        EnterQueue(WasteBin)
    Stop
```

4. Case Study 1

In this chapter a case study is performed to get answers to the research questions. First a case is set out, input variables are determined and described in the first paragraph. The second part of this chapter consists of a calculation based on averages which is compared with the results of simulation runs. Paragraph three describes a simulation run with only seven cashiers, while in the fourth paragraph the input variables are the same, but six cashiers and one runner are employed. Paragraph five handles the case of five cashiers and two runners. The sixth paragraph deals with the case of four cashiers and three runners. To compare these results with the use of kiosks in paragraph seven the case of five cashiers, zero runners, two kiosks and two pick up points is described. The case of using four cashiers, two runners, two pick up points and one kiosk and the case of three cashiers, three runners, two kiosks and one pick up point are described in paragraph eight and nine, respectively. The latter two cases show the most beneficial results. More cases are dealt with and all the results can be found in appendix B. Finally, in the tenth paragraph the results are compared.

4.1 Input variables

The input variables are shown in figure 4-1. The explanation of the variables can be found direct after the figure. All the input variables are kept the same for this case study. They are input variables because they can be changed to get realistic behavior of the simulation model. These input variables will differ for different restaurants, but should be kept the same once a realistic scenario is found. After this only the planned turnover and average spending should be varied and of course the number of cashiers, runners, kiosks and pick up points to get to an optimal occupation.

Planned turnover per hour (€)	3000.-
Average spending (€)	10.-

Product	**BM**	**HB**	**QP**	**Chicken**	**Croquette**	**Fish**
Price (€)	3.35	1.00	3.35	3.35	2.10	2.00
Division	8	16	4	6	4	1
Order time (s)	5	5	5	5	5	5
Collect time (s)	5	5	5	5	5	5
Preparation time (s)	90	90	150	30	30	30
System	2	2	2			

Product	**Wrap**	**Nuggets**	**Drink**	**Ice**	**Fries**
Price (€)	2.00	2.00	2.00	2.35	0.80
Division	2	12	23	8	15
Order time (s)	5	5	5	5	5
Collect time (s)	5	5	3	3	5
Preparation time (s)	20	10	7	12	5

4-1: Input variables

Planned turnover per hour: The input of planned turnover per hour is in euros. A planning is always made in the restaurants and is based on historical data, weather forecast, holiday period and commercial promotions. Together with the average spending this number is the input for a uniform distribution that decides the inter arrival time of guests.

Average spending: This is the average a guest will spend. This number will differ per restaurant and is based on historical data. The simulation model uses this number as average for a uniform distribution. The number of products a guest orders is derived from this distribution.

Price: Here the price per product is put in. There are advised and regulated prices.

Division: This is the division over the products. In the example are eight BMs, 16 HBs, four QPs, six chickens, four croquettes, two wraps, twelve portions of nuggets, 23 drinks, eight ice creams and 15 portions of French fries sold for every fish. It is about the ratio, not the exact number. This numbers can be derived from historical data.

Order time: This is the time in seconds it takes to order this product. This number depends on the experience of the crew and the type of guests.

Collect time: The number of seconds it takes to collect a product.

Preparation time: This number is particularly important for the modeling of the kitchen. For the drinks and ice creams however, this number directly influences the service employee, because he has to prepare that product himself.

System: Only of interest for the grill, so only for BM, HB and QP a number is given. In the real world this number defines how many employees are dedicated to that product. So if the system of BM is set to two, two employees are making BMs. In the simulation model the preparation time is simply divided by this number.

For realism the number of cashiers is kept to a maximum of nine, the maximum number of kiosks is two and therefore the maximum number of pick up points is also two.

4.2 Averages calculation

In the case a planned turnover of € 3000.- per hour is proposed with an average spending of € 10.- per guest. This means on average 300 guests per hour will arrive.

The average price of a product is calculated by the total number of products divided by the sum of the division of the product times their price. The average number of products per guest can be derived by dividing the average spending by the average price. The formula for the average number of products in this case is:

$$\frac{10\cdot(8+16+4+6+4+1+2+12+23+8+15)}{8\cdot 3.35+16\cdot 1.00+4\cdot 3.35+6\cdot 3.35+4\cdot 2.10+1\cdot 2.00+2\cdot 2.00+12\cdot 2.00+23\cdot 2.00+8\cdot 2.35+15\cdot 0.80}=$$

$=\dfrac{10\cdot 99}{171.5}=5.77$, so on average 5.77 products are ordered per guest.

It takes five seconds per product to order. The employee will use five seconds per product to collect it. This seems a bit short, but in reality multiple products are collected at once. Drinks and ice creams also have a preparation time, so in 23 out of 99 cases the collect time is not five, but ten seconds and in eight out of 99 cases the time to collect the product is 15 instead of five seconds. In the other 66 cases the time to collect the product is the given five seconds. So the average time to collect a product is:

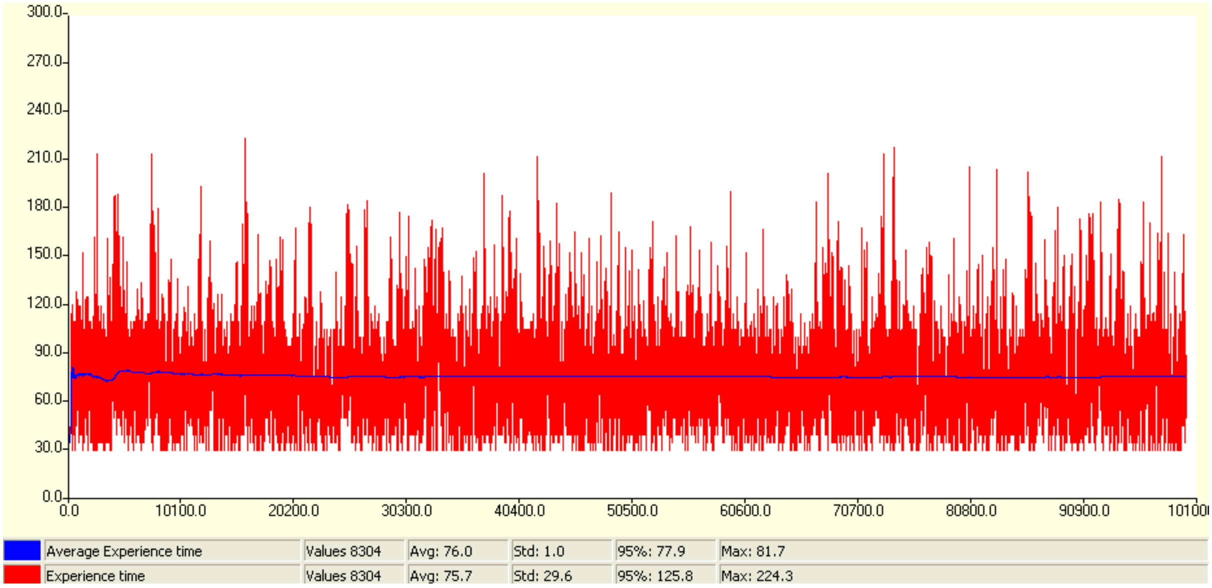
$\dfrac{66\cdot 5+23\cdot 10+8\cdot 15}{99}=6.87$ seconds.

The time the payment takes is set to ten seconds. Thus, the average service time per guest is the number of products times the time to order, plus the number of products times the average time to collect, plus the time to pay, this is: $5.77\cdot 5+5.77\cdot 6.87+10=78.5$ seconds. This is the average service time.

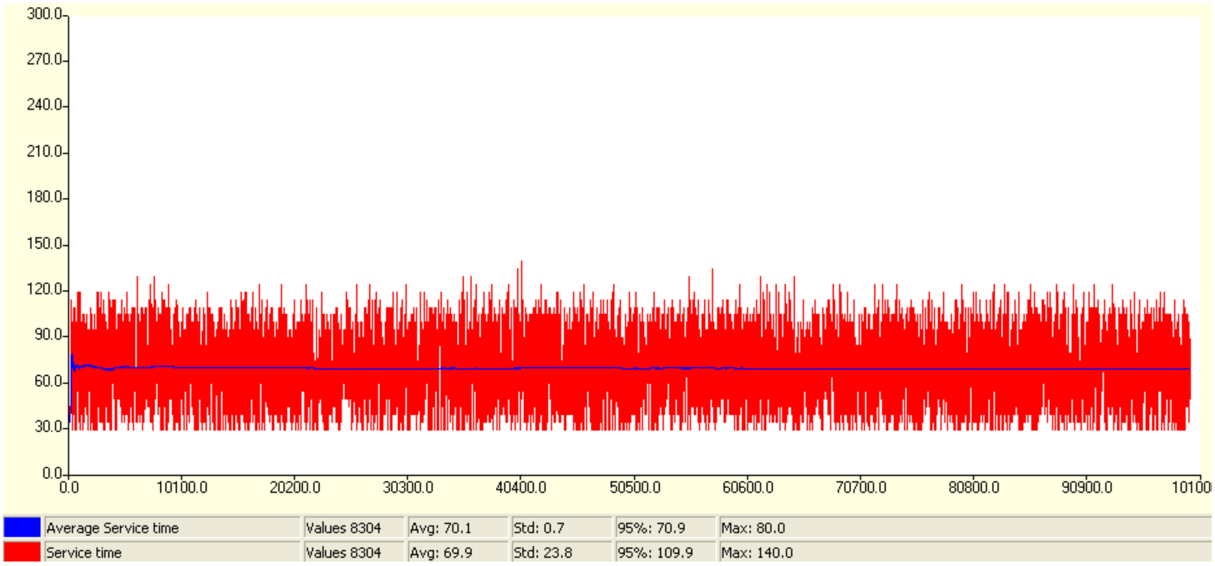
If there are 300 guests per hour, it means that one guest should be helped every $3600/300=12$ seconds. Recalling that helping a guest takes 78.5 seconds on average, at least 6.5 cashiers should be helping guests.

4.3 Seven cashiers

In this paragraph the results of the simulation run with seven cashiers are shown. No runners, kiosks and thus no pick up points are used. This is the initial situation. The input variables are given in paragraph one of this chapter, in figure 4-1. Figure 4-2 shows the experience time per guest in red and the average experience time in blue. The number of seconds of experience time is on the y-axis, the x-axis is the time in the simulation.



4-2: Experience time 7 cashiers, 0 runners



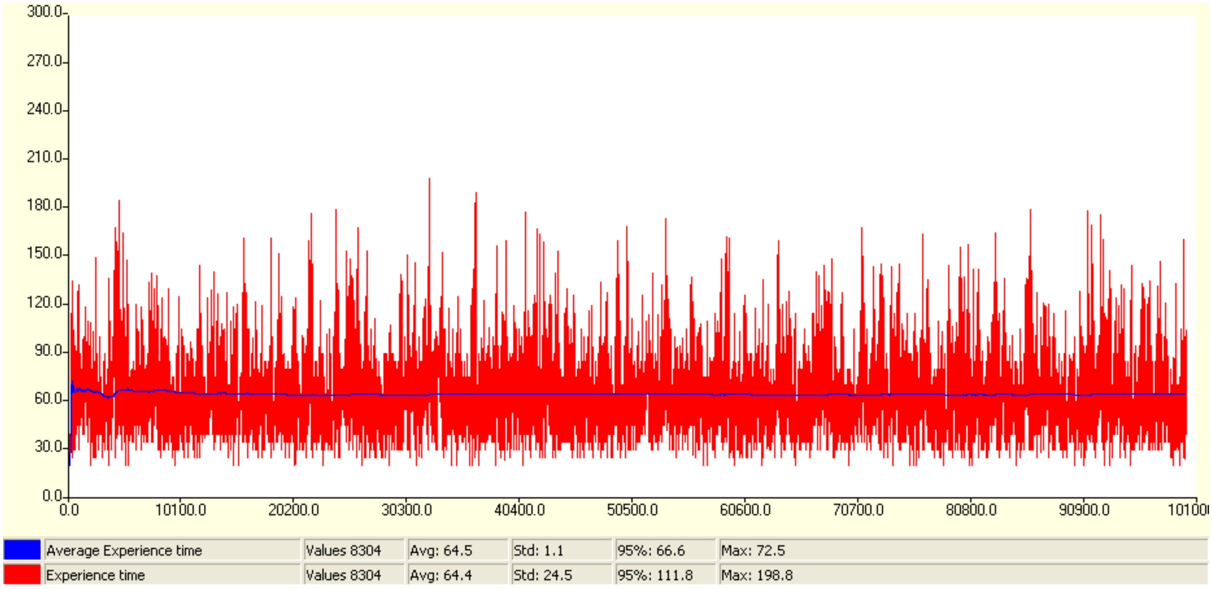
4-3: Service time 7 cashiers, 0 runners

The average experience time is 75.7 seconds, with 95% of the values lying beneath 125.8 seconds. The maximum peak is at 224.3 seconds.

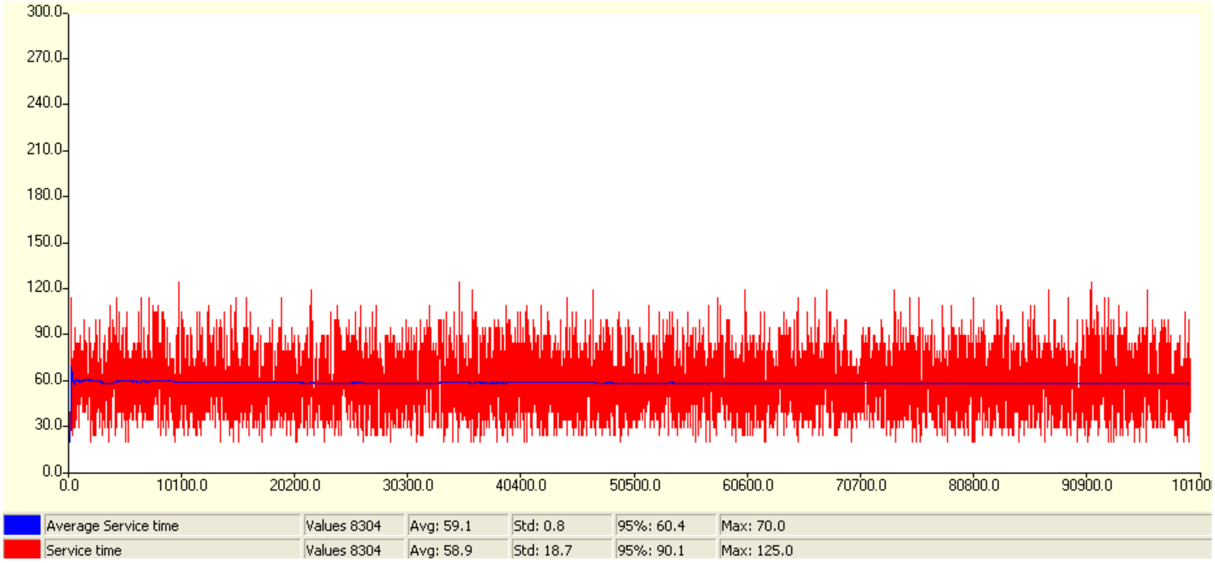
Besides the graph of the experience time, also a graph of the service time is made. This graph is shown in figure 4-3. With seven cashiers the average service time is about 70 seconds. The service time resulting from the simulation model is in the same range as the estimated service time in paragraph two of this chapter.

4.4 Six cashiers, one runner

Paragraph 4.4 deals with six cashiers and one runner. Again, graphs of the experience time and the service time are shown in figures 4-4 and 4-5 respectively.



4-4: Experience time 6 cashiers, 1 runner

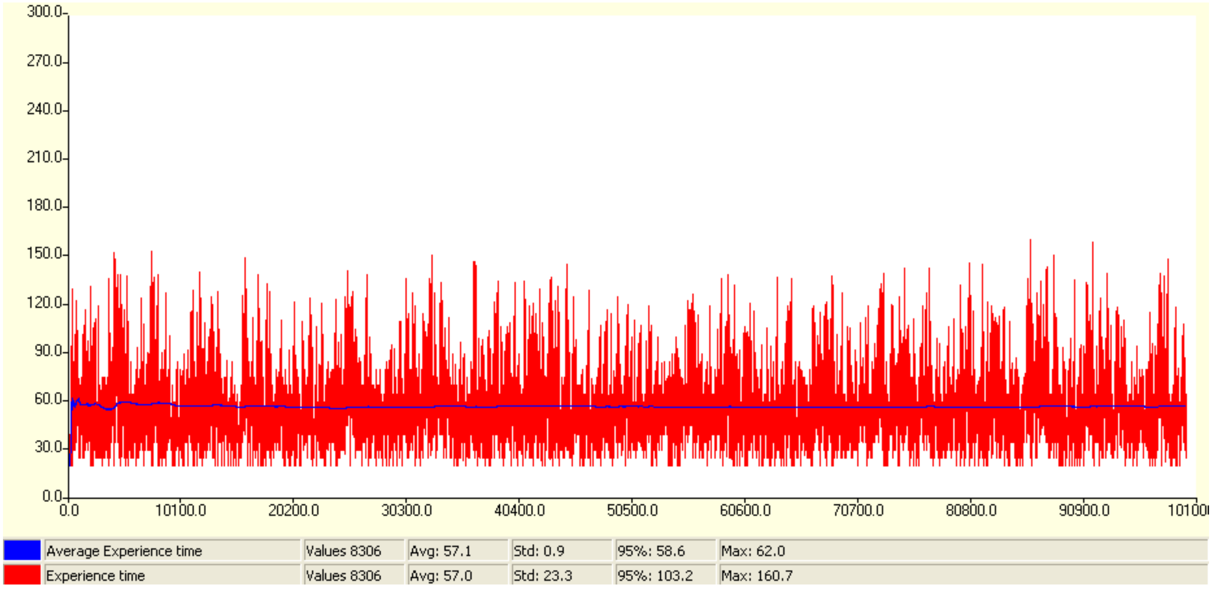


4-5: Service time 6 cashiers, 1 runner

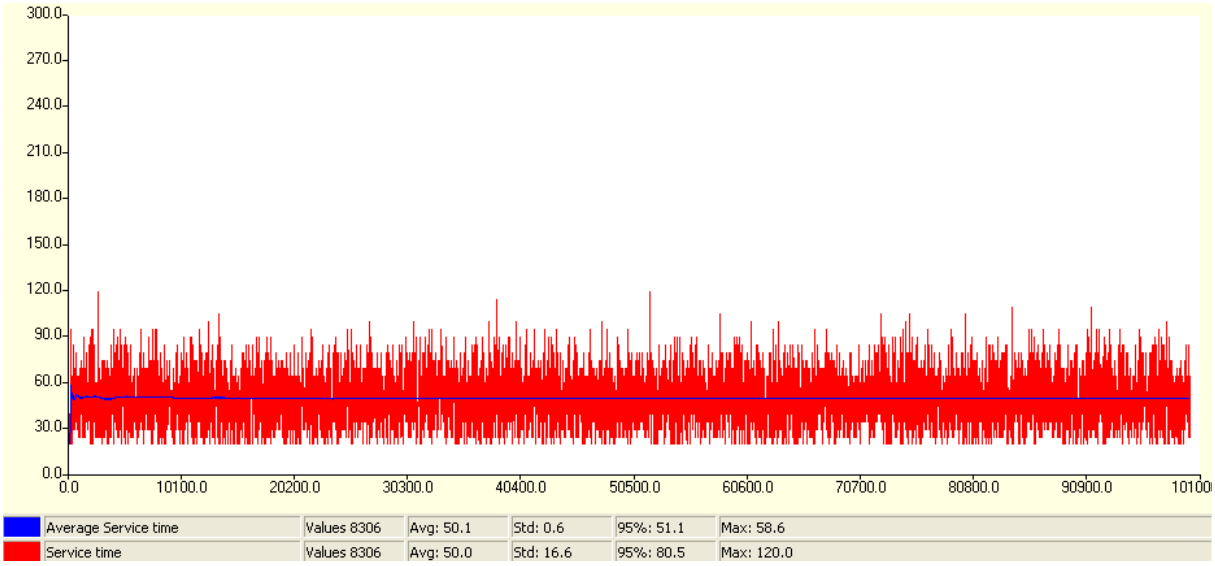
The average experience time is 64.4 seconds, this is 11.3 seconds lower than the experience time with seven cashiers in the previous paragraph. Also the peaks are much lower, with a maximum of 198.8 seconds, compared to 224.3 seconds with only seven cashiers. For 95% of the guests, the 95% quantile, the time from arrival till having the food is lowered with 14 seconds.
The service time shows a significant improvement as well. The time from start helping a guest till the order is ready is lowered with 11.1 seconds on average.

4.5 Five cashiers, two runners

In this paragraph the number of cashiers is reduced to five, but they are backed up by two runners. First the resulting experience time is shown in figure 4-6 and direct below the service time is visualized in figure 4-7.



4-6: Experience time 5 cashiers, 2 runners



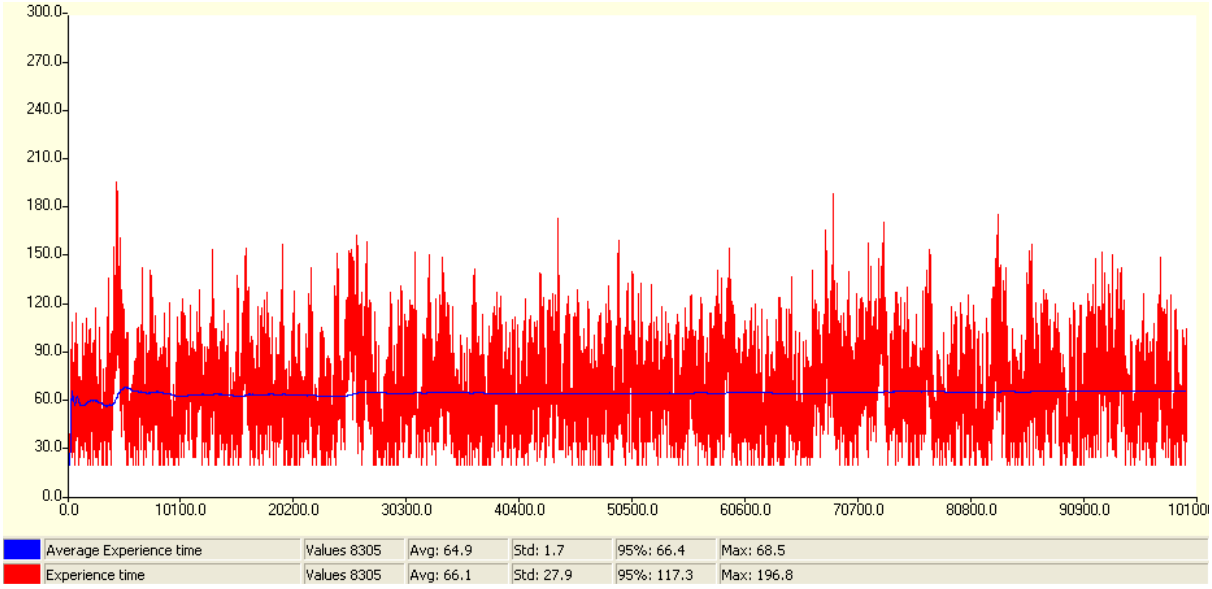
4-7: Service time 5 cashiers, 2 runners

With the use of two runners, the experience time reaches down to 57 seconds on average. This is 7.4 seconds less compared to the situation in the previous paragraph and 18.7 seconds, or 24.7% faster compared to the case of seven cashiers. The peaks are all below 160.7 seconds and 95% of the values are below 103.2 seconds. Besides the improvement of the average service time, the reduction of the maximum of the peaks of 63.6 second, more than a minute, is very beneficial.

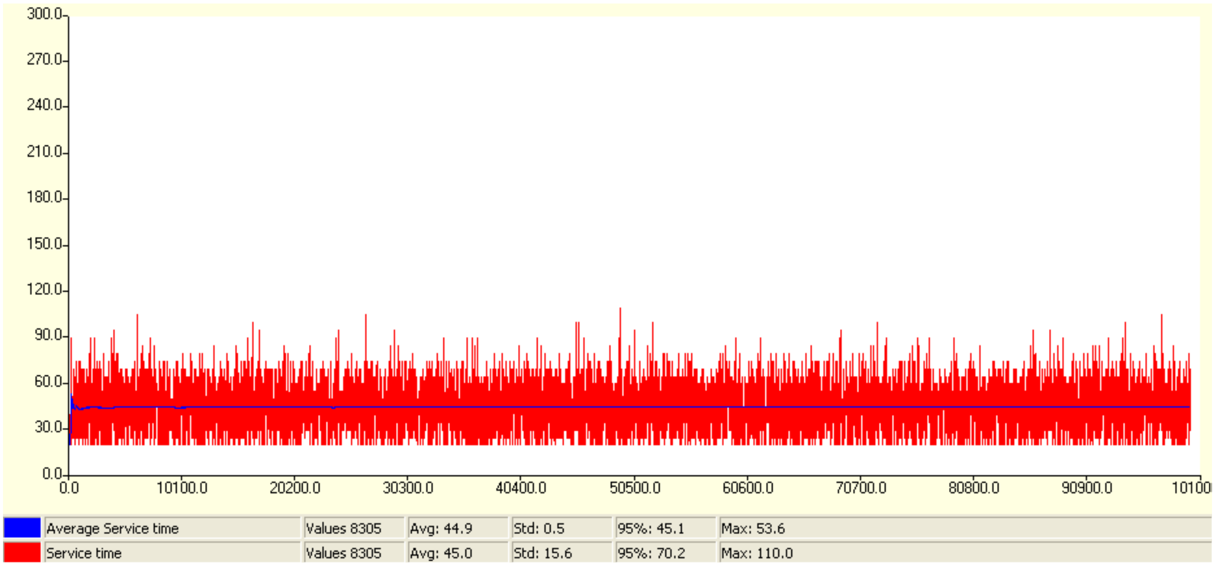
Especially the time of the guest at the cash register is reduced with 19.9 seconds compared to the initial situation with seven cashiers.

4.6 Four cashiers, three runners

The case of four cashiers and three runners checks if the experience time and service time could be reduced further. The results of this simulation run are shown in figure 4-8 and 4-9.



4-8: Experience time 4 cashiers, 3 runners



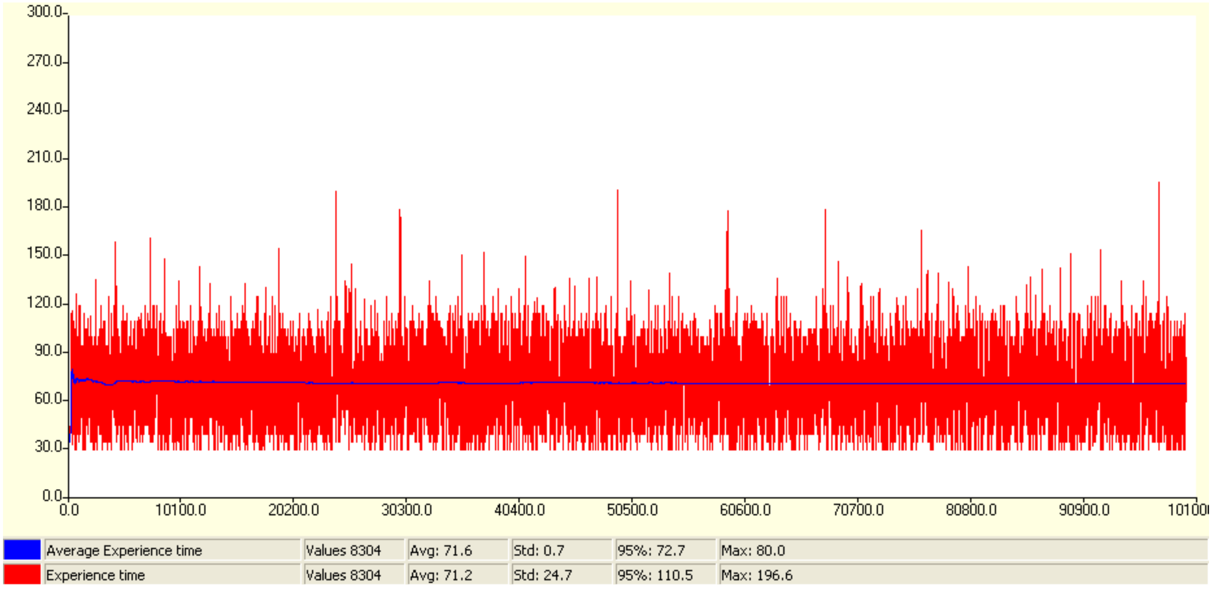
4-9: Service time 4 cashiers, 3 runners

Without the use of kiosks or more personnel the minimum experience time seems to be reached in the case with five cashiers and two runners. With one runner more and one cashier less, the average experience time goes up again to 66.1 seconds with a peak up to 196.8 seconds.

The service time is, however, still lowered by five seconds. The explanation for this is that guests are helped faster when they arrive at the cash register, but are much longer standing in line before they are helped. So the service time is lower, but the overall experience time is much higher compared to the case of five cashiers and two runners.

4.7 Five cashiers, no runners, two kiosks, two pick up points

To try to reduce the experience time further with only seven employees, the use of order kiosks is simulated. Guests have, when they enter the restaurant, the choice between ordering at a cash register or kiosk. In this paragraph the case of five cashiers, no runners, two kiosks and two pick up points is worked out and compared to the initial situation with seven cashiers and to best situation without kiosks, the case of five cashiers and two runners. Only the experience time is shown and can be found in figure 4-10. The service time is less significant, because this is the time an employee uses per guest. Now the guest is entering his order himself, the service time says less about the quality of the service. The service time can, however, be found in appendix B.



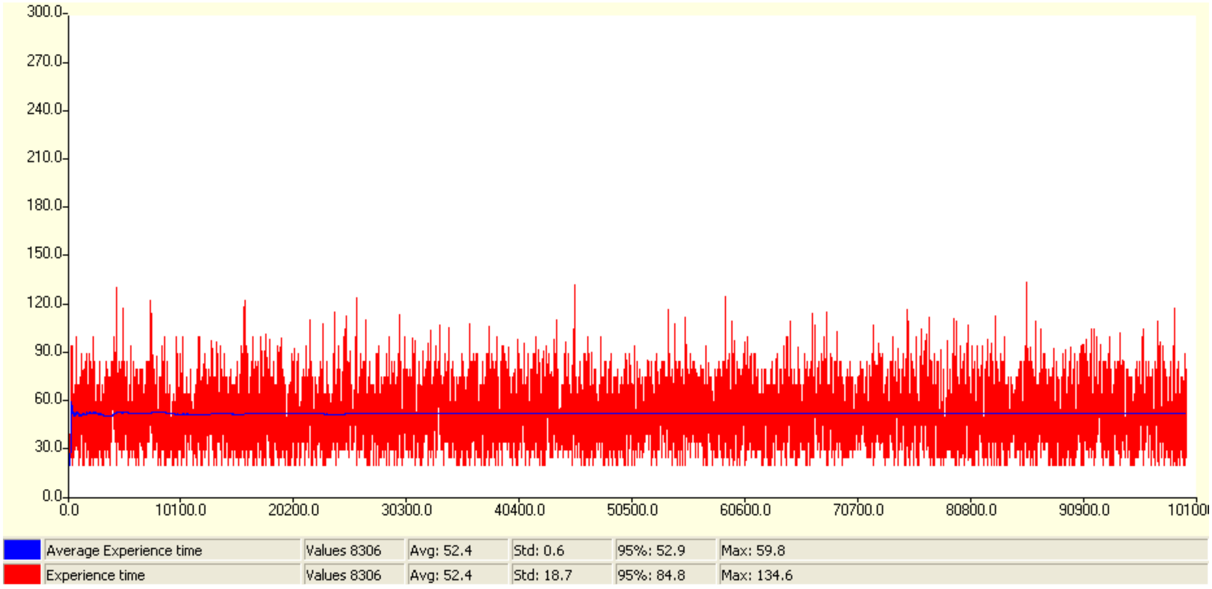
4-10: Experience time 5 cashiers, 0 runners, 2 kiosks, 2 pick up points

The resulting average experience time is 71.2 seconds. 95% of the values are below 110.5 seconds and the maximum peak is at 196.6 seconds.

Compared to the initial situation with only seven cashiers this is an improvement of 4.5 seconds for the average experience time, 15.3 seconds for the 95% quantile and 27.7 seconds for the highest peak. So if no runners are implemented, there can be made an improvement by installing order kiosks. This, however, needs an investment in equipment. If there are runners used, the average experience time, 95% quantile and maximum can be 57.0, 103.2 and 160.7 seconds respectively, according to paragraph 4.5. So without an investment in equipment, the average experience time can be lowered by 14.2 seconds compared to the case in this paragraph. The 95% quantile can be lowered by 7.3 seconds and the highest peak can be 35.9 seconds less. It would thus be more beneficial to invest in the training of runners instead of installing new equipment.

4.8 Four cashiers, two runners, two kiosks, one pick up point

Supposing there are runners trained, this paragraph consists of a combination of cashiers, runners, kiosks and thus a pick up point. Simulation runs proved no need for a second pick up point when two order kiosks are installed. This employee can be used better elsewhere. This paragraph shows that with seven employees, divided over four cashiers, two runners, two kiosks and one pick up point, the lowest 95% quantile is reached. The graph of the experience time is shown in figure 4-11 below.



4-11: Experience time 4 cashiers, 2 runners, 2 kiosks, 1 pick up point

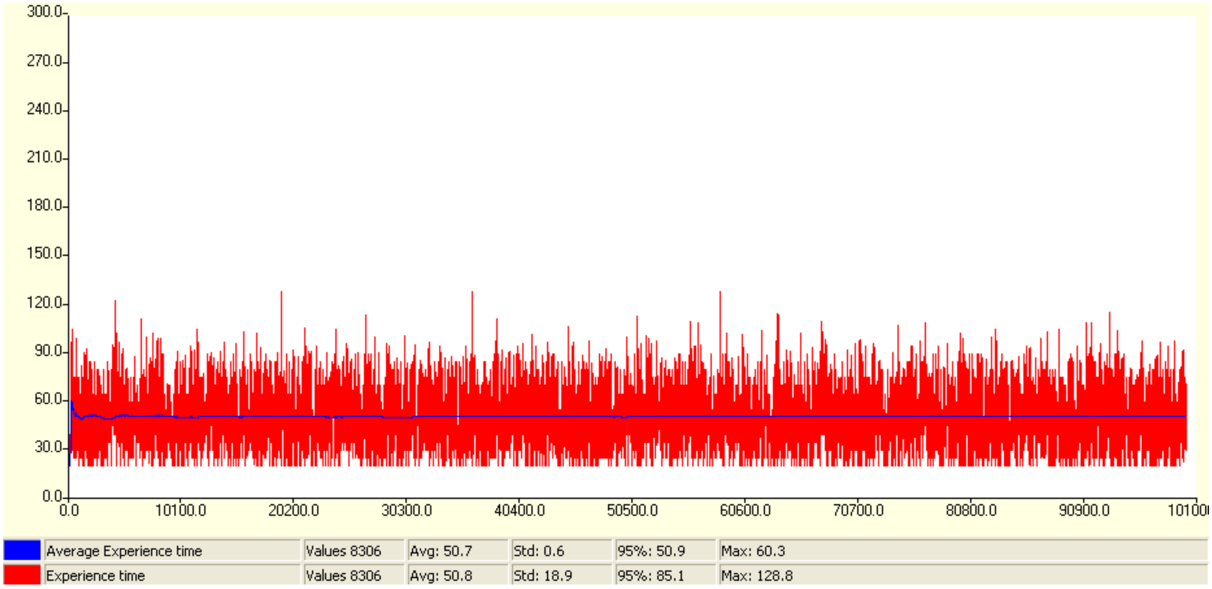
The results for this simulation run are 52.4 seconds for the average experience time, 84.8 seconds for the 95% quantile and 134.6 seconds for the highest peak. These results are, again, compared with the initial situation of only seven cashiers and with the case of five cashiers and two runners, the best case without kiosks.

Compared to the initial situation advantages of 23.3 seconds for the average experience time, 41.0 seconds for the 95% quantile and 89.7 seconds are reached. For the 95% quantile this means a benefit of 32.6%. For 95% of the guests, the implementation of both runners and kiosks means a 32.6% shorter experience time.

To determine the influence of the order kiosks to the result above, the case of this paragraph is compared with the best result without kiosks. For the average experience time, the implementation of order kiosks means an advantage of 4.6 seconds. The advantages for the 95% quantile and maximum peak are 18.8 and 26.1 seconds respectively. For 95% of the guests this means an improvement of 18.2% due to the order kiosks.

4.9 Three cashiers, three runners, two kiosks, one pick up point

In the previous paragraph the optimal occupation for the best 95% quantile of the guest was treated. In this paragraph the best results for both the average experience time and height of the maximum peak are obtained. The result of the simulation run for the experience time is shown in figure 4-12.



4-12: Experience time 3 cashiers, 3 runners, 2 kiosks, 1 pick up point

A similar result as in the previous paragraph is found. The average experience time in this case with a cashier less and a runner more, so with three cashiers, three runners, two kiosks and one pick up point is 50.8 seconds. The 95% quantile is at 85.1 seconds and the highest peak is at 128.8 seconds. Compared to the initial situation with only seven cashiers this is an improvement of 24.9 seconds, or 32.9% for the average experience time. For the 95% quantile the experience time is 40.7 seconds less and the highest peak is 95.5 seconds or 42.6% less.

When comparing with the best results without kiosks, the benefits are 6.2 seconds for the average experience time, 18.1 seconds for the 95% quantile of the guests and 31.9 seconds for the highest peak. These are benefits of 10.9%, 17.5 and 19.9% for the average experience time, the 95% quantile and the highest peak respectively, due to the use of kiosks.

4.10 Comparison results

In this paragraph the results of the previous paragraphs are summarized in figure 4-13. Also results from simulation runs that are performed, but not pointed out in this chapter are shown. Further details of these runs can be found in appendix B.

				Experience time (s)					
# Cashiers	# Runners	#Kiosks	#Pick up points	Average	95% quantile	Maximum value	Service time (s)	Bring after (%)	Waste (%)
7	0	0	0	75.7	125.8	224.3	70	0.73	0.44
6	1	0	0	64.4	111.8	198.8	59	2.46	0.42
5	2	0	0	57.0	103.2	160.7	50	4.65	0.39
4	3	0	0	66.1	117.3	196.8	45	5.80	0.40
6	0	2	1	71.5	110.0	200.7	65	1.07	0.46
5	0	2	2	71.2	110.5	196.6	61	1.45	0.50
5	1	2	1	60.6	95.2	145.0	53	2.90	0.43
4	1	2	2	61.5	94.9	172.7	50	3.55	0.51
4	2	2	1	52.4	84.8	134.6	44	5.08	0.42
3	2	2	2	54.3	84.9	171.0	40	5.50	0.44
3	3	2	1	50.8	85.1	128.8	37	6.07	0.40
2	3	2	2	55.3	88.8	135.8	33	6.50	0.57

4-13: Results simulation runs

Recall that seven employees are available and the input variables are as described in paragraph 4.1. The initial situation is with seven cashiers, no runners, no kiosks and no pick up points. The results are then 75.7 seconds for the average experience time, 125.8 seconds for the 95% quantile and 224.3 seconds for the highest peak. These numbers can be found in the top row of the table in figure 4-13.
When no kiosks are installed the best average experience time is 57.0 seconds and is found in the case of five cashiers and two runners, row three of the table. The 95% quantile is at 103.2 seconds and the highest peak at 160.7 seconds. These are improvements of 18.7 seconds for the average experience time, 22.6 seconds for the 95% quantile and 63.6 seconds for the highest peak, or 24.7%, 18.0% and 28.3% respectively.
When there are kiosks installed, further improvements can be made. There is a best case for the 95% quantile and a best case for the average experience time and maximum value. The best case for the 95% quantile is with four cashiers, two runners, two kiosks and one pick up point. The best case for the average experience time and highest peak is with the use of three cashiers, three runners, two kiosks and one pick up point. The benefits are shown in figure 4-14, with in the two columns on the right of the obtained values the benefits in seconds and percentages with respect to the initial situation.

Case				Experience time (s)								
# Cashiers	# Runners	# Kiosks	# Pick up points	**Average**	Improvement (s)	Improvement (%)	**95% quantile**	Improvement (s)	Improvement (%)	**Maximum value**	Improvement (s)	Improvement (%)
7	0	0	0	**75.7**	-	-	**125.8**	-	-	**224.3**	-	-
5	2	0	0	**57.0**	18.7	24.7	**103.2**	22.6	18.0	**160.7**	63.6	28.4
4	2	2	1	**52.4**	23.3	30.8	**84.8**	41.0	32.6	**134.6**	89.7	40.0
3	3	2	1	**50.8**	24.9	32.9	**85.1**	40.7	32.4	**128.8**	95.5	42.6

4-14: Comparison results

As can be seen from 4-14, significant improvements can be made by implementing runners to help the guests, instead of manning more cash registers. A lower experience time can be obtained by using two runners and five cashiers, compared to using only seven cashiers.

This benefit can be improved further by implementing order kiosks. The average experience time becomes a bit less and thus better overall, but for 95% of the guests there is a real significant improvement. The implementation of order kiosks almost doubles the reduction in experience time compared to the use of just runners for 95% of the guests.

5. Case Study 2

In this chapter a smaller case is set out and comes closer to a day to day situation in an average fast food restaurant in the Netherlands. A planned turnover of €1500.- per hour is used, with still an average spending of €10.- per guest. Furthermore the time to collect products is set longer compared to the case of chapter four. The summary of the input variables is shown in figure 5-1. The number of cashiers needed based on averages is calculated in paragraph 5.1. In paragraph 5.2 the case with four cashiers and no runners is described, while in paragraph 5.2 one runner and only three cashiers are employed. The simulation run with two cashiers and two runners is performed, but the experience time rose towards infinity. In this case study no order kiosks are used, because they only are installed in a few restaurants. In paragraph 5.3 the results are compared.

Planned turnover per hour (€)	1500.-					
Average spending (€)	10.-					
Product	**BM**	**HB**	**QP**	**Chicken**	**Croquette**	**Fish**
Price (€)	3.35	1.00	3.35	3.35	2.10	2.00
Division	8	16	4	6	4	1
Order time (s)	5	5	5	5	5	5
Collect time (s)	10	10	10	10	10	10
Preparation time (s)	90	90	150	30	30	30
System	2	2	2			
Product	**Wrap**	**Nuggets**	**Drink**	**Ice**	**Fries**	
Price (€)	2.00	2.00	2.00	2.35	0.80	
Division	2	12	23	8	15	
Order time (s)	5	5	5	5	5	
Collect time (s)	10	10	3	3	10	
Preparation time (s)	20	10	7	12	5	

5-1: Input variables

5.1 Averages calculation

When the turnover per hour is €1500.- and the average spending is €10.- per guest, 150 guest per hour will arrive on average.

To come to an average number of products per guest, the average price of the products, with respect to the division over the products, is needed. The average price of a product is calculated by the total number of products, so the total of the row 'division' in figure 5-1, divided by the sum of the division of the product times their price, so the number in the row 'division' times the number in the row 'price' in figure 5-1. The average number of products per guest can then be derived by dividing the average spending of a guest, €10.-, by the average price. The formula for the average number of products in this case is:

$$\frac{10\cdot(8+16+4+6+4+1+2+12+23+8+15)}{8\cdot3.35+16\cdot1.00+4\cdot3.35+6\cdot3.35+4\cdot2.10+1\cdot2.00+2\cdot2.00+12\cdot2.00+23\cdot2.00+8\cdot2.35+15\cdot0.80}=$$

$=\frac{10\cdot99}{171.5}=5.77$, so on average 5.77 products per guest are ordered.

A guest takes five seconds to order a product. The employee will use ten seconds per product to collect it. Drinks have to be prepared by the cashier. The preparation time plus the collect time is again ten seconds. Ice creams also have a preparation time, but this is twelve seconds plus the collect time. So in eight out of 99 cases the time to collect the product is fifteen instead of ten seconds. In the other 91 cases the time to collect the product is the given ten seconds. So the average time to collect a product is:

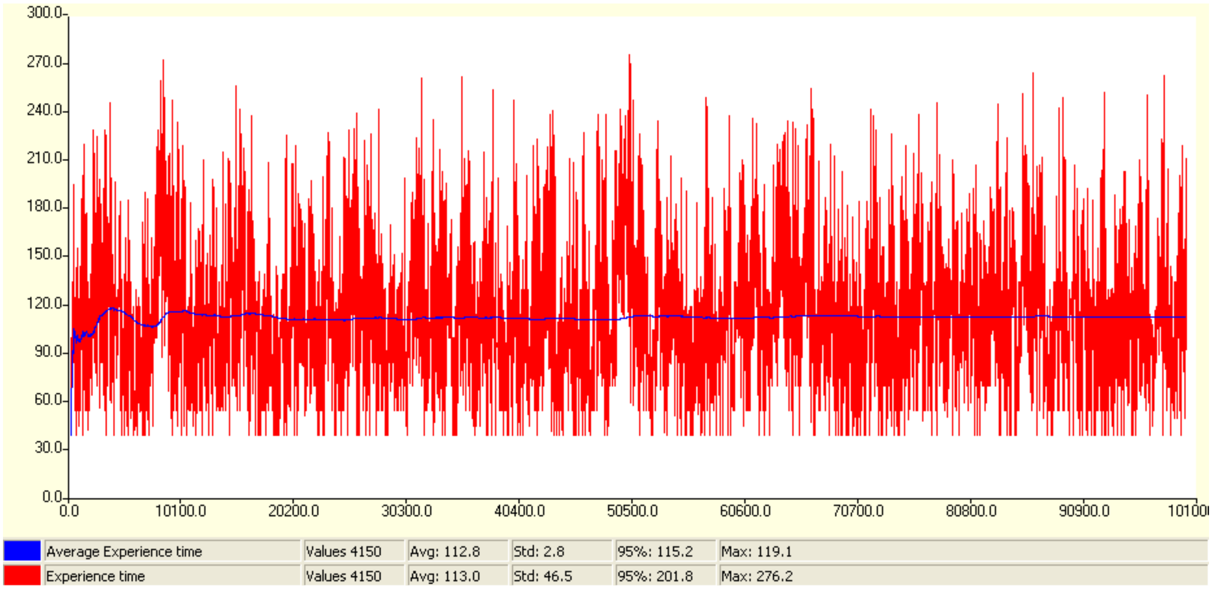
$\frac{91\cdot10+8\cdot15}{99}=10.4$ seconds.

The payment time is set to ten seconds. Thus, the average service time per guest is the average number of products times the time to order, plus the average number of products times the average time to collect, plus the time to pay, this is: $5.77\cdot5+5.77\cdot10.4+10=98.9$ seconds. This is the average service time.

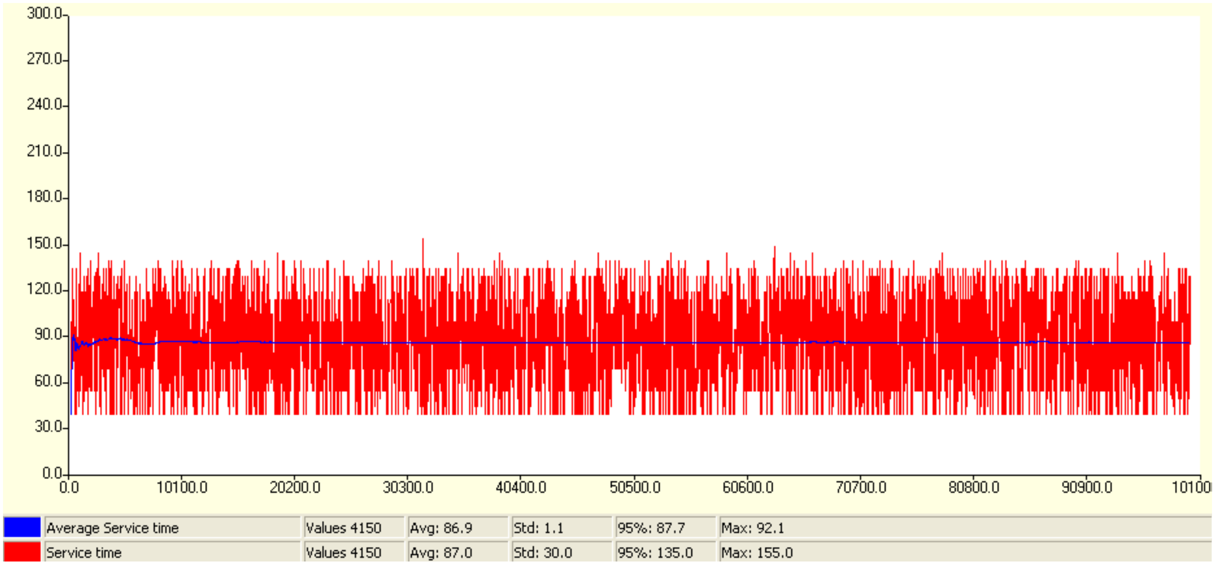
With 150 guests per hour, it means that one guest must be helped every $3600/150=24$ seconds. Recalling that helping a guest takes 98.9 seconds on average, 4.1 cashiers should be helping guests.

5.2 Four Cashiers, no Runners

The initial situation for this case study is four cashiers and no runners. The results for the experience time and the service time are shown in figures 5-2 and 5-3 respectively.



5-2: Experience time 4 cashiers, 0 runners

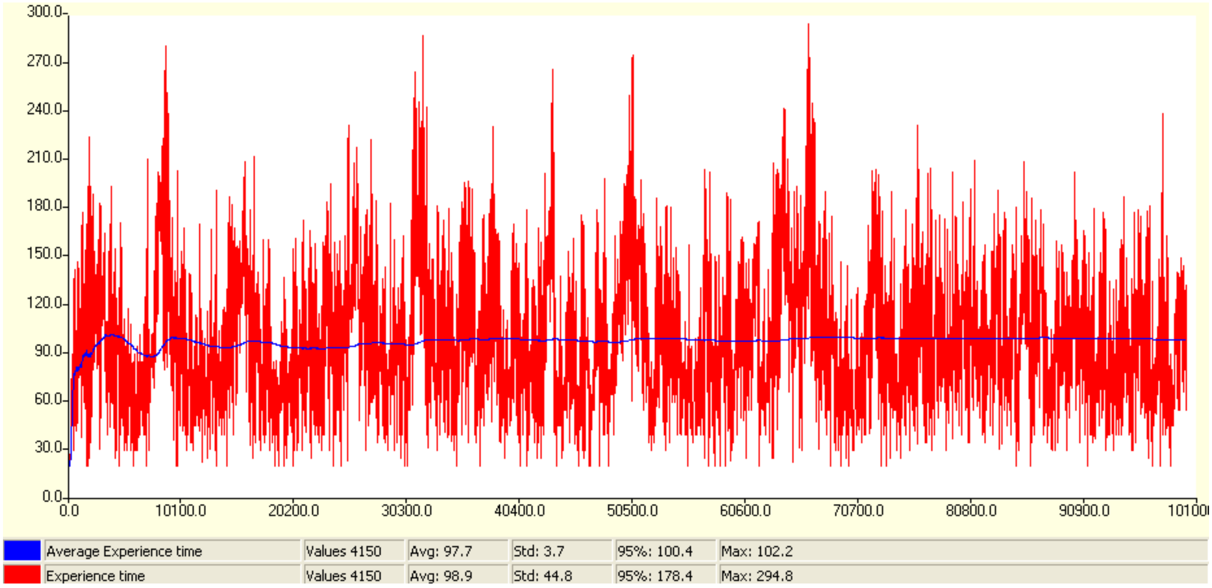


5-3: Service time 4 cashiers, 0 runners

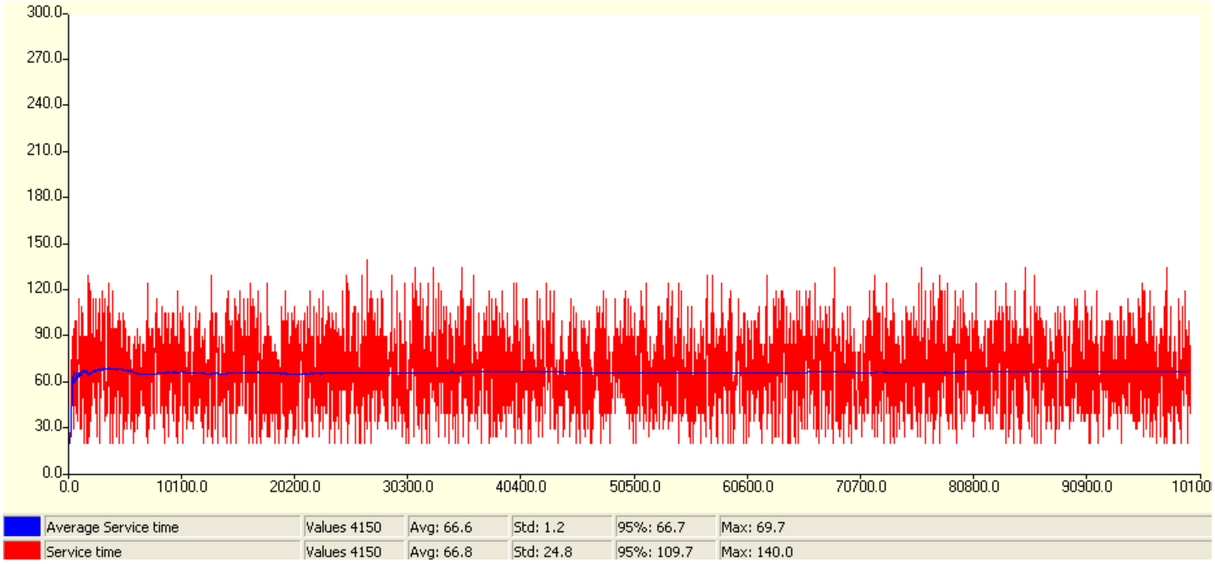
The average experience time is 113.0 seconds. The 95% quantile is at 201.8 seconds, this means that 95% of the values are below 201.8 seconds. The maximum value is 276.2 seconds. The average service time is 87.0 seconds.

5.3 Three Cashiers, one Runner

The situation that is simulated next is the case of three cashiers and one runner. The resulting experience times are shown in figure 5-4 and the service times in 5-5.



5-4: Experience time 3 cashiers, 1 runner



5-5: Service time 3 cashiers, 1 runner

The average experience time is 98.9 seconds, the 95% quantile is at 178.4 second and the highest peak is at 294.8 second. The average service time is 66.8 seconds. The graphs show more peaks, but the average is lower compared to the initial situation.

5.4 Comparison Results

In this paragraph the results from the previous two paragraphs are compared and summarized in 5-6. As stated before, the case with two cashiers and two runners results in unwanted behavior with peaks reaching above 2500 seconds for the average experience time.

Case		Experience time (s)									Service time (s)		
# Cashiers	# Runners	**Average**	Improvement (s)	Improvement (%)	**95% quantile**	Improvement (s)	Improvement (%)	**Maximum value**	Improvement (s)	Improvement (%)	**Average**	Improvement (s)	Improvement (%)
4	0	**113.0**	-	-	**201.8**	-	-	**276.2**	-	-	**87.0**	-	-
3	1	**98.9**	14.1	12.5	**178.4**	23.4	11.6	**294.8**	-18.6	-6.7	**66.8**	20.2	23.2

5-6: Comparison results

The average experience time for the case with three cashiers and one runner decreases with 14.1 seconds compared to the initial situation. This is an improvement of 12.5%. Also the value under which 95% of the measured experience times lie is lowered with 23.4 seconds. This is an improvement of 11.6% compared to the situation with only four cashiers. The average service time shows an improvement of 20.2 seconds or 23.2%.

The maximum value however, shows a deterioration of 18.6 seconds, which is 6.7% compared to the initial situation. This is a good reflection of the higher peaks and more scattered graphs in paragraph 5.3.

The conclusion is that, however the peaks are higher, it is beneficial to implement a runner instead of a fourth cashier. This because not only the average experience time is lower, but also the 95% quantile. 95% of the guests experience a shorter time till having their meals.

6. Conclusions

The main research question is: given a certain turnover per hour and average spending, how is the experience time influenced by employing more or less cash registers, more or less runners and more or less order kiosks?

The first conclusion from the two performed cases is that it is beneficial to employ runners, but to a certain level. If the kiosks are excluded, the first case has the best result with two runners and five cashiers, instead of only seven cashiers. An improvement of 24.7% for the average experience time is realized. With more than two runners the results deteriorate again. The second case shows an improvement of 12.5% for the average experience time when one runner and three cashiers are working instead of just four cashiers.

Simulation runs showed there is a negligible effect of using two pick up points instead of one for two kiosks. Further simulations are thus performed with two order kiosks and one pick up point.

The next conclusion that can be drawn from the first case is the advantage of using order kiosks. Here the guests enter the orders themselves, leaving more manpower to collect the products. The runners are able to choose between helping a cashier and help collecting products at a pick up point. When again seven employees are working in the service, the best occupation of the service spots is: three cashiers, three runners and one pick up point with two order kiosks. The average experience time then improves with 32.9% compared to the situation with only seven cashiers.

So without using more employees an improvement in average experience time can be made by implementing runners. No investments in equipment are necessary, only the training of these employees should be extended. If there is a will to invest, even further improvements can be made by installing order kiosks. The consideration and calculation of the costs and benefits of this investment are beyond the scope of this research.

Appendix A: Code Simulation Model

Due to the length of the simulation code, over 100 pages, it can be found in a document on the attached CD. The document is titled: SimulationCode.txt.

The CD furthermore contains:

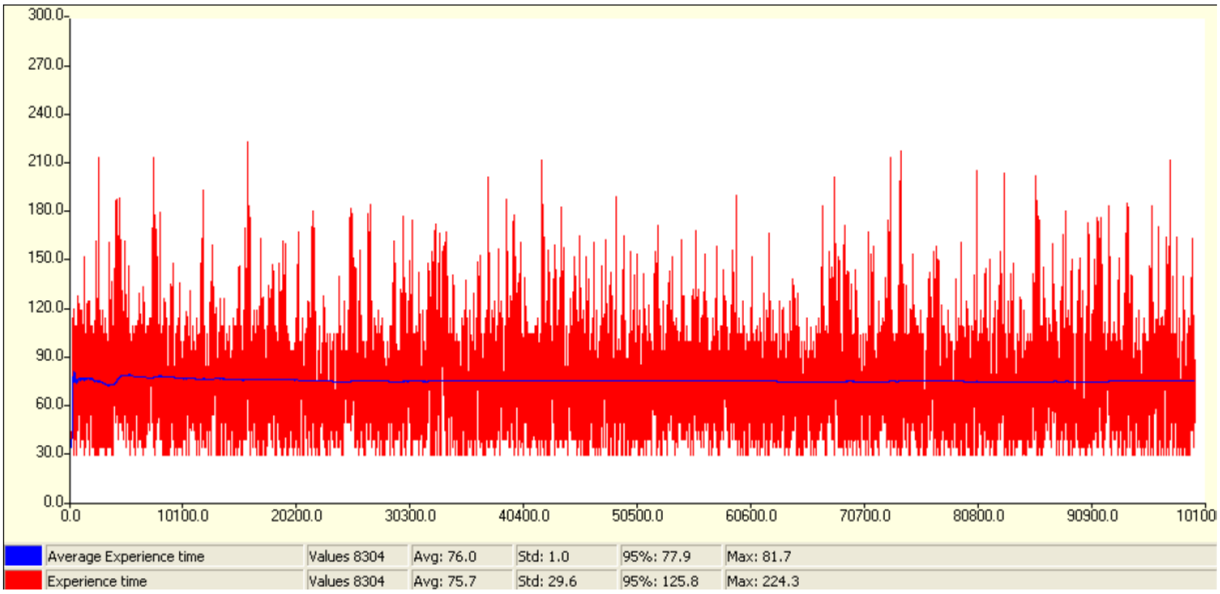
- This report as document: Report.doc
- This report as reader: Report.pdf
- The graphs of the two cases
- The used simulation
- A ReadMe.txt file with the information on this page

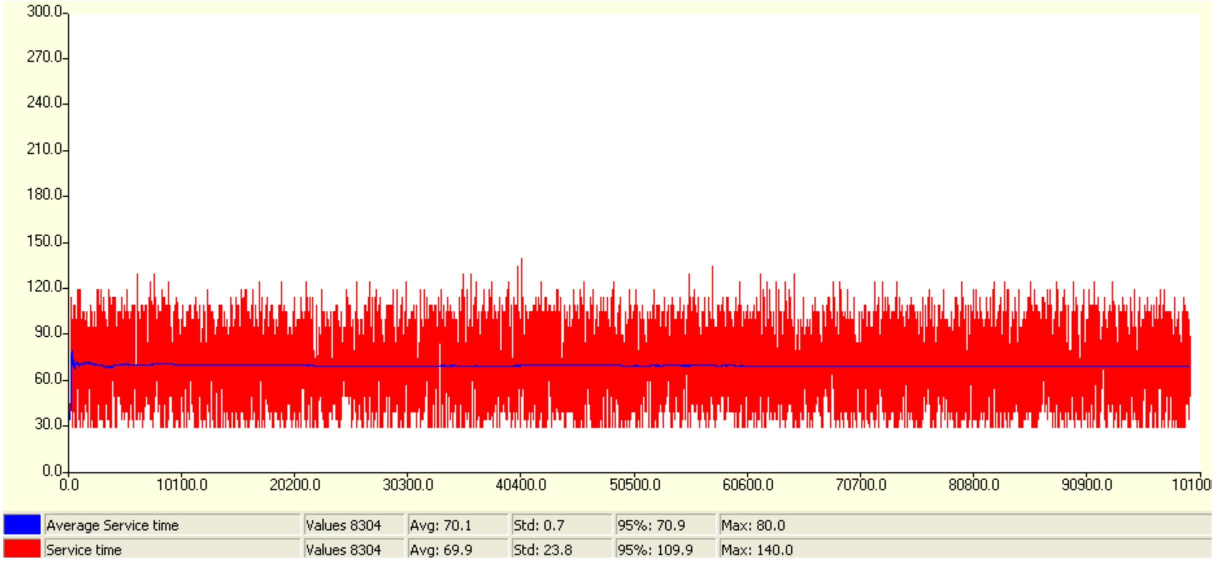
The simulation can only be run with Delphi, this is a program that has to be purchased. The simulation also uses TOMAS, an attachment to Delphi, and is free to download from www.tomasweb.com.

Appendix B: Results Simulation Model

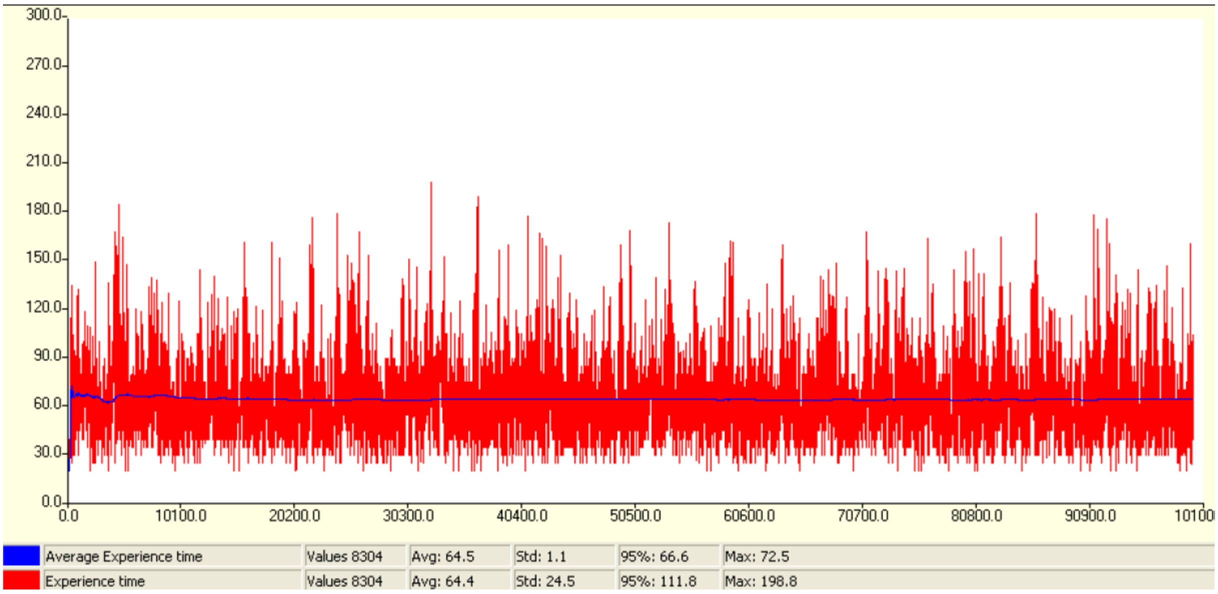
In this appendix the results of the simulation runs can be found.

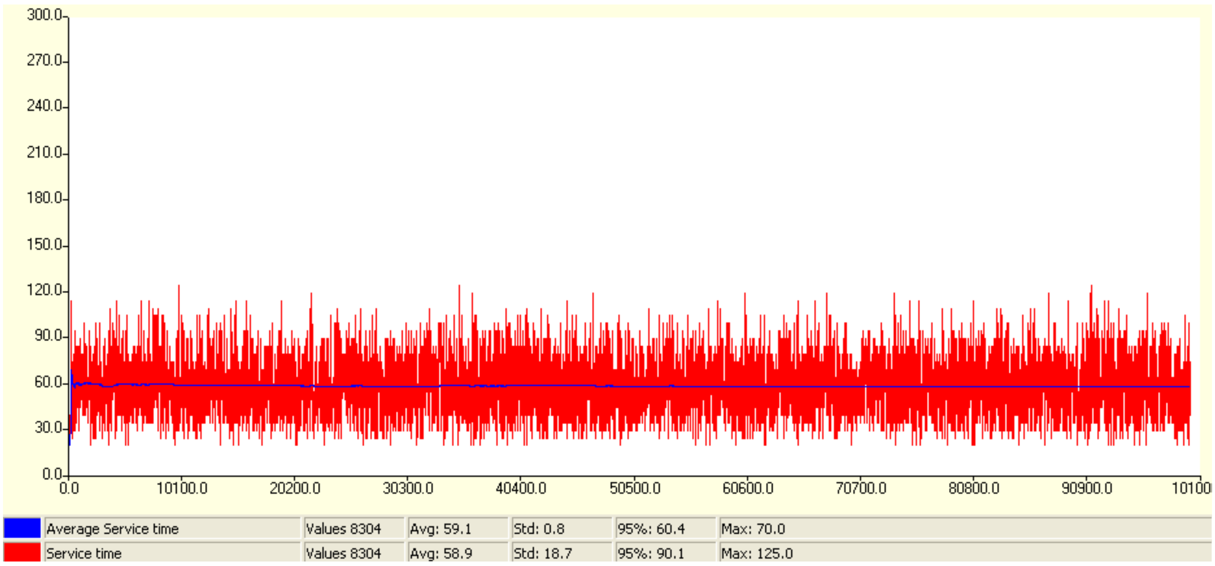
Seven cashiers



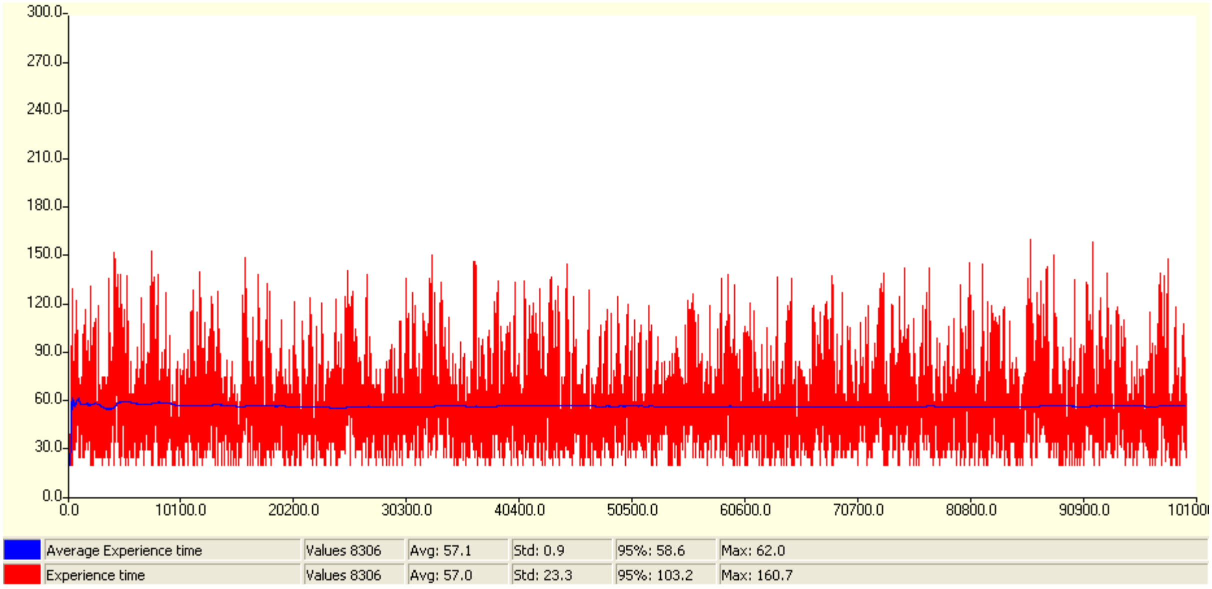


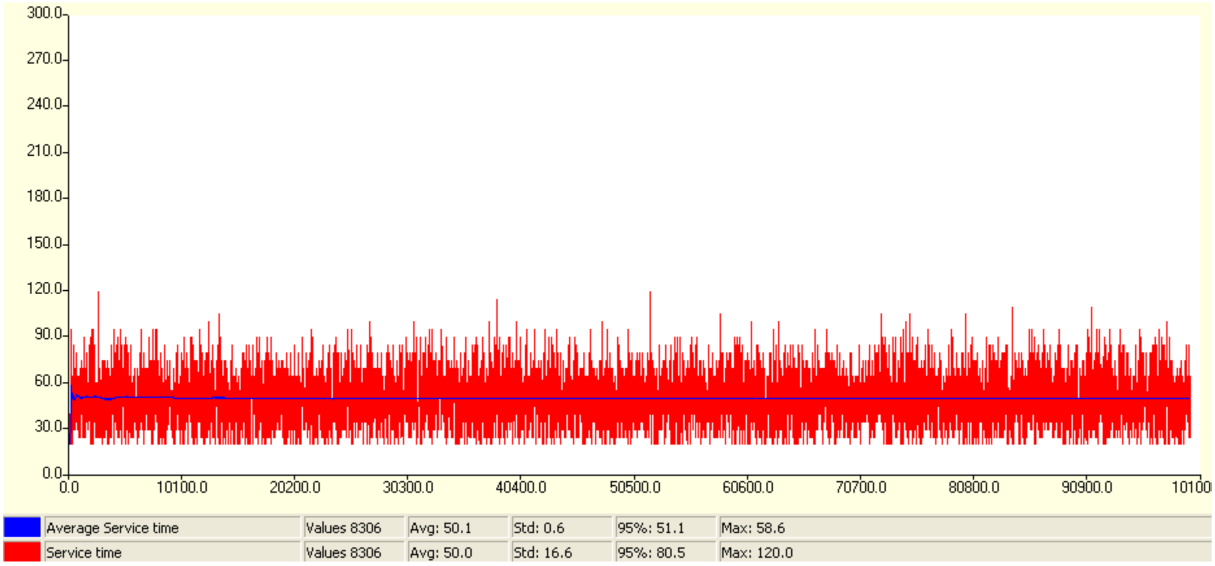
Six cashiers, one runner



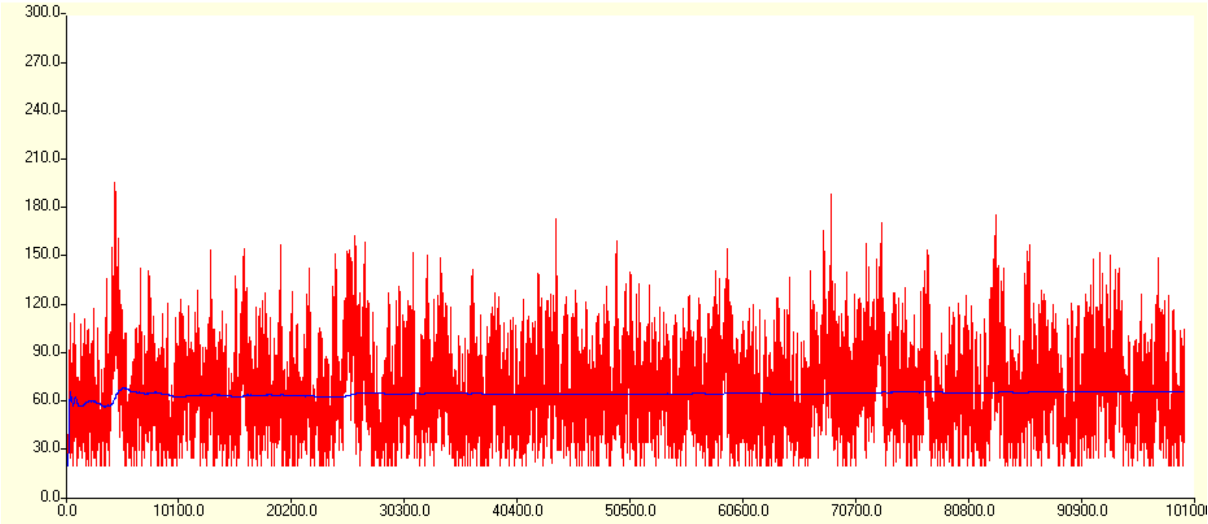


Five cashiers, two runners

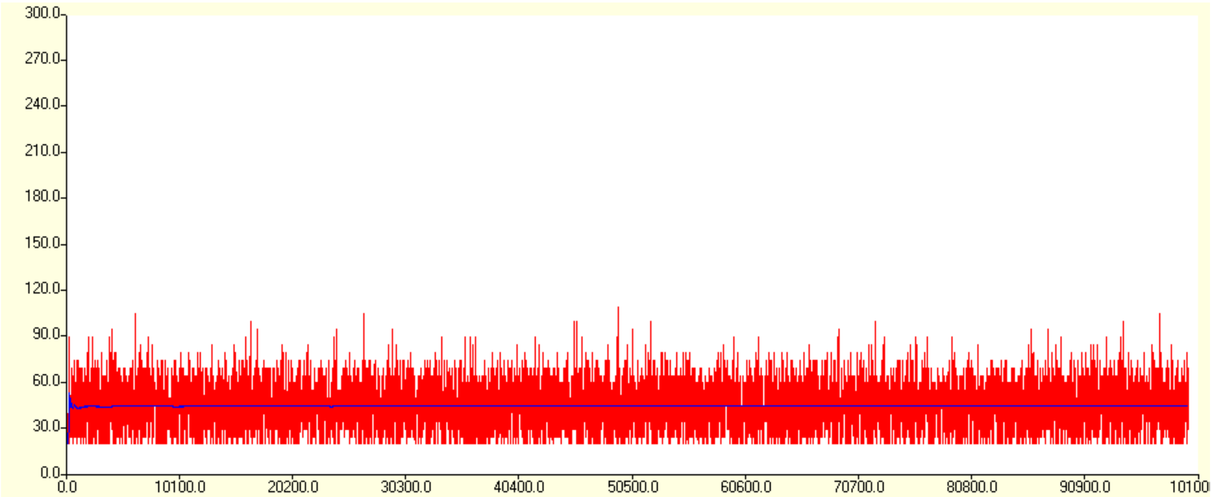




Four cashiers, three runners

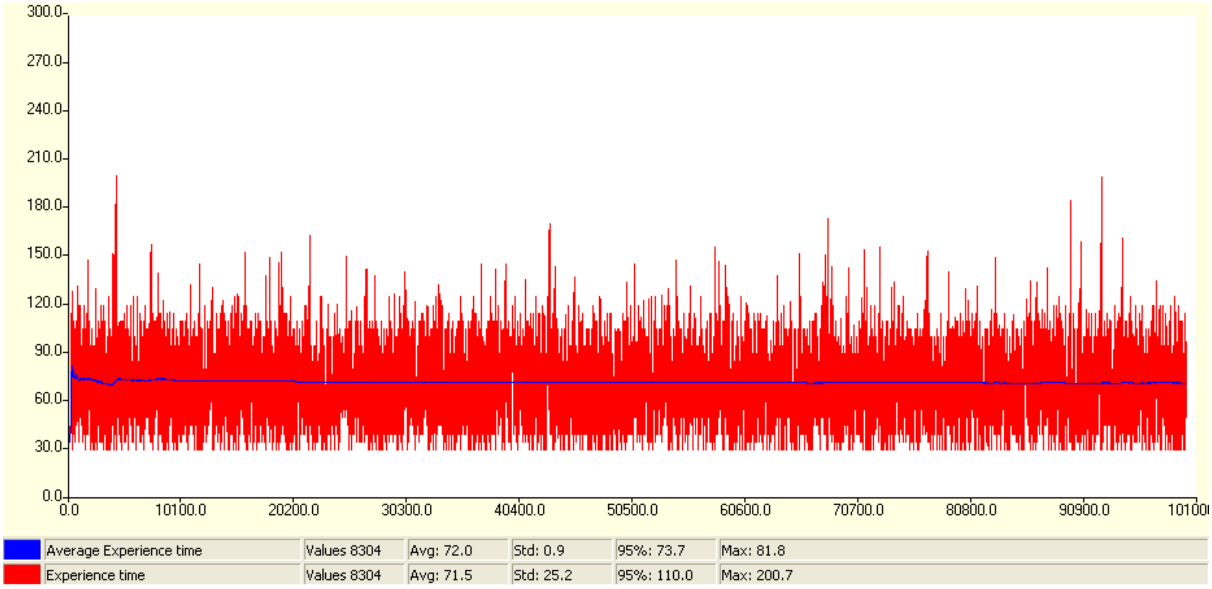


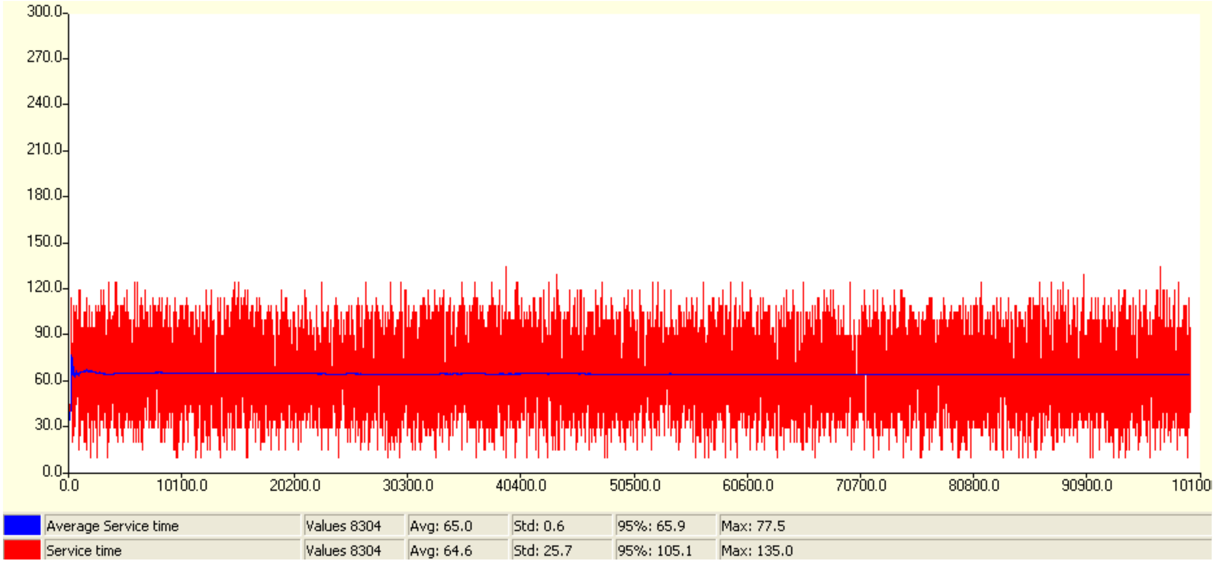
Average Experience time	Values 8305	Avg: 64.9	Std: 1.7	95%: 66.4	Max: 68.5
Experience time	Values 8305	Avg: 66.1	Std: 27.9	95%: 117.3	Max: 196.8



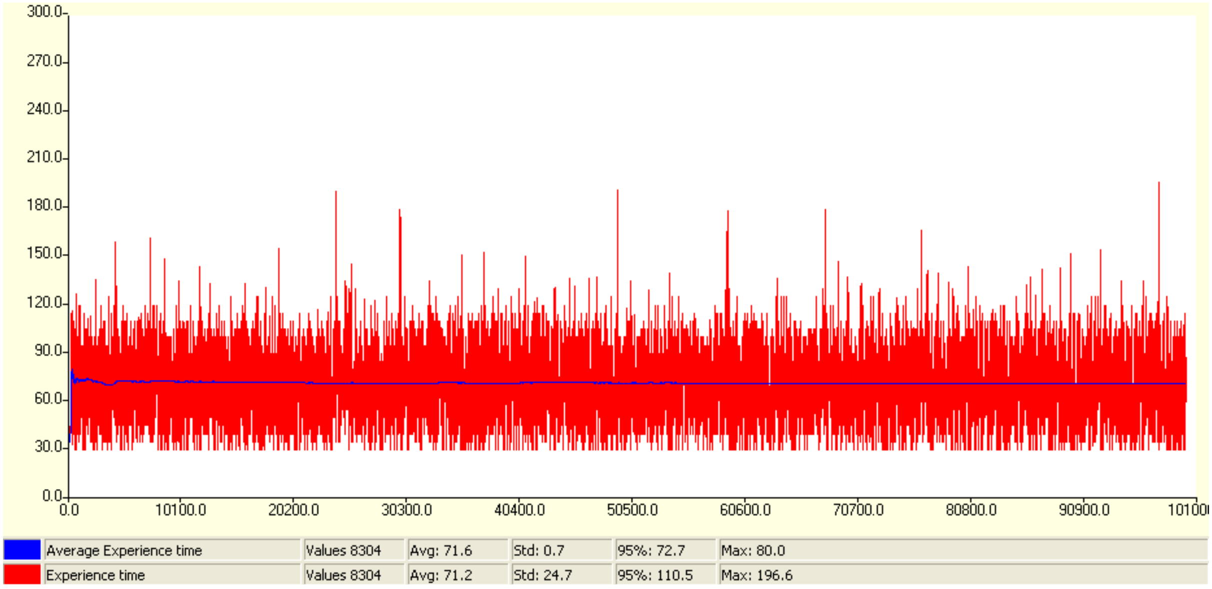
Average Service time	Values 8305	Avg: 44.9	Std: 0.5	95%: 45.1	Max: 53.6
Service time	Values 8305	Avg: 45.0	Std: 15.6	95%: 70.2	Max: 110.0

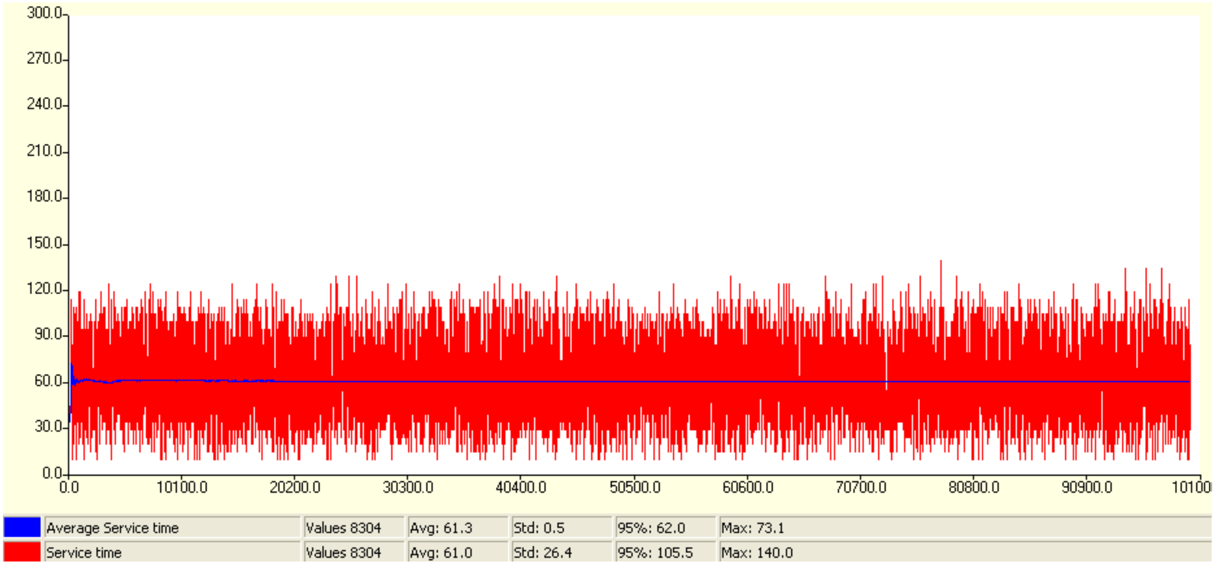
Six cashiers, zero runners, two kiosks, one pick up point



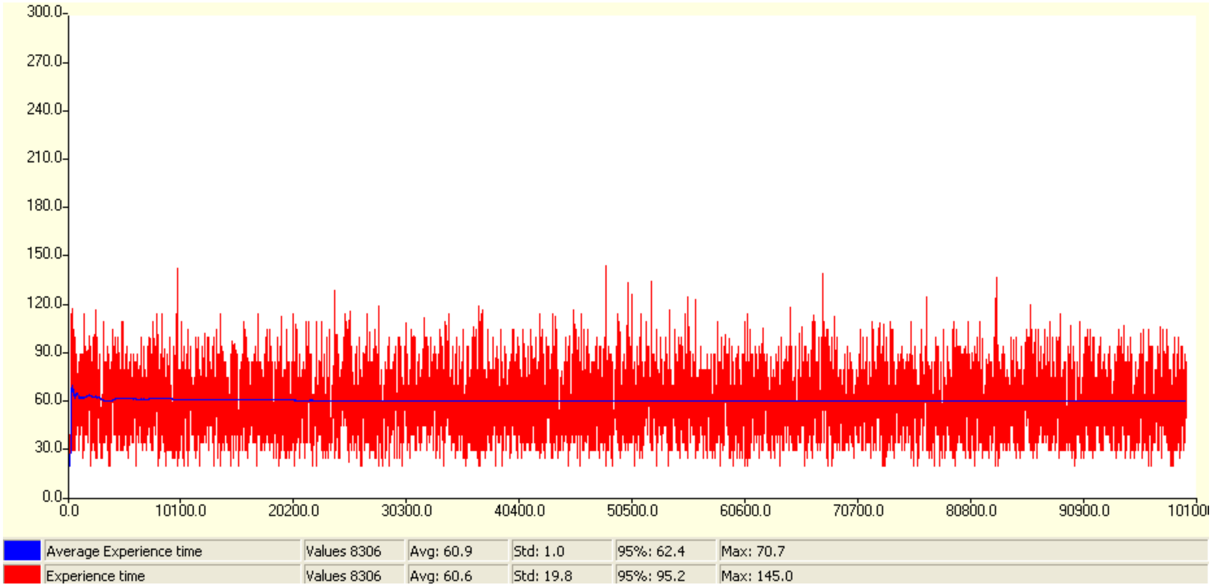


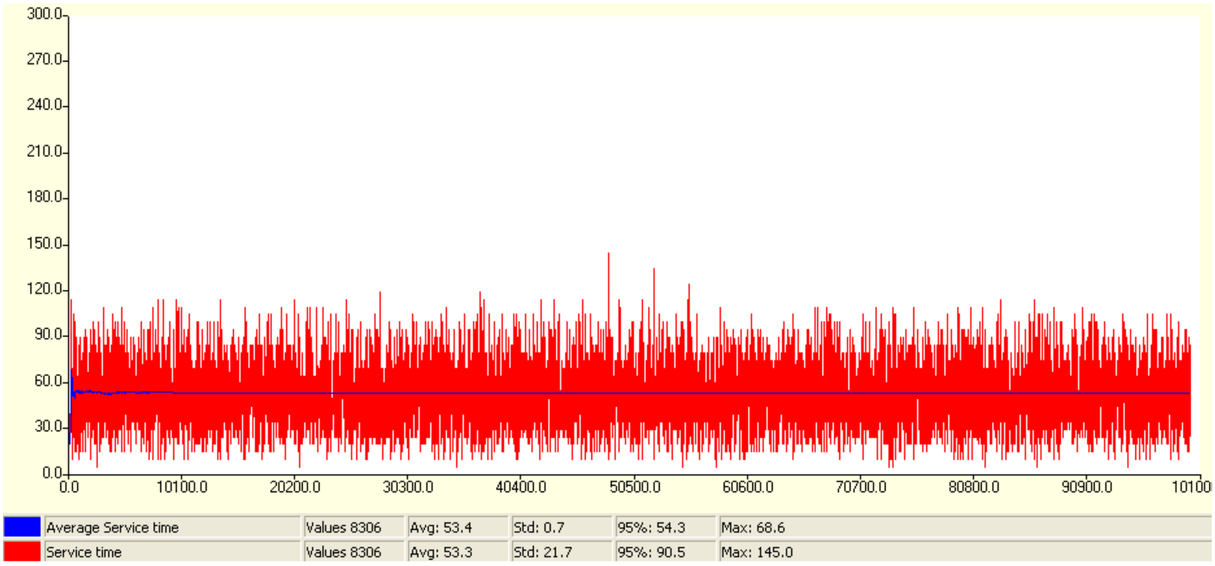
Five cashiers, zero runners, two kiosks, two pick up points



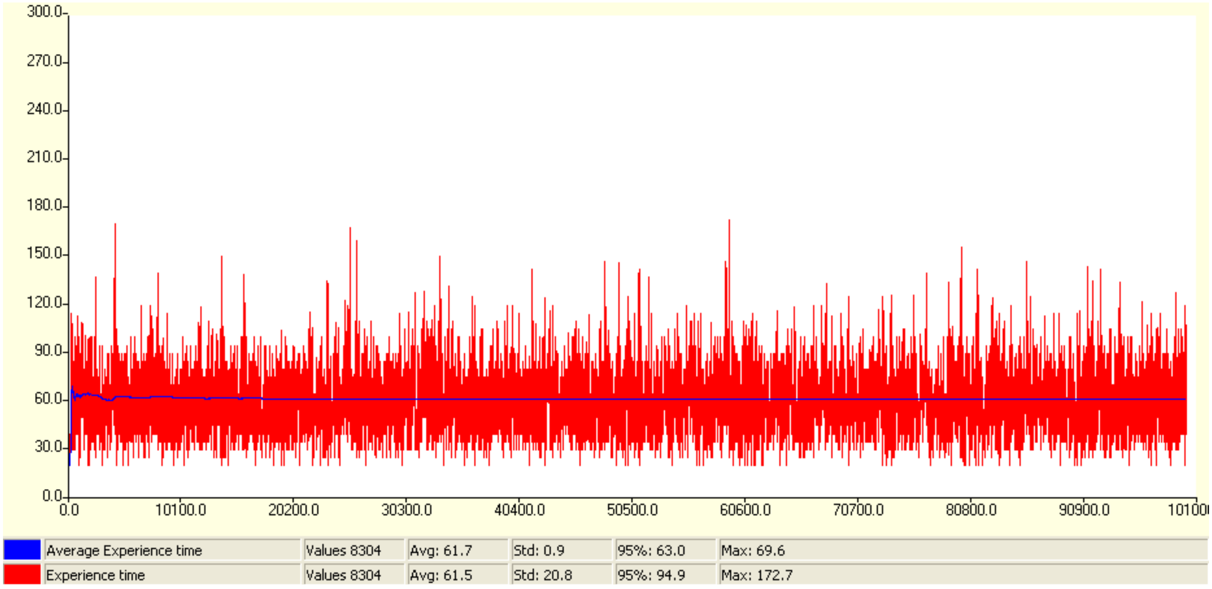


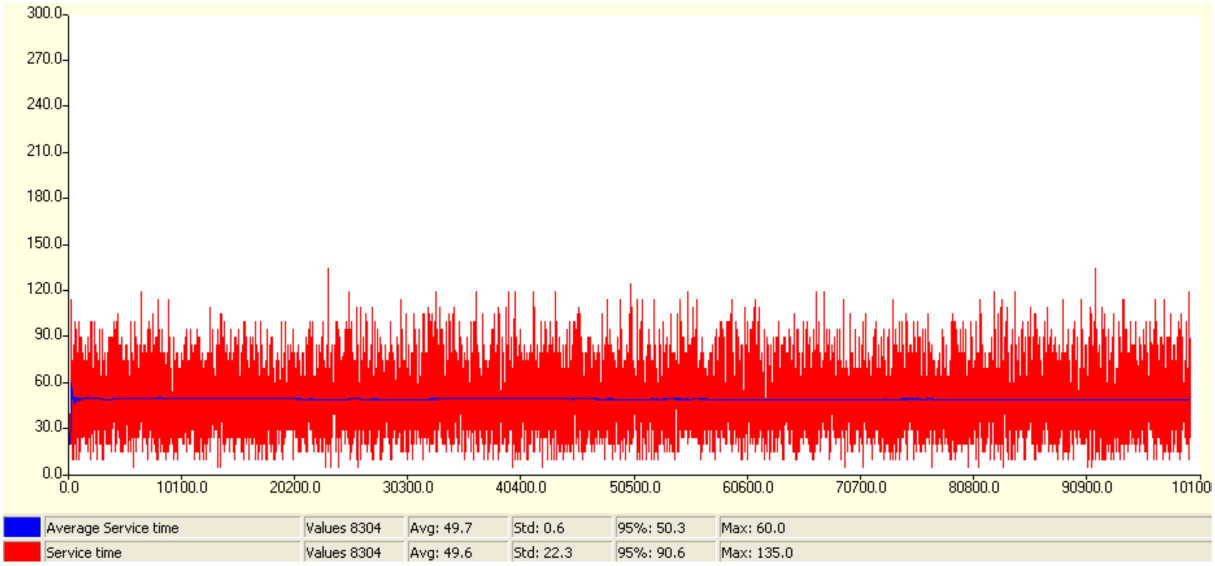
Five cashiers, one runner, two kiosks, one pick up point



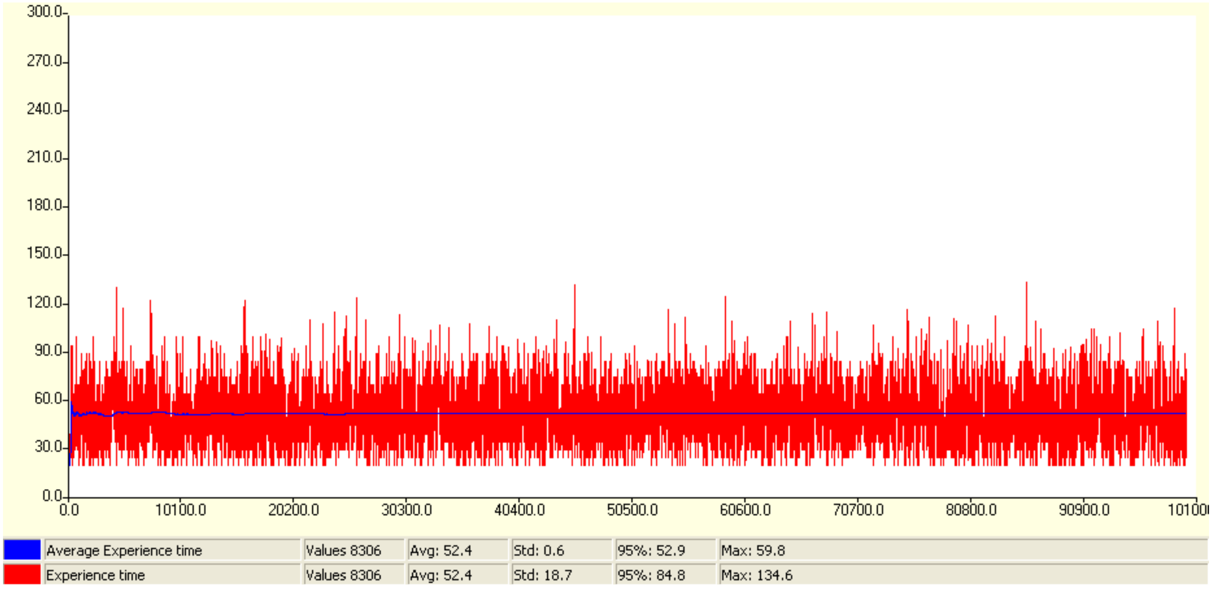


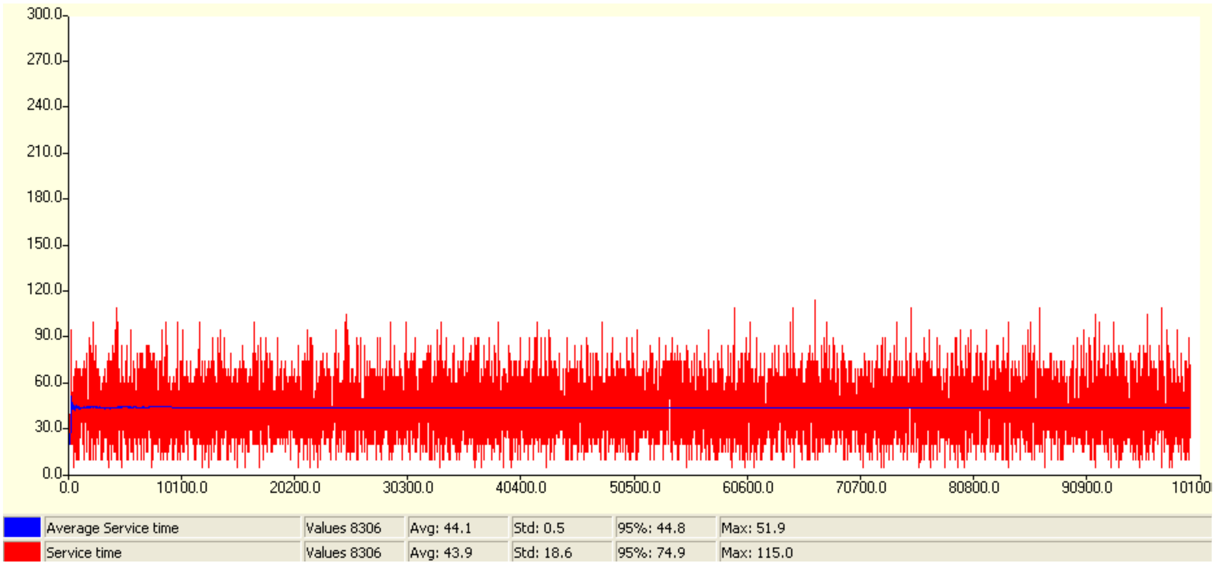
Four cashiers, one runner, two kiosks, two pick up points



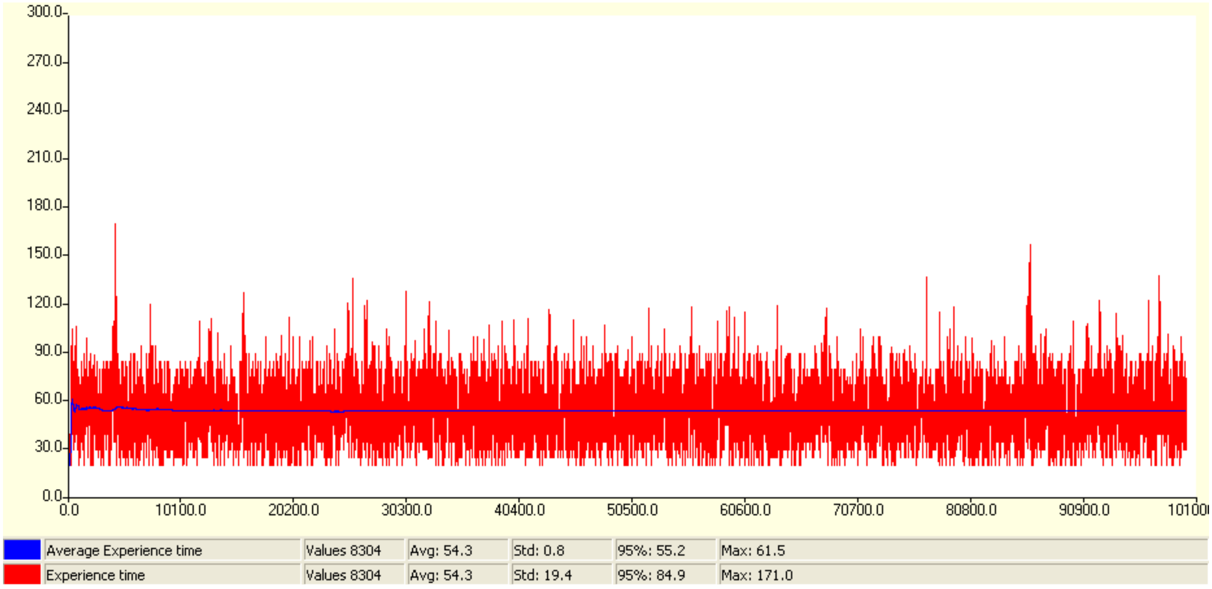


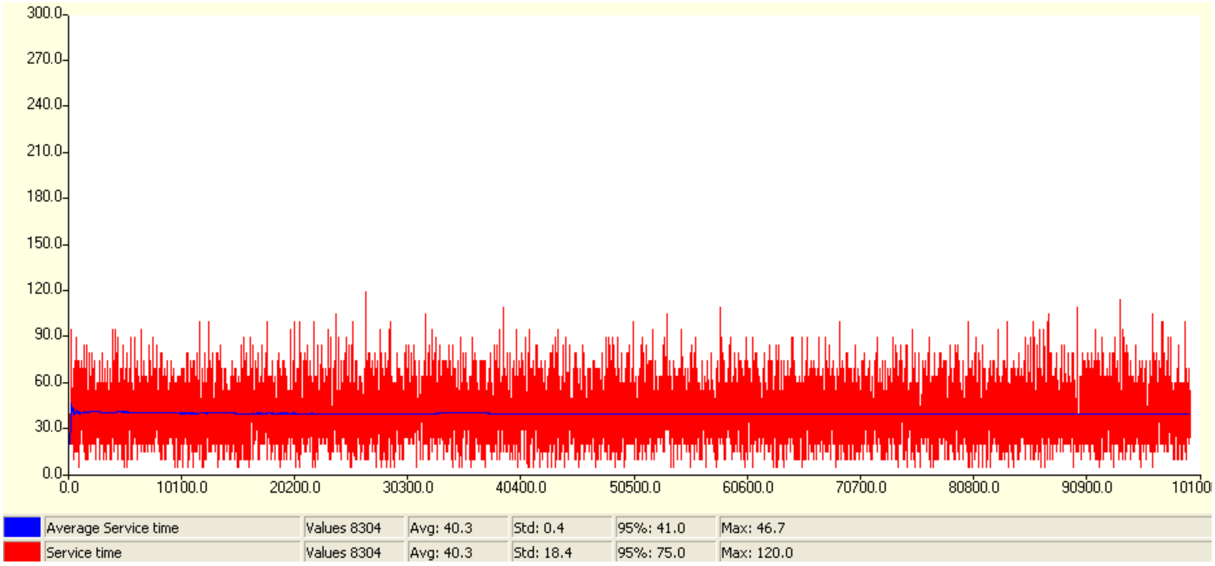
Four cashiers, two runners, two kiosks, one pick up point



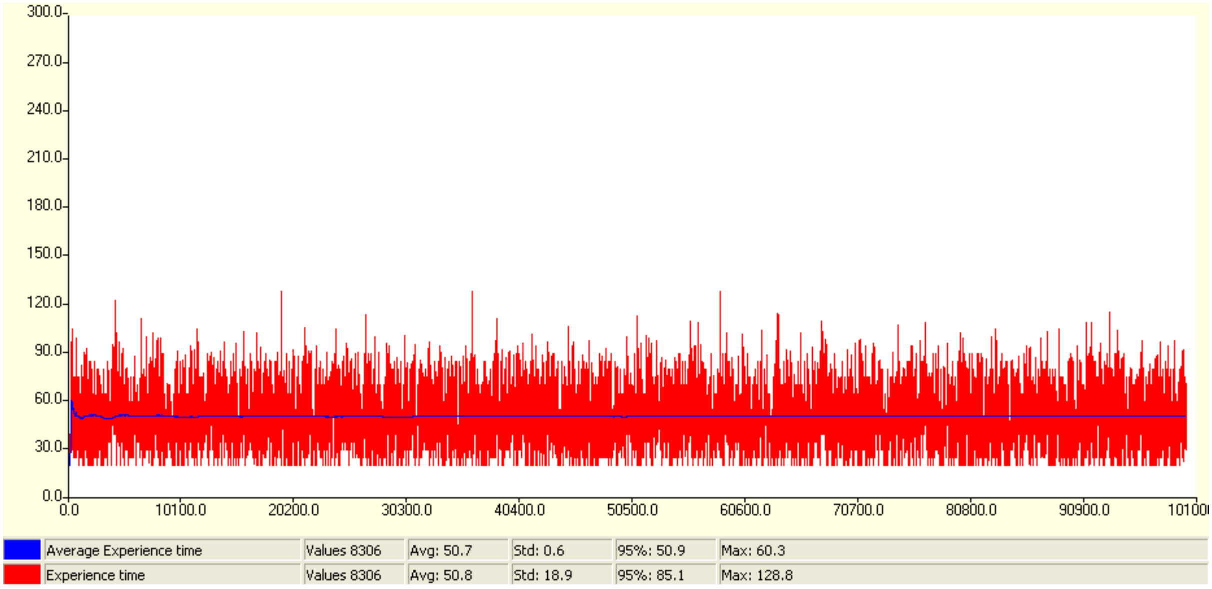


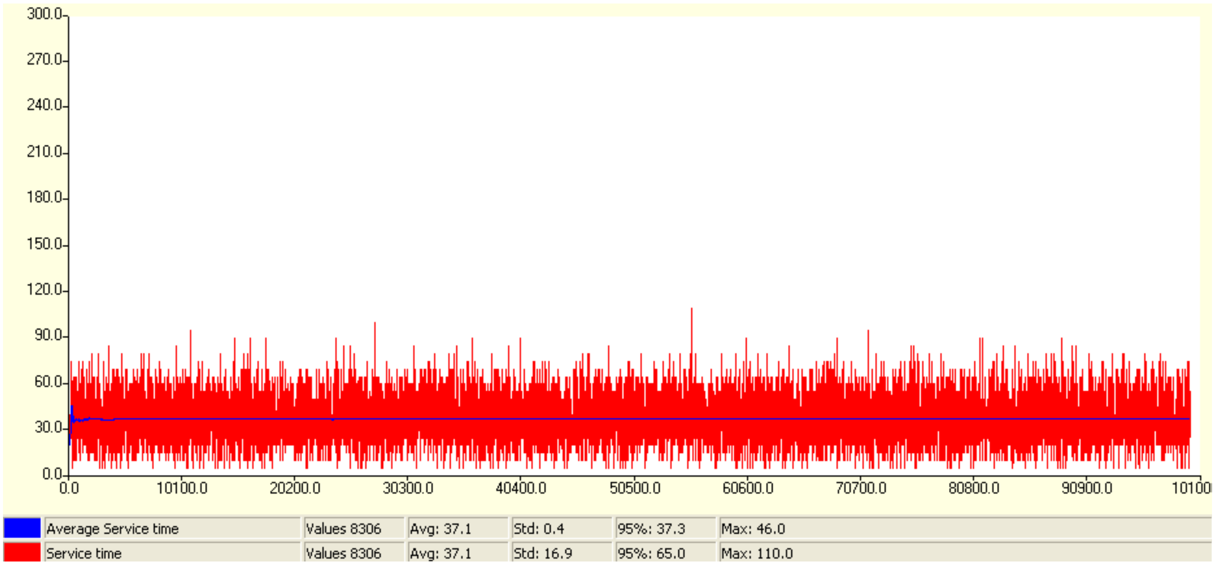
Three cashiers, two runners, two kiosks, two pick up points



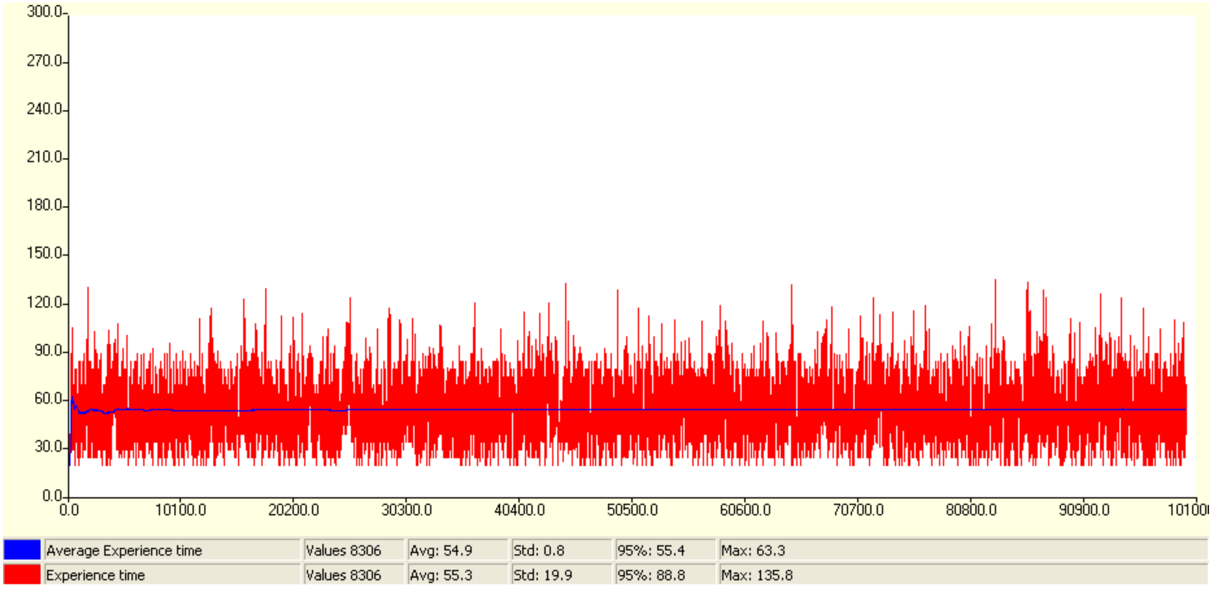


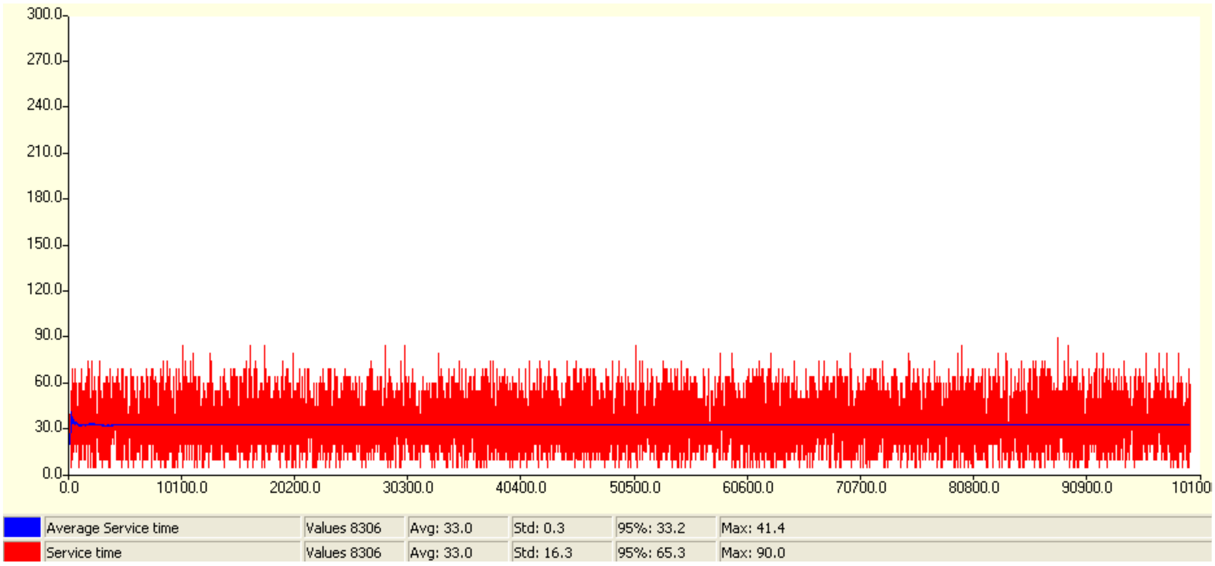
Three cashiers, three runners, two kiosks, one pick up point





Two cashiers, three runners, two kiosks, two pick up points





Results

# Cashiers	# Runners	#Kiosks	#Pick up points	Experience time (s) Average	Experience time (s) 95% quantile	Experience time (s) Maximum value	Service time (s)	Bring after (%)	Waste (%)
7	0	0	0	75.7	125.8	224.3	70	0.73	0.44
6	1	0	0	64.4	111.8	198.8	59	2.46	0.42
5	2	0	0	57.0	103.2	160.7	50	4.65	0.39
4	3	0	0	66.1	117.3	196.8	45	5.80	0.40
6	0	2	1	71.5	110.0	200.7	65	1.07	0.46
5	0	2	2	71.2	110.5	196.6	61	1.45	0.50
5	1	2	1	60.6	95.2	145.0	53	2.90	0.43
4	1	2	2	61.5	94.9	172.7	50	3.55	0.51
4	2	2	1	52.4	84.8	134.6	44	5.08	0.42
3	2	2	2	54.3	84.9	171.0	40	5.50	0.44
3	3	2	1	50.8	85.1	128.8	37	6.07	0.40
2	3	2	2	55.3	88.8	135.8	33	6.50	0.57