Simulation study for cash management in a bank branch

Project for the Simulation course held by Professor Alberto Ceselli,
University of Milan, July 2022

Alessio La Greca

Abstract

The management of a bank branch is, in general, a very complex task. Something that needs to be taken care of is the cash management inside of it. The cashiers and the branch manager (from now on, simply "manager") are in fact responsible, each day, of managing the cash registers of the branch. In the case we are interested in, we imagine that there are three kind of cash registers (that we'll simply call registers) in the branch: a Served register, where customers interact with cashiers to manage their money, a Withdrawal register and a Special register, which are both ATMs, where the first can be only used for withdrawals, while the second can be used also for deposits. All of this registers have a bounded quantity of cash that they can contain, and each working day must end with the registers respecting their constraints. In fact, if a working day is expected to conclude with too much or too little cash in one of those registers, a group of Security Guards (that, for short, we'll identify as"SG") must be notified, that the following day will come in the branch to take the excessive cash or to transport in the bank the missing one. The problem is that the transport service of the SG has a cost, so the manager would like to call them as little as possible. Another interesting thing would be understanding if, by changing the constraints on the cash that can be present every day in the branch, we would have a benefit in terms of calls made to the SG. Those constraints come both from the fact that the bank is insured against theft only for a given quantity of cash, and from the hardware specifics of the ATMs. The goal of this project is to simulate a bank branch in terms of customers, cashiers and manager, by using random variables to simulate the requests of the customers, specifically in terms of when they occour, what is their nature and the cash involved in the operation. The main objective of this study is understanding how many times the SG have to be called, to then analyze how the system would change given different parameters.

Contents

1 Introduction

Let's now give a more formal definition of our problem. Our simulation takes place in a bank branch, where our figures of interest are: the cashiers, the manager and the customers that come in the branch to to withdraw or deposit, independently of the acutal register they use. Now, the branch is open 5 days out of 7 every week: from monday to friday. However, on saturday and sunday, customers are free to use the ATMs of the branch. On the working days of the week, the cashiers and the manager work from 8.30 to 17.00 by doing different activities. We won't model all their activities in detail, just the ones of our interest for our particular problem.
Of our interest are three registers present in the branch:

1. The *Served register*. It is a register to which the cashiers can access to fulfill the requests of withdrawals or deposit of the customers. This register has some constraints on it: if x represents (in euros) the cash present in this register, by the end of a working day x must be such that: $min\_served = 50.000 \leq x \leq max\_served = 150.000$. 50.000 and 150.000 are just values given by an actual bank branch, but to not lose generality we've given variable names to them: *min_served* and *max_served*. Moreover, this register can contain bills of 5, 10, 20, 50, 100, 200 and 500 euros. However, while all of these bills can be deposited by a customer, during a withdrawal a cashier can never give a 200 or 500 bill to the customer. Those bills, even though they can influence the value of x, will have to, sooner or later, be handed over to the SG.

2. The *Withdrawal register*. It is an ATM from which one can only withdraw, not deposit. This register can have at maximum $max\_withdrawal = 180.000$ euros expressed as bills of either 20 or 50 euros, that can be given to the customers. It follows that each request of withdrawal to this register is expressed as $20a + 50b$, where (a, b) are nonnegative values, and the overall value is strictly greater than 0. Since money can only come out of this ATM, at least because of customers requests, it follows that sometimes the cashiers need to open it manually to fill it with bills. This operation is crucial so that, every day, there are enough bills such to satisfy all the possible requests coming from the customers. We also assume that the cashiers are able to query, whenever they want, the ATM, so they can know its balance (they basically always know the content of this register). This is an operation that the cashiers tend to not do often, and are willing to carry it out only when strictly necessary. We also assume that the ATM is always honest about its balance, even if this is not the case in real situations. Sometimes, in fact, the cashier needs to check if the balance returned from the ATM matches the actual quantity of money contained in it, but we won't model this aspect. To fill this machine, we assume that a cashier needs 20 minutes of its time. So, to recap: the cashier, in a certain moment of the day, before 16.40 (because we assume the branch closes at 17.00), queries the register, and if he thinks that the money contained are not enough to satisfy the requests of the following day, the register is opened, and filled with bills of 20 and 50 euros, in such a way that a value very close to *max_withdrawal* is reached (let's say, the value must be greater than $max\_withdrawal - 10.000$).

3. The *Special register*. This is an ATM that allows for both withdrawals and deposits. Also this has a maximum capacity, that is NOT measured in terms of actual money, but instead in terms of *number of bills contained*: this register can never have more than 2.000 bills inside of it, independently of their type (5, 10, 20...), otherwise it won't be able to accept any other bill. This register, that allows for both withdrawal and deposit operations, demonstrated over time that is used more for the latter than the former. This doesn't mean that, at the end of the day, no money have been withdrawn from it: more simply, it's to be expected that the quantity of money it contains will increase over time. This ATM too can be queried about its money, both the actual euros contained and the number of bills. It can also, of course, be opened by the cashiers, that need 60 minutes of time to empty it. The bills that one can deposit in this register are of 5, 10, 20, 50, 100, 200 and 500 euros, but just like before, only bills of 20 and 50 euros can be withdrawn.

All of those registers can change their values not only because of the operations of the customers, but also thanks to the intervention of the SG. They can in fact carry inside the branch, or outside from it, once a day (if needed), a sum of money x such that $x >= min\_sg = 60.000$. Such a safe transport service has indeed a cost, that is why we want to minimize the number of times those SG are called. In order to make them operate, the manager needs to call them at the start of the working day, before it opens to the customers, telling them how much money do they need to transport and in which way. The SG, by accepting the request, will fulfill it the following day, before the branch opens. The order of those operations is as follows: first, the SG called the day before arrive, to carry in/out some cash. Next, the manager calls, if necessary, the SG asking them to come the next day. Finally, customers start to arrive.
In all of this, we have to consider that the three registers are not completely independent: the cash present in one of them can be moved to another one, at the discretion of the cashiers. The objective is in fact to minimize the number of times in which the manager has to call the SG to transport some money. I think that, given those terms, it can be useful to model the SG as a tuple (input, output), that models the number of times in which the SG came to the branch to add some money and the number of times in which they came to carry out some cash, respectively. During the day, the three registers can change their values, but we have some constraints that must always be respected during any day:

1. The amount x of money present in the Served register must always be such that $min\_served \leq x \leq max\_served$

2. The next day, there must be enough money in the Withdrawal register in order to fulfill all the possible requests. Let's do an example: let's imagine that, on average, each day 40.000 euros are withdrawn from this register. If X is the current working day and, at its beginning, the manager estimates that at the end of the day there won't be more than 40.000 euros inside of this register, he calls the SG so that the day $X + 1$ they bring in the branch some money.

3. The next day, the Special register must be empty enough to host all the cash that will be deposited inside of it. We have already stated, in fact, that the deposit operation is in general much more requested than the withdrawal one, for this register. Experience tells that, on average, 20.000 euros are withdrawn and 35.000 are deposited each day. Let's take those values as an example. On average, 15.000 euros are deposited in this register, and they are divided in approximately 300 different bills. If the current day is X, the manager must make sure that the day won't end with more than 1.700 bills inside the register. Because, if it is so, the manager must immediately call the SG to ask them to come and take some of them during the day $X + 1$. When this ATM is emptied, a minumum amount of money (useful for possible withdrawals) must be left, like 15.000 euros (expressed in bills of 20 and 50 euros).

As we have described the last two constraints, seems like that the days on which the SG will come are predictable. This isn't however the case, because there is a technique that we can exploit: making the registers exchange their content. The idea is as follows: if at the end of a working day there is a register that has too much money, and another one that has too little, we can put some of the former in the latter, so that we don't make the SG intervene. This makes sense, considering how we presented the problem: the Withdrawal register will empty itself during the day, while the Special register will tend to fill up, and the Served register can either go up or down in content, there isn't a fixed law for that. With this idea, it is possible to adjust the registers' money without letting the SG intervene. That being said, there is a constraint that doesn't allow to exploit as we wish this technique: as stated before, to insert or remove cash from the ATMs, at least one cashier is required. However, since the cashier has usually a lot of other work to do during the day, he prefers to adjust the content of the ATMs **only when strictly necessary**. Basically, the Withdrawal register must be opened only when it really needs to be filled, in such a way that it can always fulfill the requests of the customers. Similarly, the Special register needs to be open only when it is necessary to remove part of the bills contained, so that it can accept all the deposit requests of the following day. To make an example: if the manager predicts, at the start of the day, that during the closing time there might be 200.000 euros in the Served register, and that the Withdrawal one will have 130.000 euros inside of it, he can't ask the cashiers to open the latter in order to move in it 50.000 euros coming from the Served register. In this case, he has no choice but to call the SG.
There are three main entities, that we can consider agents, in our system:

1. *The Manager.* At the start of a working day, the manager needs to estimate how much money each register will hold, trying to understand if it is possible to move some cash from

a register to another one by respecting the constraints mentioned previously. He also has to call the SG if necessary.

2. *The Cashier*: each cashier, from 8.30 to 13.00, needs to be at the desk in order to serve possible customers that might want to do a deposit or a withdrawal. After 13.00, the desk closes, and the cashiers only need to know if they have to open the ATMs in order to add or remove some cash. In doing this, they have to consider that people can always use the ATMs to withdraw or deposit.

3. *The Customer*. A customer is an individual that, during the day, will go the the bank branch, and make subsequent choices. As we will see, we'll model the arrival times of customers differently depending on the fact that they want to use either the Served register or one of the ATMs. In the first case, they will choose to do either a withdraw or a deposit. In the second, they choose the ATM to use. If it's the Withdrawal one, they'll do a withrawal. If instead it's the Special one, they will choose to do either a withdrawal or a deposit. Also consider that the Withdrawal and the Special registers (the ATMs) can be used at any time, even when there is no personal working at the branch, given that they are not being opened by one of the cashiers.

Another thing to consider is that the SG, when called to carry cash inside of the branch, will always handle it in terms of 20 and 50 bills. One last thing: whenever the cashiers need to work on both the two ATMs in the same afternoon, they will do them sequentially, no matter what. This is a fact that comes from observations of real systems of this kind. In particular, the Special register has higher priority than the Withdrawal one.

2 Symplifying assumptions and Random Variables

For our model, many aspects that deal with randomness can be modeled as Random Variables with a given distribution. What follows is a list of assumptions we did (in order to simplify some aspects of our modeling) and Random variables we chose to use with their meaning. Also, please note one important aspect: in order to produce meaningful results, we personally consulted some sample data of an actual bank branch, observed during some working days. In order to preserve privacy though, these data can't be made public, and well'often give as granted some probabilities or values, that come directly as an approximation of the observed data. Please, understand the privacy issues related and accept the given values as approximations of the real ones. In **some cases**, though, we'll also explain how we got some values, like the average, out of a sample.

2.1 Simplifying assumptions

From the observation of a real system of this kind we've extrapolated some facts that will simplify our modelling.

2.1.1 Cash imported in the branch by the SG

When there aren't enough money in the registers, the manager needs to call the SG that will come the following day with a certain amount of cash. But how is this organized, in terms of bills? Well, since the ATMs can only give bills of 20 and 50 euros to their customers, and also because they are the most preferred bills by the customers, the SG will always bring the cash in the following fashion. Say x is the amount of money they've been asked to carry in the branch:

- 80% of this cash must be expressed in terms of 50 euros bills. If $x = 100.000$, this means that 80.000 euros are in fact the sum of bills of 50 euros. In total, we would have $80.000/50 = 1.600$ bills of 50 euros.

- The remaining 20% of the cash has to be composed by bills of 20 euros. In our example where $x = 100.000$, we would have a total of $20.000/20 = 1.000$ bills of 20 euros.

2.1.2 When the ATMs are opened

We've already stated that the cashiers will tend to open the ATMs only when strictly necessary. We've also mentioned how they require some time to be filled or emptied: 20 minutes for the Withdrawal register, 60 minutes for the Special register. However, we assume that the cashiers will always tend to open them in the latest possible part of the day. This is because we also assume that the predictions of the manager regarding the content of the registers refer exactly to the closure time of the branch, namely 17.00. So:

- If only the Withdrawal one needs to be opened, the cashier will open it at 16.40.
- If only the Special one has to be opened, the cashier will start working on it at 16.00.
- If both need to be taken care of, the cashier will first open the Special one at 15.40, so that he can hold the money in excess. Part of those money can then be used to fill the Withdrawal one, that will be opened at 16.40.

2.2 Customers arrival: Poisson Process

Around the customer gravitate different random variables. The first thing that was noted during the analysis of the real system is that the customers that come to use the Served register are completely different from the ones that use the ATMs. This is because the Served register is opened for just a small fraction of the day, while the ATMs are basically always available, and also because the customers of the first kind are usually older people, that don't go very often to the bank. Given this, we've decided to model the arrival times of the customers as two distinct Poisson Processes: one for the Served register during its public opening hours, and one for the ATMs in general. Please note that, to be accurate, this should be modeled as a Nonhomogeneous Poisson Process, given the fact that, for example, during the first days of the month, a lot of elder people come to take the money of the retirement. Anyway, to simplify, we assume that this are Homogeneous Poisson Processes, with their parameter λ given by observations. Please note that our reference time unit will always be a single minute.

2.2.1 Homogeneous Poisson Process for customers going to the Served register

The process that models the arrival times of customers willing to use the Served register is modeled through a Homogeneous Poisson Process of parameter $\lambda = 0.15$. In such a way, it is to be expected that, on average, every 7th minute, a new customer will show up. This process takes place only between 8.30 and 13.00 of the working days, so between monday and friday.

2.2.2 Homogeneous Poisson Process for customers going to the ATMs

Also the process that models the arrival times of customers to the ATMs is modeled with a Homogeneous Poisson Process, where we have $\lambda = 0.21$. So, approximately every 5 minutes, a new customer will come to use either the Withdrawal register or the Special register. In this case, however, the parameter λ refers to a whole solar day: from 00.00 of day X to 00.00 of day $X + 1$. So we have $\simeq 300$ customers per day that use the ATMs.

2.3 Choice of a customer: Bernoulli Random Variables

Fixed the instants in which the customers arrive at the branch, we have to let them choose what kind of action they want to carry out. To do this, we employ three distinct Bernoulli random variables.

2.3.1 1° Bernoulli: Served register operation

A customer that goes to the Served register either does a withdrawal or a deposit. We model this with a Bernoulli R.V. X such that:

$$X = \begin{cases} \text{withdrawal} & \text{if } u \leq 0.7 \\ \text{deposit} & \text{otherwise} \end{cases}$$

where u represents a random number drawn from a uniform distribution between $[0, 1)$.

2.3.2 2° Bernoulli: Withdrawal register vs Special register

It may come as a surprise, but the Bernoulli R.V. that models the choice of the ATM chosen in order to carry out the desired operation is such that:

$$X = \begin{cases} \text{Withdrawal register} & \text{if } u \leq 0.5 \\ \text{Special register} & \text{otherwise} \end{cases}$$

This actually has a physical explanation: the Withdrawal one is closer to the sidewalk, and so, if two people come at the same time, they will most likely use the two ATMs in parallel, instead of having one of them waiting for the other one to complete. One could argue that this holds only when the two customers want both to withdraw. Yes, that is true, but the next Bernoulli will tell us that very few people deposit.

2.3.3 3° Bernoulli: Special register operation

The operation that a customer carries out when it is in front of the Special register can be modeled as:

$$X = \begin{cases} \text{withdrawal} & \text{if } u \leq 0.65 \\ \text{deposit} & \text{otherwise} \end{cases}$$

which might seem a contraddiction with respect to the introduction, where we stated that, in the Special register, we have more deposits than withdrawals. Well, this holds, but in terms of money, not in terms of performed operations. We'll see that, on average, a deposit is much bigger that a withdrawal, in this ATM.

2.4 Withdrawals and Deposits: Normal Random Variables

The observed values for the withdrawal and deposit operations showed that a Normal Distribution can be a good approximation of the values of the money involved in such operations. Before presenting the values obtained, let us mention the approach taken in order to measure the mean μ and the variance σ^2 from our samples of withdrawals and deposits.
First of all, different sample sets were collected. Let's say that each of them contained n elements. We have considered:

- A sample for the withdrawals from the Served register.
- A sample for the deposits to the Served register.
- A sample for the withdrawals from the Withdrawal register.
- A sample for the withdrawals from the Special register.
- A sample for the deposits to the Special register.

In order to estimate the actual mean μ and variance σ^2 (or, equivalently, the standard deviation σ), we used two estimators: the Sample Mean and the Sample Variance.
The sample mean is an estimator that comes from the literature and can be easily demonstrated that it is both unbiased and reliable, which mean, respectively, that its expected value is the parameter it tries to estimate, μ, and that the more samples we have, the greater the accuracy of estimating our target parameter μ. If $\overline{X}$ represents our sample mean, it can be calculated as:

$$\overline{X} = \sum_{i=1}^{n} \frac{X_i}{n}$$

While instead, if S^2 represents the sample variance that we get out of our sample, it can be shown that this variance approximator is unbiased when it is computed as follows:

$$S^2 = \frac{\sum_{i=1}^{n}(X_i - \overline{X})^2}{n-1}$$

Where the sample standard deviation is simply computed as $S = \sqrt{S^2}$.
Another thing to note is that, by using the Normal Distribution to approximate the distribution of the withdrawal or deposit operations, we are exposing ourselves to the possibility that sometimes a user withdraws or deposits a negative number of euros. To solve this, we will use the Truncated Normal distribution that takes values only in the interval $[0, +\infty)$. Very well. We can now show the values we computed out of our samples.

2.4.1 1° Normal: withdrawals at the Served register

For the withdrawals at the Served register we have:

$$\mu = 918.2, \sigma = 606.6$$

2.4.2 2° Normal: deposits at the Served register

For the deposits at the Served register we have:

$$\mu = 1801, \sigma = 1765.5$$

2.4.3 3° Normal: withdrawals at the Withdrawal register

For the withdrawals at the Withdrawal register we have:

$$\mu = 357.8, \sigma = 476.3$$

2.4.4 4° Normal: withdrawals at the Special register

For the withdrawals at the Special register we have:

$$\mu = 329.9, \sigma = 287.4$$

2.4.5 5° Normal: deposits at the Special register

For the deposits at the Special register we have:

$$\mu = 1705.5, \sigma = 2841.4$$

2.5 Management of the bills: Custom Random Variable

We have already explained how it is important to model not only the quantity of the money present in each register, but also how they are expressed in terms of bills. The main difference is between the withdrawals and the deposits, but it's not the only one. In this case, let's start with the ATMs, instead of the Served register.

2.5.1 1° Custom: withdrawals at the Withdrawal and Special register

The ATMs can always give money to their customers only in the form of 20 and 50 euro bills. That said, what an ATM actually tries to do is to balance the number of 50 euro bills given with the number of 20 euro bills given. In particular, what we propose is an algorithm for generating the given bills based on a Bernoulli random variable. Please note that the money requested by a customer will always be expressed as $50a + 20b$, with (a,b) being nonnegative values such that the overall sum is ≥ 0.

2.5.2 2° Custom: withdrawals at the Served register

2.5.3 3° Custom: deposits at the Served and and Special registers

Algorithm 1 Bills given by an ATM

```
$tot$ = requested money
$bills\_50 = 0$
$bills\_20 = 0$
while $x > 90$ do
    $u$ = uniformly distributed random number in [0,1)
    if $u \leq 0.6$ then
        $X = 50$
        $bills\_50 = bills\_50 + 1$
    else
        $X = 20$
        $bills\_20 = bills\_20 + 1$
    end if
    $tot = tot - X$
end while

if $x = 90$ then
    $bills\_50 = bills\_50 + 1$
    $bills\_20 = bills\_20 + 2$
end if
if $x = 80$ then
    $bills\_20 = bills\_20 + 4$
end if
if $x = 70$ then
    $bills\_50 = bills\_50 + 1$
    $bills\_20 = bills\_20 + 1$
end if
if $x = 60$ then
    $bills\_20 = bills\_20 + 3$
end if
if $x = 50$ then
    $bills\_50 = bills\_50 + 1$
end if
if $x = 40$ then
    $bills\_20 = bills\_20 + 2$
end if
```