UTRECHT UNIVERSITY

BACHELOR THESIS
BSC ARTIFICIAL INTELLIGENCE

How to Find the One

Secretary Problem Extended with Uncertain Observations

Author:
C.P. (Tamara) Florijn
Student number:
5856442

Supervisor:
Rosalie Iemhoff
Second corrector:
Cassio P. de Campos

A 7.5 ECTS thesis submitted in fulfillment of the requirements
for the degree of Bachelor of Science

in the

Faculty of Humanities

June 28, 2019

Abstract

C.P. (Tamara) Florijn

How to Find the One
Secretary Problem Extended with Uncertain Observations

Making a good decision is often a challenge. How to choose your ideal house? How to find your true love? These problems resemble the mathematical puzzle called the secretary problem.

In the classic secretary problem, administrators want to hire the best secretary from n applicants. The secretaries present themselves one by one. Each decision, rejecting or accepting, needs to be immediate, is permanent, but can only be based on the ranking of seen applicants. Who should they hire?

The optimal strategy is a cut-off strategy, where the number of applicants one needs to wait is $\frac{n}{e}$, with n is the number of applicants, which is approximately 37 with $n = 100$. After 37%, the administrators should hire the first who is better than all they have seen before.

In this study, the effect of uncertain observations on the chance of success of cut-off strategies has been evaluated by the use of computer simulations. Computer simulations were used to simulate versions of the secretary problem with intervals, normal distributions and pareto distributions. The trend showed that the larger the uncertainty, the lower the cutoffpoint from the optimal strategy. The reserved advice is: if you are not sure, choose sooner!

Keywords: Secretary problem, Googol problem, best choice, uncertainty, computer simulations, optimization

Contents

Chapter 1

Introduction

Making a decision is both difficult, challenging and important. Choosing is part of our daily life. How to find that one perfect house when accepting and refuting an offer is permanent? How to find your true love? How to find the one and only? These examples have a few characteristics in common. For example, the possibilities to choose from appear sequentially, refuting and accepting an offer is permanent, and there is no information available about coming options. These situations can be modelled as a mathematical puzzle called the secretary problem.

In the classic secretary problem, administrators want to hire the best secretary of n applicants (note, second best is regarded as a failure). After each job interview, they can decide whether they will hire the applicant or not. Once rejected, there is no way of returning. During the interviews, they are able to make a ranking of the seen applicants without ties, but are unaware of the quality of unseen applicants. The decision needs to be made immediately. Who should they hire?

The optimal strategy for the original secretary problem has been proven elegantly by Gilbert and Mosteller (1966), which will be shown in the chapter 2. After 37 % of the applicants, the administrators should hire the first who is better than all seen before. The optimal strategy is a cutoff strategy, where the number of applicants one needs to wait is $\frac{n}{e}$, with n is the number of applicants, which is approximately 37 with $n = 100$.

The secretary problem was first published in a column by Martin Gardner (1966) in the *Scientific American*, introduced as "the Game of Googol". In this game, one person may write down any number on slips of paper. The other person may turn the slips one by one, and aims to stop at the highest number. Lindley (1961) was the first to publish the solution, and he extended it to an arbitrary utility. In 1966, Gilbert and Mosteller published a paper with the optimum values and their probabilities of winning for a range of values of n, and discussed, among other, the variation of being allowed to try and choose more than once. In reaction to this paper, a lot of articles and journals have been published. Many variations have been discussed, including an infinite number of applicants (Gianini and Samuels, 1976) or finding the second best applicant (Vanderbei, 1983). Others reviewed versions with more information, for example where secretaries where chosen from a normal distribution with an unknown mean (DeGroot, 1968). Every one of the six assumptions has been altered to solve a different form of the original secretary problem.

In addition to these, Freeman (1983) published a clear and elaborate overview of extensions that had been researched up to that time, including a version with an unknown number of items, an uncertain employment, and an unknown recall. A few years later, Ferguson (1989) answered the question "Who solved the secretary problem?" by providing an overview of the history, and constructing the proof himself. Gnedin (1994) investigated the problem with absolute values (game of Googol) in comparison to the problem with a ranking , and proved that the optimal solution is a rule only based on relative ranks for $n > 2$. Seale and Raport (1997) conducted an experiment to research human behaviour in this setting. They used computer simulations to evaluate the found strategies and compared them to the optimal strategy.

"All human knowledge is uncertain, inexact, and partial," Bertrand Russsel once said. In real life, making a choice involving uncertainty is an everyday task. Uncertainty did not gain much importance in the original secretary problem, other than not knowing anything about who will come next and with what quality. The certain knowledge about the absolute ranking could be considered as unrealistic. Some of the articles published offered formulations to come closer to a realistic setting, such as cooping with an unknown number of applicants (Seale and Rapoport, 2000), or modelling human heuristics (Seale and Rapoport, 1997).

This study will not attempt to model human behaviour. Instead, we will create a situation that might come a step closer to reality by involving uncertainty, and research the optimal strategy in that scenario. This uncertainty and decision making connects with challenges the field of artificial intelligence, such as the design of autonomous decisive agents. Uncertainty, chaos and noise in observations will be crucial to deal with in order to operate safely and effectively in the real world. To bridge the gap between ideal mathematical models and the chaotic real world remains a challenge.

The aim of this paper is to investigate the influence of the amount of uncertainty in the observations of the secretary problem to evaluate the effect on the cutoffpoint of the strategy using computer simulations. Whereas the original secretary problem used absolute values to grade the secretaries, we will let uncertainty have an influence. The uncertainty will be reflected in the observations of the secretaries in the form of intervals, normal distribution and pareto distribution, where the values of the secretaries will be chosen from randomly. Intuitively, we expected that the exploration phase (before the cutoffpoint) would now require more time, to gather enough information about the population. This suggested that the cutoffpoint would have increased when more uncertainty was added.

This study will begin with theoretical background, where we provide the proof for the optimal strategy of the classic secretary problem and lay the mathematical foundations for the concepts used in the methodology and results. After that, we will elaborate on the design of the computer simulations in chapter 3, methodology. In chapter 4, we will present the results of the computer simulations and give an interpretation of the observations we made. Chapter 5 will conclude the study, where we answer the research question and discuss the applications of the results.

Chapter 2

Theoretical background

In this chapter, several mathematical concepts will be introduced, as concise as possible while remaining correct and sufficiently elaborated. At first, we will enlist terminology that will be used in the rest of this study. At second, we will explain the proof of the optimal strategy for the classic secretary problem, step by step. At third, we will briefly introduce mathematical concepts concerning the implementation of uncertainty, namely intervals, the normal distribution and the pareto distribution. This will lay the theoretical basis necessary to gain insight in the choices made in the methodology, and the interpretation of the results.

2.1 Terminology

This section will give a short overview of the meaning of abbreviations and terms that will be used.

The secretary problem will be abbreviated as *SP*. This will refer to the mathematical puzzle as has been explained in the introduction. *DM* is short for the Decision Maker. The DM is the metaphorical person who decides which secretary is chosen. The secretaries are presented to the DM one by one. The *number* of the secretary will refer to the order in which this occurs. The first secretary has number 1, the second number 2, till the last one with number n, the total number of secretaries. The aim of the SP is to find the best secretary, which means that each secretary better than all the secretaries the DM has seen before, might be the one he is looking for. These secretaries are called a *candidates*. The DM chooses according to a cutoff strategy. After a certain number of applicants, the *cutoffpoint*, the DM will choose the first candidate, so the first secretary who is better than all seen before.

2.2 Proof optimal strategy

As has been said in the introduction, the optimal strategy for the classic secretary problem is a strategy with a cutoffpoint of 37% of the applicants. This problem has a surprisingly simple and elegant solution. The here provided proof is extended from the proof in the article from Ferguson, 1989. We will use n as the number of applicants, i as the number of the selected secretary and $r - 1$ is the cutoffpoint (the decisionmaker (DM) chooses the first one better than the first $r - 1$ applicants). We construct the probability of selecting the best of n applicants, and will derive the cutoffpoint that optimizes the chance of success.

Probability of selecting the best of n

$$
\begin{aligned}
&= \sum_{i=1}^{n} P(\text{i selected} \cap \text{i is the best}) \\
&= \sum_{i=1}^{n} P(\text{i selected} \mid \text{i is the best}) \cdot P(\text{i is the best}) && [P(A \cap B) = P(A|B) \cdot P(B)] \\
& && (Conditional probability) \\
&= \sum_{i=1}^{r-1} 0 + \sum_{i=r}^{n} P(\text{i selected} \mid \text{i is the best}) \cdot \frac{1}{n} && [P(\text{is the best}) = \frac{1}{n}] \\
&= \sum_{i=r}^{n} P(\text{the best of i}-1 \text{ is in r}-1 \mid \text{i is the best}) \cdot \frac{1}{n}
\end{aligned}
$$

The number of the best of the first *i−1* secretaries should be lower than r−1 (before the cutoffpoint). This secretary may be the first (chance $\frac{1}{i-1}$), the second (chance $\frac{1}{i-1}$), till *i−1*. If we add up all the chances, we obtain $\underbrace{\frac{1}{i-1} + \frac{1}{i-1} + \cdots + \frac{1}{i-1}}_{\text{r-1 times}} = \frac{r-1}{i-1}$.

$$
\begin{aligned}
&= \left[\sum_{i=r}^{n} \frac{r-1}{i-1} \right] \cdot \frac{1}{n} \\
&= \left[\sum_{i=r}^{n} \frac{r-1}{(i-1) \cdot n} \right] \\
&= \frac{r-1}{n} \cdot \left[\sum_{i=r}^{n} \frac{1}{i-1} \right]
\end{aligned}
$$

Letting n tend to infinity, we will write x as the limit of $\frac{r-1}{n}$, use t as $\frac{i-1}{n}$ and dt as $\frac{1}{n}$. $\frac{1}{i-1} = \frac{n}{i-1} \cdot \frac{1}{n} = \frac{1}{t} \cdot dt$. Thus, the sum now can be approximated using the integral

$$x \int_{x}^{1} \frac{1}{t} dt$$

The boundaries from x to 1 can be explained by rewriting t. Because x is the limit of $\frac{r-1}{n}$ and $t = \frac{i-1}{n}$, one can derive $i = r$ from $t = x$, which corresponds with the sum we approximate. The upper boundary of the integral is 1. With $n \to \infty$, it follows from $t = 1$ that $\frac{i-1}{n} = \frac{i}{n} - \frac{1}{n} \approx \frac{i}{n}$. Given $\frac{i}{n}$=1, it follows that $i = n$, which corresponds with the sum we approximate.

$$
\begin{aligned}
&x \int_{x}^{1} \frac{1}{t} dt \\
&= x \cdot (\ln(1) - \ln(x)) \\
&= x \cdot (0 - \ln(x)) \\
&= -x \cdot \ln(x)
\end{aligned}
$$

We want to maximize the chance of success, by taking the derivative with respect to x, and setting it to *0*. The derivative of $-x \cdot \ln(x)$ is:

$$\begin{aligned}
&-(1 \cdot \ln(x) + x \cdot \frac{1}{x}) \\
&= -\ln(x) - 1 \\
&= 0 \\
&\Rightarrow \ln(x) = -1 \\
&\Rightarrow x = \frac{1}{e}
\end{aligned}$$

x was the limit of $\frac{r-1}{n}$, thus $\frac{r-1}{n} = \frac{1}{e}$. It follows that the cutoffpoint $r-1$ should be $\frac{n}{e}$.

2.3 Mathematical background

Discrete uniform distribution (intervals)

A discrete uniform distribution is characterised by a lower boundary l_b, an upper boundary u_b, and an interval size $ivs = u_b - l_b$. All the discrete values in the interval are equally likely to be selected. The expected value is the middle of the interval, $E(X) = l_b + \frac{1}{2} \cdot ivs$.

The larger the interval, the less information one has about the value that will be chosen, because there are more values possible. The interval size therefore defines the amount of uncertainty.

Normal distribution

The top of the normal distribution is given by the mean and the amount of dispersion is indicated by the standard deviation σ (sigma).

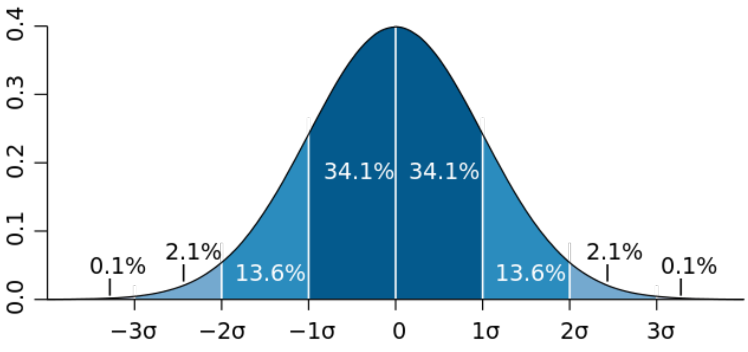

FIGURE 2.1: Normal distribution (made by W. Toews, 2007).

As can be seen in figure 2.1, 68.2 % falls into the interval [mean $-\ \sigma$, mean $+\ \sigma$]. The smaller the standard deviation, the more the data points tend to be close the mean.

The standard deviation therefore defines the amount uncertainty, while the mean defines the shift of the figures, in other words the location of the top. The expected value is the mean, for the distribution is symmetric.

Pareto distribution

The parameters of the pareto distribution are x_m and α (alpha). In figure 2.2, one can see that the chance of the value that is generated from the distribution is large when it is close to x_m, and small when it is further away. How large and how small, is defined by α. The larger α, the higher the starting point of the graph, so the bigger the chance is that the generated value is close to x_m. In other words, the larger the alpha, the smaller the uncertainty (note, this is the other way around in comparison to the other two distributions). The mean of a pareto distribution is given by $\frac{\alpha \cdot x_m}{\alpha - 1}$ for $\alpha > 1$.

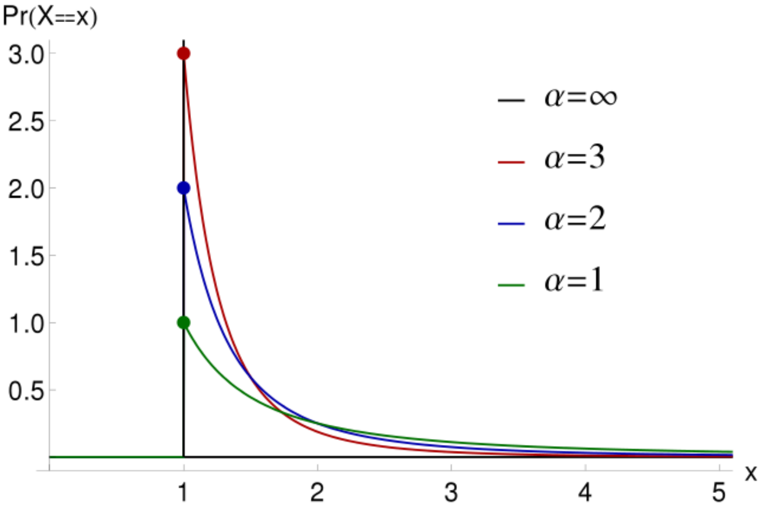

FIGURE 2.2: Pareto distribution (made by Danvildanvil, 2014).

An example will illustrate the role of α. The probability density function of the pareto distribution is $\frac{\alpha \cdot x_m^\alpha}{x_m^\alpha}$. This can provide an relative likelihood that the value (x) is randomly chosen. Suppose, $x_m = 400$.

$$\text{When } \alpha = 10, f(x) = \frac{10 \cdot 400^{10}}{400^{11}} = \frac{10}{400} = \frac{1}{40}.$$
$$\text{When } \alpha = 50, f(x) = \frac{50 \cdot 400^{50}}{400^{51}} = \frac{50}{400} = \frac{1}{8}.$$
$$\text{When } \alpha = 100, f(x) = \frac{100 \cdot 400^{100}}{400^{101}} = \frac{100}{400} = \frac{1}{4}.$$

Thus, the chance that the generated value is close to x_m, increases when α increases. The larger α, the smaller the uncertainty about the generated value.

Chapter 3

Methodology

We conducted computer simulations to research the influence of uncertainty on the optimal strategy of SP. Analytical mathematical methods have been used to proof the optimal solution for the original SP. Including uncertainty limited the use of analytic proving methods. The events occur probabilistic, and therefore cannot be described with differential equations directly. By involving uncertainty in the observations, the values of the secretaries can not be determined exactly, but the values can be generated randomly. Each strategy can be evaluated and approximated to find the optimal solution. In this chapter, we will make clear how the computer simulations are developed, and elaborate on the choices we made to create it.

3.1 Design

Computer simulations were conducted to assess the effectiveness of the cutoff strategies in the SP. The code of the C# implementation is included in appendix C, and we will here summarize its design. In a simulated secretary problem, the DM receives the number of secretaries (n) that will be presented. The set of applicants is constructed using a random generator. One by one the value and the number of the secretary at that moment is presented, along with the last seen candidate. The DM is given a strategy in the form of a cutoffpoint, 1 to n. Once an applicant is seen who is a candidate and has a number higher than the cutoffpoint, the applicant is chosen. If the DM has chosen, the rest of the secretaries are generated. The highest value of all applicants is compared to the chosen value of the DM. If these are the same, the round is saved as a win; if that is not the case, it will be saved as a loss. Aside from that, there is another case where the DM has not been able to find anyone after the cutoffpoint. Then, the best was already presented before the cutoffpoint and no one was chosen by the DM. That is gathered under the name of losses, but can be split when counting the amount of selections, whether a selection has been made or not.

In the adapted SP version, we applied uncertainty to the observation. The DM receives the same sort of information as before, now called the *apparent* value, while the *actual* value is generated "behind the scenes" with a certain distribution, namely intervals, normal distribution or pareto distribution. The actual values are generated as following. A random value is generated between 0 and a boundary, that value is for example 800. In the case of intervals, when the DM encounters the value 800 for a secretary, the actual value is randomly chosen from the interval $[800, 1000], [800, 1200]$, or some other, dependent on the size of the interval. In the case of a normal distribution, the value is randomly chosen from a normal distribution with 800 as mean, and a given standard deviation such as 10 or 20. In the case of pareto distribution, the value is chosen from a pareto distribution with 800 as x_m, and a specific α, such as 8 or 12. For all the distributions applies: the larger the uncertainty, the larger the chance that the actual value deviates from the apparent value. The DM still chooses in the same manner as before, namely that he will choose the first candidate after the cutoffpoint, but he will only base this decision on apparent values. In the end, the actual value is used to evaluate who is the best, and whether the round will be saved as a win or a loss.

3.2 Choices made

Distributions

To model uncertain observations, we have evaluated three distributions. At first, a specified interval under a discrete uniform distribution is a quite standard way of representing deviation. When asked: "What grade do you expect to receive for your test?", one might answer: "Well, between 8 and 9." Intervals will be used to reflect this sort of uncertainty in the observations. At second, normal distributions are often used to approximate quantities we find in nature, such as people's heights and examination grades (Lyon, 2013). We might have a reasonably good idea of the quality of a job applicant, but it is possible that the applicant is much better or much worse, although the chance is small. This is reflected when using a normal distribution. At third, pareto distributions could be known from the 80–20 rule. For example it has been said that 80 % of the wealth is owned by the richest 20% of the people (M. Levy and H. Levy, 2003). Though this 80–20 rule is only shown with a specific alpha ($\log_4 5 \approx 1.16$), the intuition that more people will show a lower capability, and only a few belong to the top, is reflected over the variety of values of α in the pareto distribution. In short, the combination of these three distributions provide a variety of interpretations of what uncertainty in the observation of a secretary entails.

Number of applicants

The number of applicants we choose has been based on the numbers used in Seale and Rapoport, 1997. $n = 40$ and $n = 80$ will be used throughout the rest of this study. To support the decision of using these two as values for n, we compared several numbers of applicants, and plotted them scaled to 100 % shown in figure A.1 in the appendix. Most of the graphs show almost exact similarity. The difference one can notice is small enough to attribute it to stochastic irregularities. The graph of $n = 20$ shows the largest deviation. This is due to the fact that 20 is a small number. The optimal strategy $\frac{n}{e}$ is an approximation for small numbers, as explained in chapter 2. The optimal cutoffpoint is $\frac{n}{e}$ when n approaches infinity, and deviates from that point for smaller numbers. For $n = 20$, the optimal cutoffpoint is *8*, and optimal probability of winning is 0.384 (Gilbert and Mosteller, 1966), which explains the deviation in the graph. In short, the number of applicants used will be $n = 40$ and $n = 80$, which should have minimal effect on the results.

Number of rounds

We ran the simulation 100,000 times for each cutoffpoint, and varied the cutoffpoint between 1 and n. This is ten times as much as similar SP simulations (Seale and Rapoport, 1997). This number of rounds shows more or less stable graphs, with a smooth line, which suggests that the results approximate the actual value. If the number of rounds is large enough, we may assume that the value is converged. Still, it will remain an approximation, but should not have a large effect on the results.

Boundary

The boundary of the value of the secretary should be chosen in such a way that it does not affect the results. It should not be too small, because then there will be too many duplicates in the generation of random values. As an extreme example, suppose the boundary is 1, then the first one is best, but the second as well, and the third, and so on. When the boundary is large enough, then the chance of the values being the same will be minimised enough to not influence the results significantly. The smoothness, stability and the similarities between the graphs in figure A.2 in the appendix indicate that the choice of the boundary is justified. We tested the boundaries for the SP with uncertainties in figures A.3 to A.5. To evaluate the boundaries correctly, we adapted the associated parameter accordingly. When the boundary is multiplied by 10, so the random value r, then the expected value should also be multiplied by 10.

For intervals, with r the random generated lower border and ivs the interval size, the expected value is $E(x) = r + \frac{1}{2} \cdot \text{ivs}$. Let $r' = 10 \cdot r$.

$$\begin{aligned}
E(x') &= 10 \cdot E(x) \\
&= 10 \cdot (r + \frac{1}{2} \cdot \text{ivs}) \\
&= 10 \cdot r + \frac{1}{2} \cdot 10 \cdot \text{ivs} \\
&= r' + \text{ivs}' \\
&\Rightarrow \text{ivs}' = 10 \cdot \text{ivs}
\end{aligned}$$

This can be seen in figure A.3 in the appendix.

For a normal distribution, the expected value is the mean, which is the random generated value r. To keep the dispersion in ratio to the enlarged mean, we also keep the standard deviation in ratio. When the boundary is multiplied by 10, also the standard deviation is multiplied by 10. To show that these will have similar distributions, the results for different variations are visually depicted in figure A.4.

For a pareto distribution, the parameter α should remain constant, regardless of the value of the boundary. The expected value of a pareto distribution is given by $\frac{\alpha \cdot x_m}{\alpha - 1}$ for $\alpha > 1$. x_m is the random generated value. When x_m is multiplied by 10, the expected value should also be multiplied by 10. Let $x'_m = x_m \cdot 10$.

$$\begin{aligned}
E(x') &= 10 \cdot E(x) \\
&= 10 \cdot (\frac{\alpha \cdot x_m}{\alpha - 1}) \\
&= \frac{\alpha \cdot 10 \cdot x_m}{\alpha - 1} \\
&= \frac{\alpha \cdot x'_m}{\alpha - 1} \\
&\Rightarrow \alpha' = \alpha
\end{aligned}$$

This can be seen in figure A.5 in the appendix.

Chapter 4

Results

In this chapter, we will show the results of the computer simulations. We varied the cutoffpoint, and studied the differences in the number of correct selections and the number of selections. We will evaluate the classic secretary problem, the secretary problem influenced by uncertainty, and provide an interpretation of the observations made.

4.1 Results original SP

The cutoff decision rule has been evaluated for the standard SP with $n = 40$ and $n = 80$. Each possible variant of the cutoff strategy (1 to n) was simulated 100,000 rounds. The black line depicts the number of correct selections, that has been counted per cutoffpoint. The red line depicts the probability of selection, the chance that the DM has found a candidate after the cutoffpoint, also measured per version of the strategy.

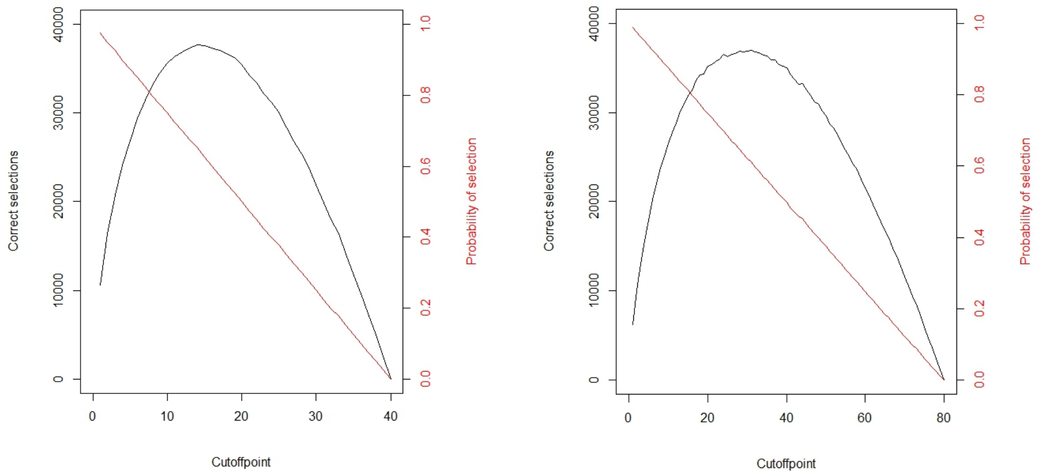

FIGURE 4.1: Graphs original SP with n = 40 and n = 80.

For $n = 40$, the optimal strategy is to reject the first 15 applicants, and then hire the first who is better. This strategy results in 0.376 chance of correct selection (Gilbert and Mosteller, 1966). Our results for the standard SP for $n = 40$ (left graph in fig. 4.1) resemble the theoretical predictions. In our simulation, the top is reached with cutoffpoint 14, with 0.377 chance of correct selection. This is in line with the simulation results from the article of Seale and Rapoport, 1997. This deviation of 1 could be explained by the random nature of the experiment. The line of the graphs are flat at the top. This suggests that the strategy is quite insensitive to minor changes. A chance of success within 90 % of the optimal strategy (so more than approximately 33,300 correct selections) is reached in the broad interval [9,21]. The probability of success shows a declining linear trend.

For $n = 80$, the optimal strategy is to reject the first 29 applicants, and then hire the first who is better. This strategy results in 0.372 chance of correct selection (Gilbert and Mosteller, 1966). Our results for the standard SP for $n = 80$ (right graph of fig. 4.1) again resemble the theoretical predictions. In our simulation, the top is reached with cutoffpoint 31, with 0.369 chance of correct selection. The line of the graph has a similar shape to the graph previous discussed. Once more, this suggests that the strategy is insensitive to small variations. A chance of success within 90 % of the optimal strategy is reached in the interval [17,42].

4.2 Results SP with uncertainty

Intervals

In figure 4.2, uncertainty is applied to the observations in the form of intervals. Each SP round, a random number is generated between 0 and the boundary, in this case 10,000. The actual value is randomly chosen from an interval. The interval size varies from 500 to 8000. The black line depicts the original SP graph, the other lines are simulations with associated interval sizes.

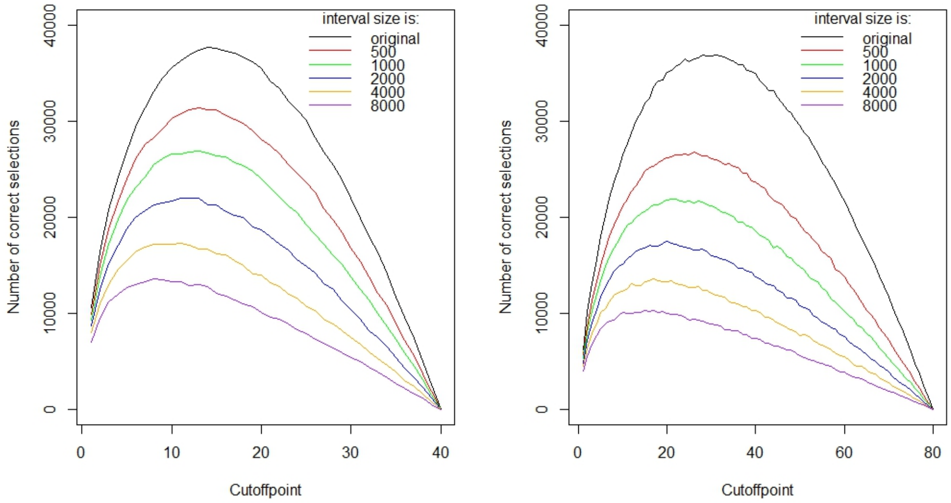

FIGURE 4.2: Graph intervals n=40 and n=80, the boundary is 10,000.

For $n = 40$, the left graph above is evaluated. Naturally, the number of correct selections decreases when the uncertainty increases. More surprisingly, the cutoffpoint seems to decline when the interval size increases, from cutoffpoint 13 with interval size 500, cutoffpoint 11 with interval size 2000 to cutoffpoint 8 with interval size 8000.

The graphs does seem to become flatter when the interval size is larger. However, this observation is not applicable if one keeps the same measures, relative success, as has been done before. As can be seen, the top of the graph with interval size 8000 is much lower, which means that the 90 % interval is smaller (10 % of 12,000 is approximately 1,000). The cutoffpoint interval of the best 90 % does not show an increase with this measure, with an interval [8,19] with 90 % effectiveness for interval size 500 to [5,14] for interval size 8000.

For $n = 80$, again the same trend is visible: the cutoffpoint decreases when the interval size gets larger. The cutoffpoint decreases from 26, to 21 to 15, with the interval size respectively 500, 2000 and 8000. This is a optimal cutoffpoint of less than 20 % of 80. Also the relative flatness of the graphs is, in line with the previous graph, does not show a clear increase or decrease. It differs from [14,28] (interval size is 500) to [8,26] (interval size is 8000). Still, the insensitivity of the strategy is clearly visible in the large intervals observed. In figure B.2 and B.3, the cutoffpoint of the top is plotted against the interval size, and the error bars represent the cutoffpoint interval with 90 % effectiveness.

Normal distribution

In figure 4.3, the observations are uncertain in the form of a normal distribution. The boundary is set at 1000, so each round a number is generated between 0 and 1000. That number is the mean of a normal distribution, with a specified standard deviation, where the value of the secretary is chosen from. The standard deviation varies from 5 to 80.

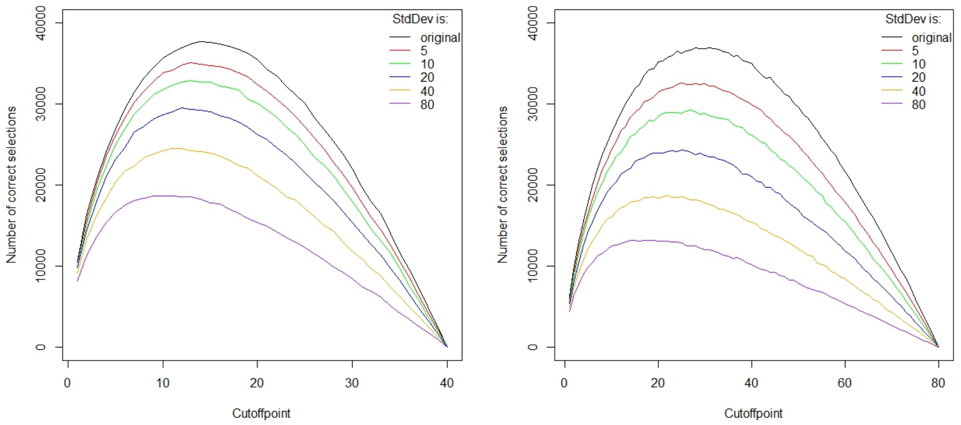

FIGURE 4.3: Graph normal distribution n=40 and n=80, the boundary is 1000.

For $n = 40$, the left graph shows the results of the influence of a standard distribution. Notice the similarities with the graphs for intervals. The same trend is shown that the cutoffpoint decreases when more uncertainty is added, in this case a larger standard deviation. The optimal cutoffpoint is 13 when the standard deviation is 5 (35.0 % chance of success), and lowers to 10 when the standard deviation is 80 (18.7% chance of success). Still, the sensitivity of the strategy is low. Whether the

DM chooses 6 or 17 as cutoffpoint, it will still achieve 90% effectiveness in comparison to the optimum when the standard deviation is 80.

For $n = 80$, the observations made are demonstrated in figure B.5. In this graph, it is clearly visible that the line declines, though it is not monotone. The line shows irregularities, at some points showing a higher cutoffpoint with more uncertainty. Even so, the decreasing trend manifests, both in the cutoffpoint of the top and in the 90 % intervals.

Pareto distribution

In figure 4.4, the results of the SP with a pareto distribution are presented. The boundary used is 10,000. The value is chosen from a pareto distribution with the parameter α varying from 80 to 10. As has been explained in chapter 2, the pareto distribution expresses more uncertainty when the parameter α is smaller.

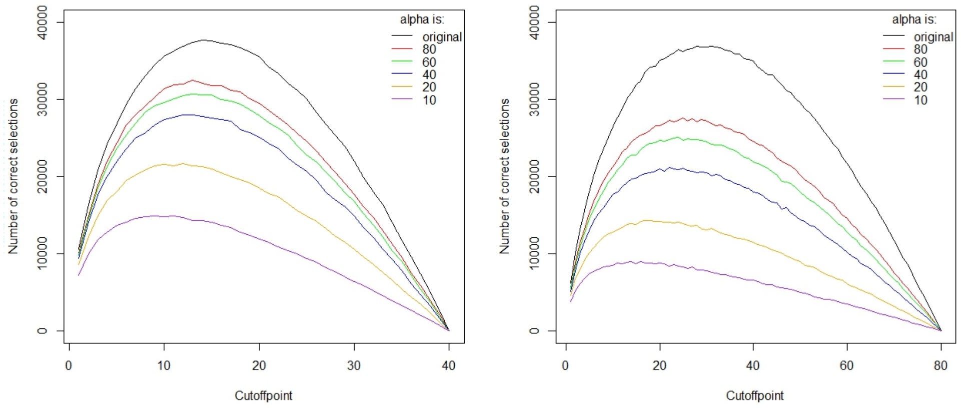

FIGURE 4.4: Graph pareto distribution n=40 and n=80, the boundary is 10,000.

For $n = 40$, the simulation results are depicted in the left graph. When α decreases, the number of selections decreases as well. Even though the top of graphs lay at cutoffpoint 13 for $\alpha = 80,60$ and 40, the shift towards the left is visible when α is lowered to 20 and 10, with optimal cutoffpoints respectively 12 and 9. Note that the number of selections for the first three variations are close, just around 2,000 apart from each other. This could give a reason for the identical optimal cutoffpoints. The 90 % intervals show the same relative insensitivity as indicated for the previous distributions.

For $n = 80$, the change of the top of the graphs is clearly evident. The optimal cutoffpoint starts at 25 ($\alpha = 80$), declines to 22 ($\alpha = 40$) and finally 14 ($\alpha = 10$). Also the pattern of rather insensitivity shows again. The size of the 90 % intervals could rise to 24 ([15,39] with $\alpha = 80$). Furthermore, the heights of the graph for $n - 40$ fall remarkably in comparison to the graph for $n = 40$, from 32481 to 27602. These results are visually depicted in appendix B.6 and B.7.

4.3 Analysis results

Selection chance

The selection chance has shown exact same patterns in all the versions, demonstrated in figure B.1 in the appendix. The apparent problem is not different from the original problem. Only the chance of success is different with the uncertainty component. The selection chance (whether a secretary is chosen), should begin with $\frac{n-1}{n}$ and end with $\frac{1}{n}$, because $\frac{n-1}{n}$ is the chance that the best one will not be the first, and $\dfrac{1}{n}$ is the chance the best one is the last one, which corresponds with the selection chance.

Shift cutoffpoint

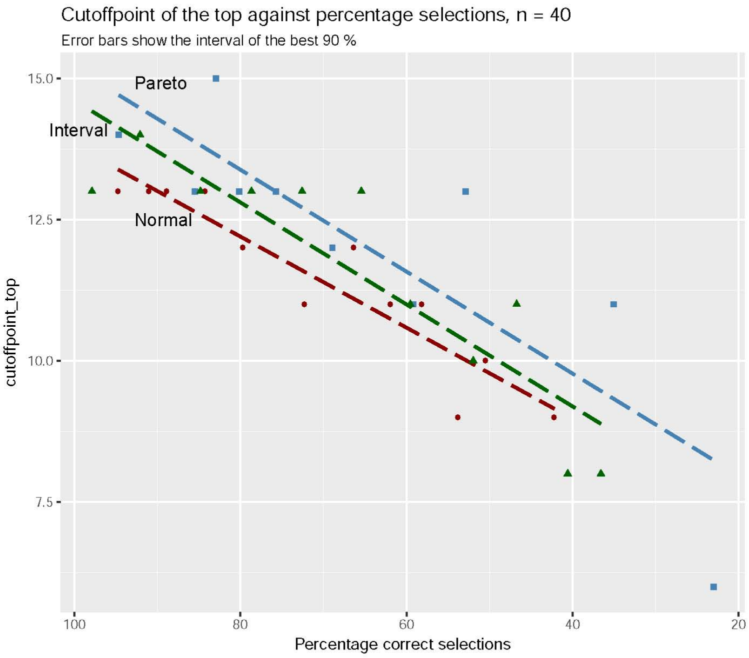

FIGURE 4.5: Multiple distributions, n = 40, cutoffpoint top against percentage selections.

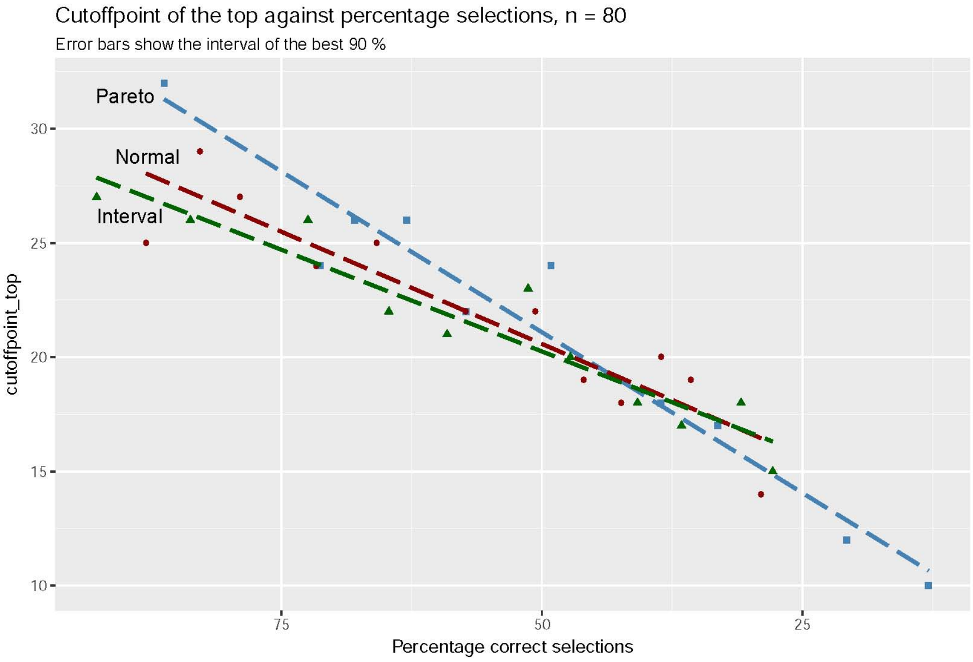

FIGURE 4.6: Multiple distributions, n = 80, cutoffpoint top against percentage selections.

In figure 4.5 and 4.6 where the cutoffpoint of the top is plotted against the amount of uncertainty as the percentage selections, the results of all the distribution are combined. Why is it effective to have a smaller cutoffpoint, when there is more uncertainty? The apparent selection process stays the same. The DM chooses as if the apparent values are the real values, but now the actual values determine the outcome of the choice of DM. If the intervals are disjoint, the DM could maintain the same strategy. However, if the intervals overlap, a difference might occur between the apparent and the actual value. We will illustrate the influence of this mechanism with a small-scale example.

Suppose you have the apparent values *10, 20, 30, 40* and *50*. In the original problem, *50* would also be the actual best one. The optimal cutoffpoint would be 2, with a chance of success of 0.43 (Gilbert and Mosteller, 1966). Then the chance of choosing 50 is the highest. The versions with uncertainty introduce a higher chance that the actual best value is not *50*, but for example *40*. The larger the uncertainty, the larger the chance that *50* is not the best, but *40*, or even *30* or *20* is the actual best. This is the case for all the sorts of distributions analysed.To evaluate when a strategy with a smaller cutoffpoint achieves more than a larger cutoffpoint, we will discuss in what cases the actual best is not chosen.

The first reason for not selecting the actual best, is when the best is at the start. The cutoffpoint is always 1 or higher, so the first one will never be selected. The second reason is that there has been an apparent higher value before the actual best, for example, *50* is seen before *40*. Even when *40* was actually better, it is then never chosen with a cutoff strategy.

We apply this to our example. When *20* is the best, the only permutations where *20* is chosen, is when *10* is the first applicant, and then *20*. After 20, the rest may follow in different orders, but it could not be the case that *30* is before *20*, because it will never result in choosing *20*. That means that the cutoffpoint in this case should be 1, because after *10*, the first one, *20* should be chosen. The same pattern can be seen when *30* is the actual best. The only values before it can be *10* and *20*, regardless of the order of those two. The cutoffpoint, again, should be low.

Permutation:	Cutoffpoint:
(10, 30, ..., ..., ...)	1
(20, 30, ..., ..., ...)	1
(20, 10, 30, ..., ...)	1 or 2
(10, 20, 30, ..., ...)	2

The cutoffpoint can not be 3 or higher, because then one of the two apparent larger numbers will definitely be seen before *30*.

This pattern will be seen in larger examples as well. The larger the cutoffpoint, the larger the chance that an apparent larger value is seen before an apparent smaller value that is actually better. That is why the larger the uncertainty, so the larger the chance that not *50* but an apparent lower number is the actual best, the more effective it is to have a lower cutoffpoint.

To illustrate this claim, we generated a version of the SP where there is the most uncertainty possible. Two random values are generated, one the apparent value, the other the actual value. As can be seen in figure B.8, the optimal cutoffpoint is very low, around 1 or 2. Because the selection process is random (the DM has no information about who is best), the theoretical chance of correct selection is 1 in n, in this case $\frac{1}{80} \approx 0.013$. This corresponds with the simulated probability of correct selections (number of selections divided by the selections made), which is 0.12 or 0.13 with the different cutoff strategies. Strategies with lower cutoffpoints result in a higher selection chances, which then result in a higher probability of correct selection.

Chapter 5

Conclusion

Choosing under uncertain circumstances has its price. To minimize the cost, we have been looking for the best cutoff strategy. In the introduction, we have asked ourselves what effect the amount of uncertainty in the SP problem. Now, we can formulate an answer. Furthermore, we elaborate on the implications of the results for the AI community, possibly even for our daily life, discuss the limitations, and explore suggestions for further research.

We have evaluated the optimal cutoffpoint in the SP with uncertain observations. In the classic SP, the optimal strategy has a cutoffpoint around 37% of the applicants. We have observed in chapter 4 that the application of intervals, normal distribution and pareto distribution has reduced the cutoffpoint to 25 % or even 20 % for both $n = 40$ and $n = 80$ when the amount of uncertainty was increased, in the form of interval size, standard deviation or α. The probability of selection appeared to be similar for all versions, including the classic SP, and decreased linearly when the cutoffpoint increased.

In the field of artificial intelligence, a challenge lies in finding a model that unites mathematical theory with the real world. Humans are not rational agents, and will possibly not always follow the mathematical optimal strategy. But intelligent systems could make more effective choices in the everyday world when following the optimal strategy for a mathematical model that resembles the real world. For problems resembling the SP, the optimal strategy might involve making a faster decision when the uncertainty gets larger.

In the introduction we mentioned our intuition that more uncertainty would require a longer exploration phase, in other words that the cutoffpoint would increase when the uncertainty increased, This intuition is countered. This could suggest that choosing sooner would be a good idea. If there is a chance that the real estate agent is lying about the house, the advice may be to not wait too long, and just pick a house. If it is hard to evaluate which relationship was better than the other, do not wait too long for the perfect partner to appear, because recognizing him or her as such might be difficult. For the SP, this phenomenon has been illustrated in section 4.3.

However, the application of the results and the result itself should be put into perspective. We tried to minimize the influence of the choices made, such as the number of applicants, the number of simulation rounds and the boundary used, as explained in chapter 3. This does not take that influence away, and should be taken into account when interpreting the results. The random nature of the simulations have caused us to be more reserved in stating the conclusion. Furthermore, the bridge between this mathematical problem and real life decision making could not been made by solely adding uncertainty to the observations, but it could be considered as a step in the good direction. When combined with other researched variations, it might come even closer.

A further research could investigate a combination of a myriad of variations, to apply the SP to real world situations even more. The way of modelling uncertainty could be varied further. Now, each strategy was only tested with a constant amount of uncertainty involving one particular distribution. If the uncertainty would be varied within one round, while sharing that information with the DM, the decision could be influenced by the amount of uncertainty at that point. Moreover, the uncertainty could influence other factors in the problem. What if the amount of applicants is unknown (Freeman, 1983)? What if the pay-off is utility based (Lindley, 1961)? The found solutions for these variations could be used to direct the search for the optimal strategy. Would the cutoffpoint still show a decreasing trend?

We evaluated the results, discussed the limitations, provided a view on the applications of this study, and touched upon possible future research. You may take from it as a reserved advice: if you are not sure, choose sooner!

Bibliography

DeGroot, Morris H (1968). “Some problems of optimal stopping”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 30.1, pp. 108–122.

Ferguson, Thomas S. (1989). “Who solved the secretary problem?” In: *Statistical science* 4.3, pp. 282–289.

Freeman, PR (1983). “The secretary problem and its extensions: A review”. In: *International Statistical Review* 51.2, pp. 189–206.

Gardner, Martin (1966). “New mathematical diversions”. In: *Scientific American*, pp. 35–36.

Gianini, Jacqueline and Stephen M Samuels (1976). “The infinite secretary problem”. In: *The Annals of Probability* 4.3, pp. 418–432.

Gilbert, John P. and Frederick Mosteller (1966). “Recognizing the Maximum of a Sequence”. In: *Journal of the American Statistical Association* 61.313, pp. 35–73. ISSN: 01621459.

Gnedin, Alexander V (1994). “A solution to the game of googol”. In: *The Annals of Probability*, pp. 1588–1595.

Levy, Moshe and Haim Levy (2003). “Investment talent and the Pareto wealth distribution: Theoretical and experimental analysis”. In: *Review of Economics and Statistics* 85.3, pp. 709–725.

Lindley, Denis V (1961). “Dynamic programming and decision theory”. In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 10.1, pp. 39–51.

Lyon, Aidan (2013). “Why are normal distributions normal?” In: *The British Journal for the Philosophy of Science* 65.3, pp. 621–649.

Russel, Betrand (1948). *Human Knowledge: Its Scope and Limits*. New York: Simon and Schuster, p. 527.

Seale, Darryl A and Amnon Rapoport (1997). “Sequential decision making with relative ranks: An experimental investigation of the" secretary problem"”. In: *Organizational behavior and human decision processes* 69.3, pp. 221–236.

Seale, Darryl A and Amnon Rapoport (2000). “Optimal stopping behavior with relative ranks: The secretary problem with unknown population size”. In: *Journal of Behavioral Decision Making* 13.4, pp. 391–411.

Vanderbei, Robert J (1983). *The postdoc variant of the secretary problem*. Tech. rep. Technical report, Princeton University.

Pictures

Figure 4.3: "Normal distribution" by mr. W. Toews (2007), available at https://en.wikipedia.org/wiki/File:Standard_deviation_diagram.svg under CC BY 2.5. Full terms at https://creativecommons.org/licenses/by/2.5/.

Figure 4.4: "Pareto distribution" by Danvildanvil (2014), available at https://en.wikipedia.org/wiki/File:Probability_density_function_of_Pareto_distribution.svg under CC BY-SA 3.0. Full terms at https://creativecommons.org/licenses/by-sa/3.0/.

Appendix A

Justification design choices

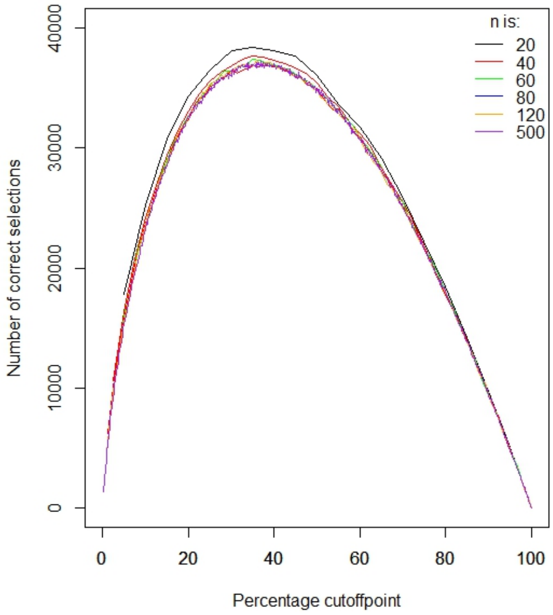

FIGURE A.1: Graph original SP, n = 20, 40, 60, 80, ,120, 500. The x-axis shows the ratio of the cutoffpoint.

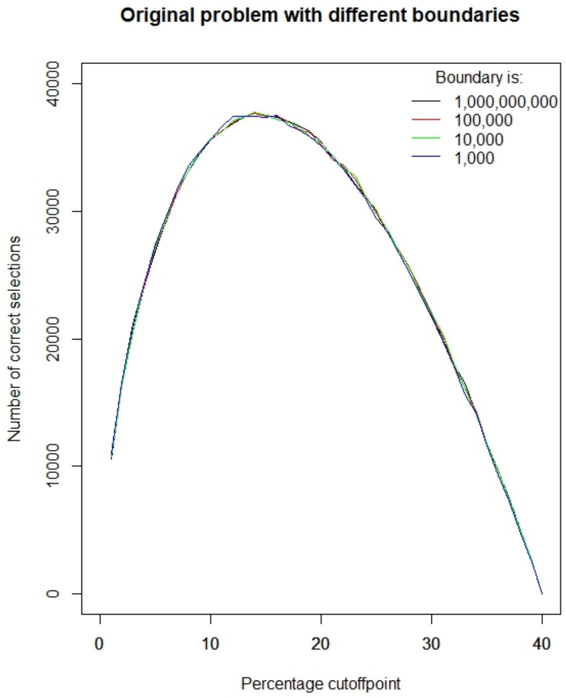

FIGURE A.2: Original SP with different boundaries.

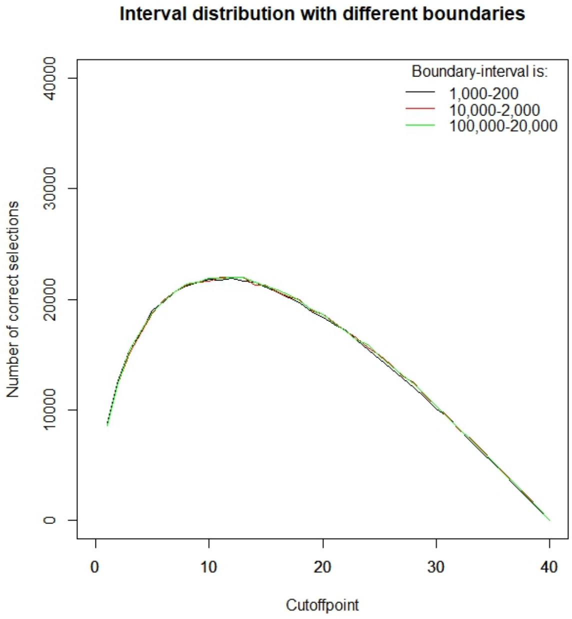

FIGURE A.3: Interval distribution with boundaries 1,000-200, 10,000-2,000, 100,000-20,000.

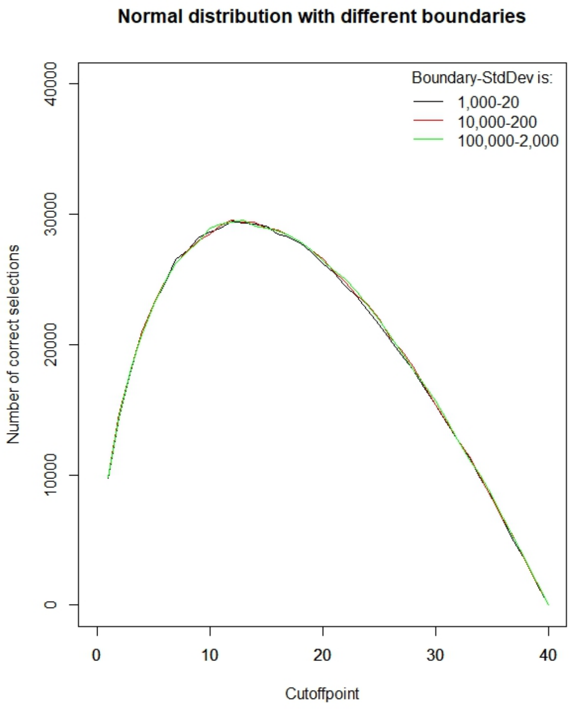

FIGURE A.4: Normal distribution with boundaries 1,000-20, 10,000-200, 100,000-2,000.

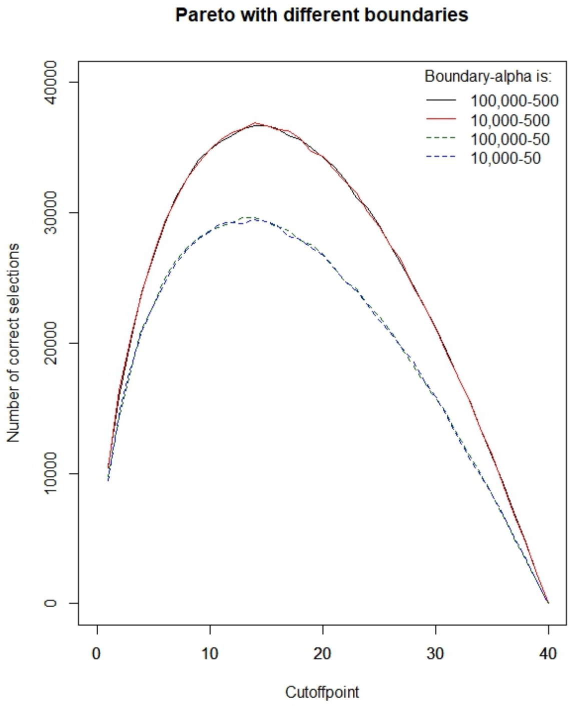

FIGURE A.5: Pareto distribution with different boundaries and α, 100,000-500, 10,000-500, 100,000-50, 10,000-50

Appendix B

Additional graphs

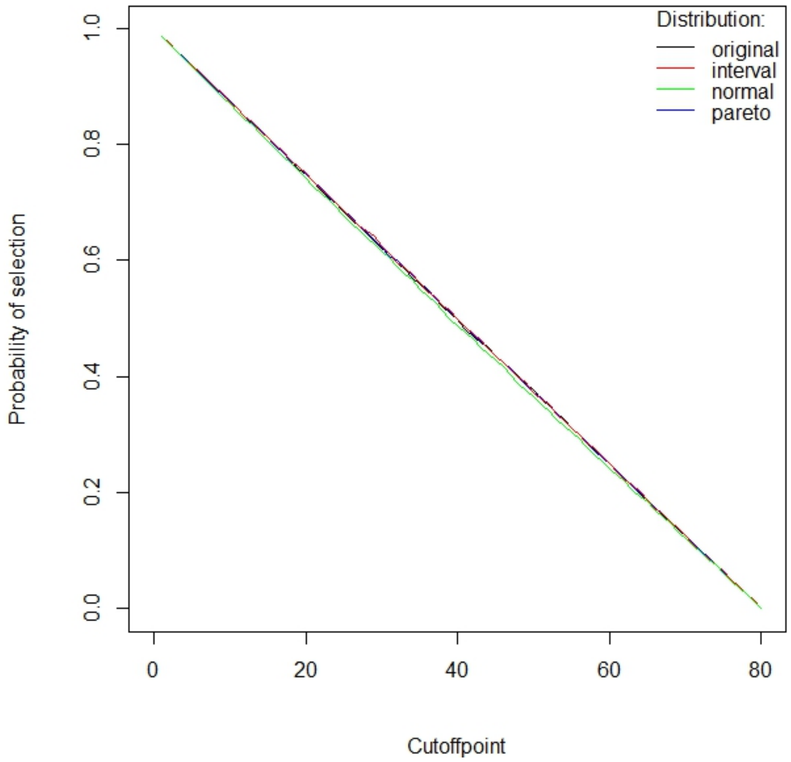

FIGURE B.1: Graph probability of selection, for the original SP, intervals, normal distribution and pareto distribution.

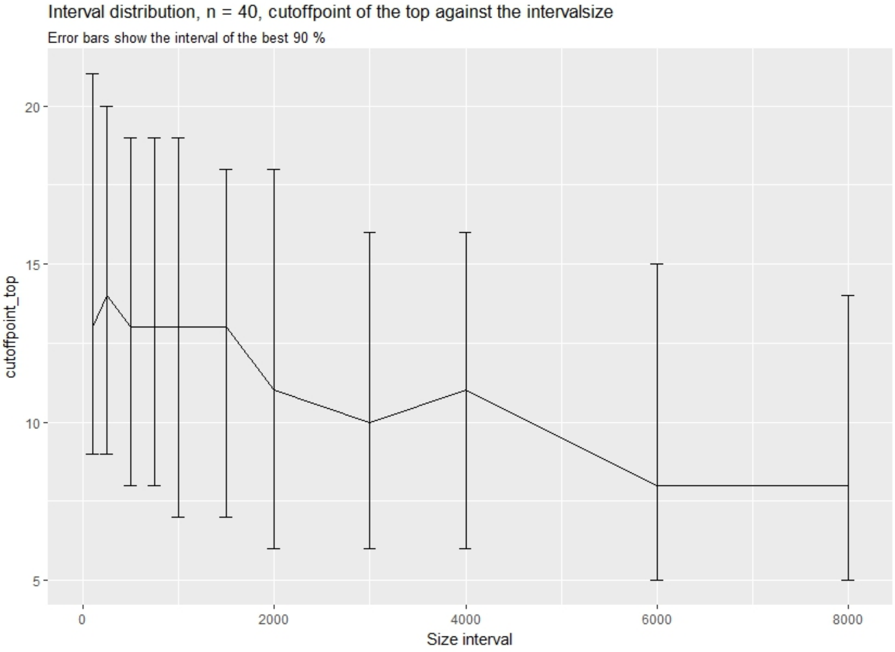

FIGURE B.2: Graph interval distribution, n = 40, cutoffpoint top against interval size.

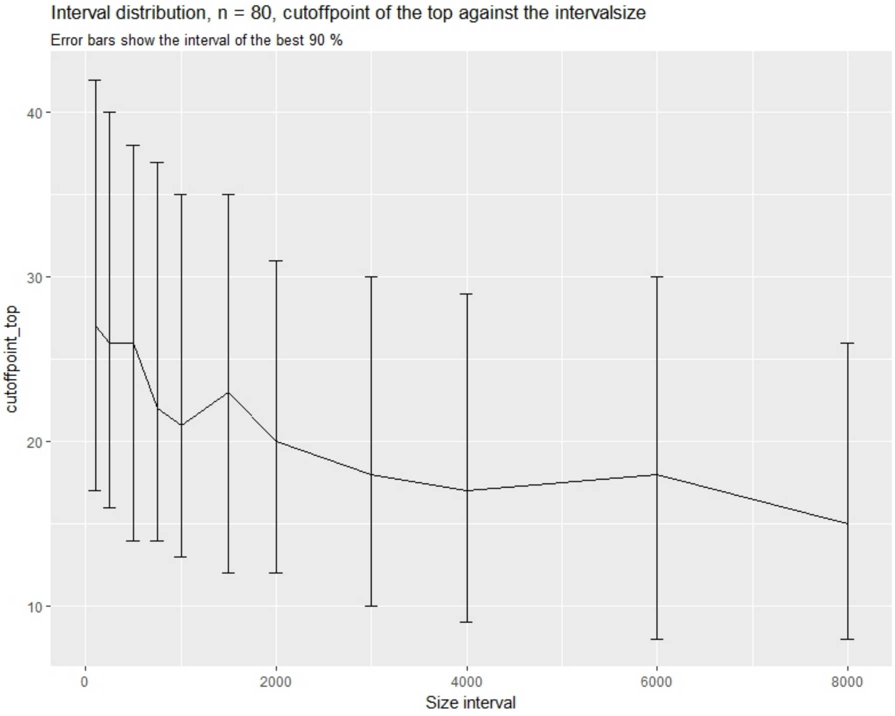

FIGURE B.3: Graph interval distribution, n = 80, cutoffpoint top against interval size.

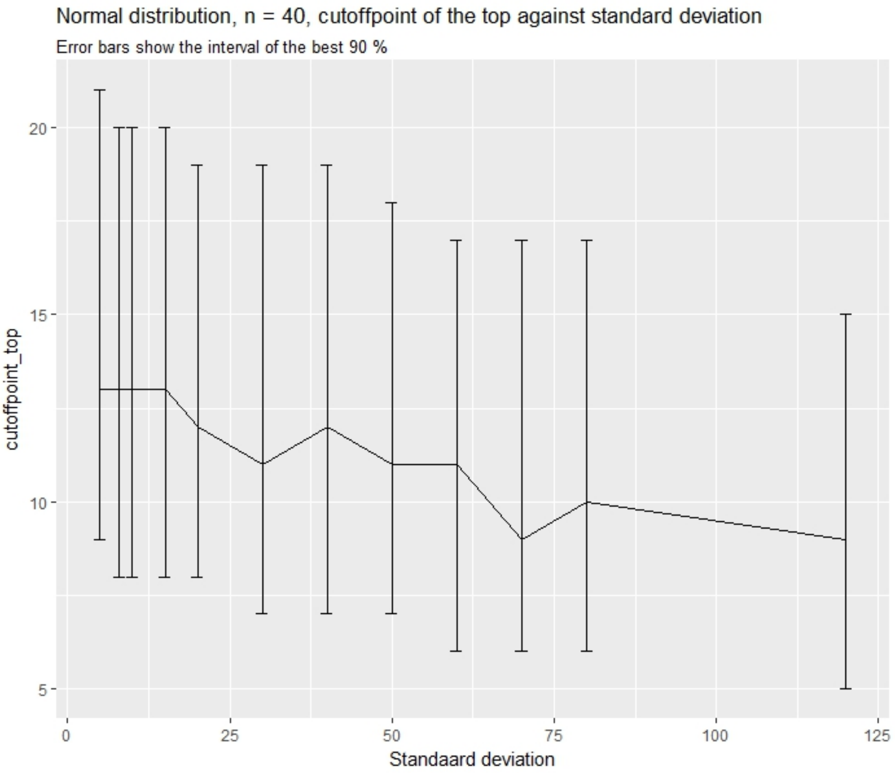

FIGURE B.4: Graph normal distribution, n = 40, cutoffpoint top against standard deviation.

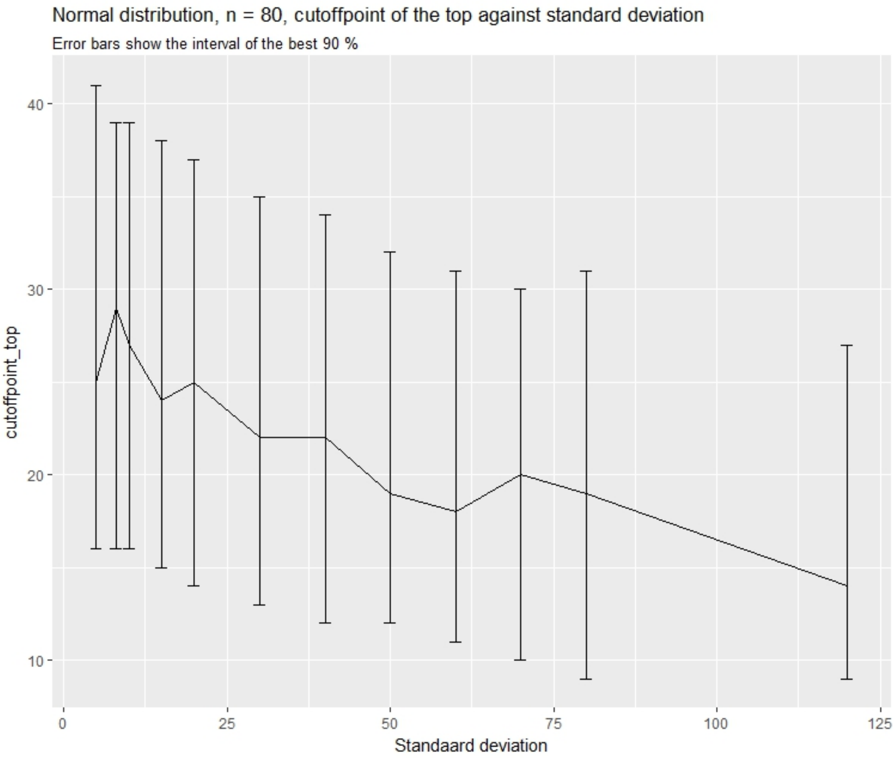

FIGURE B.5: Graph normal distribution, n = 80, cutoffpoint top against standard deviation.

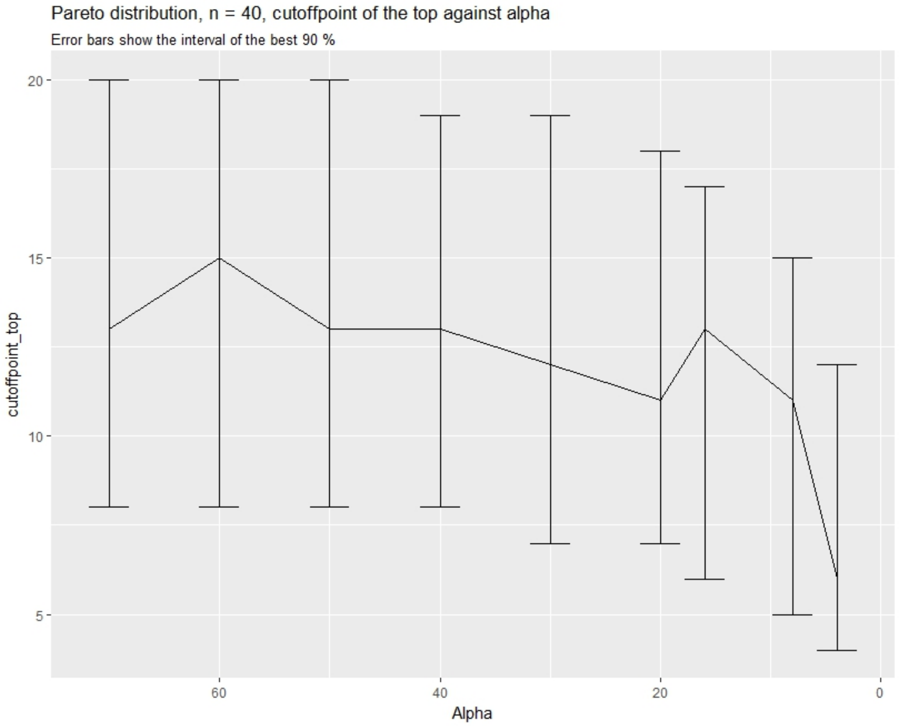

FIGURE B.6: Graph pareto distribution, n = 40, cutoffpoint top against alpha.

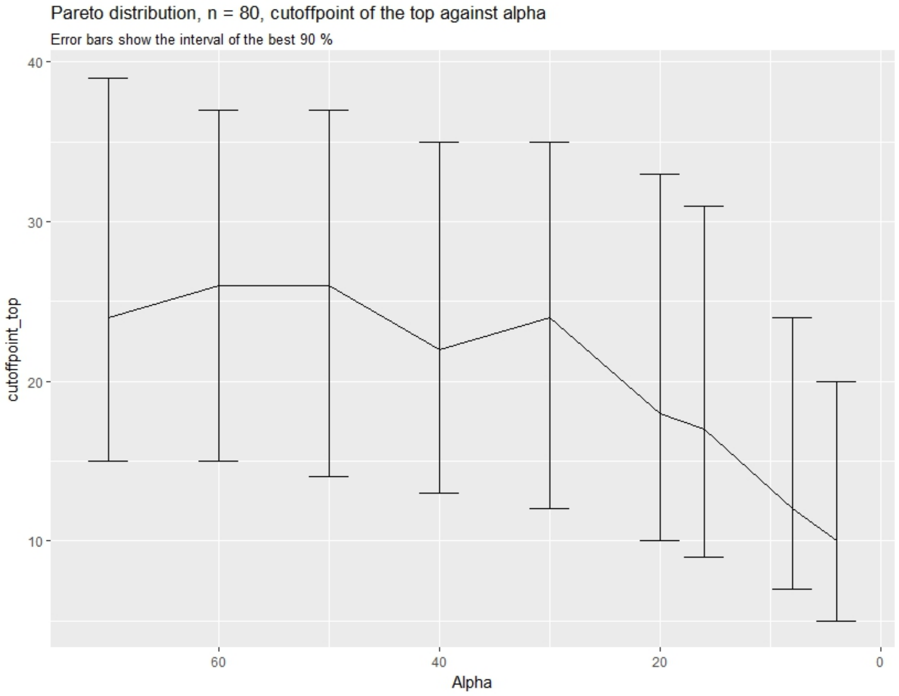

FIGURE B.7: Graph pareto distribution, n = 80, cutoffpoint top against alpha.

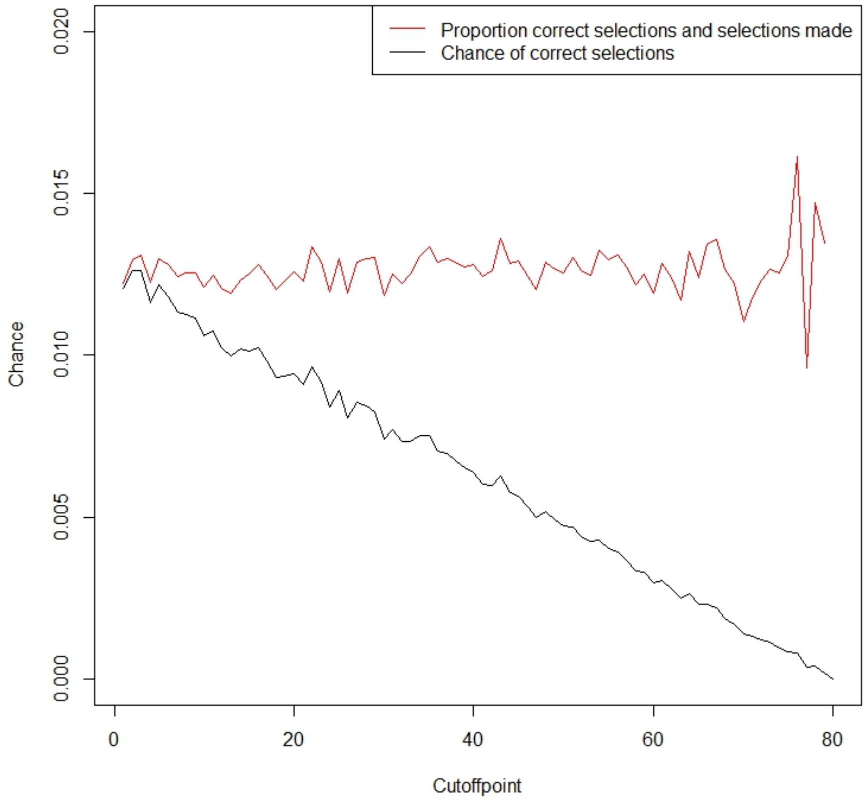

FIGURE B.8: Two random values generated as apparent and actual value.

Appendix C

C# implementation

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO;
using Accord.Statistics.Distributions.Univariate;


namespace SPsimulations
{
    public partial class Program
    {
        // parameters
        static string version;// original, normal distribution,
        // pareto distribution or fixed intervals
        static int n;
        static int rounds;
        static int boundary;
        static int param; // value of parameter, such as alpha.

        static Observation observation;
        static decisionmaker DM;
        static int[] waarden = new int[1];

        static string best; // Used for intervals:

        // dependent variables
        static int wins;
        static int losses;
        static int nochoose;

        // other global variables
        static bool print = false; // print intermediate results on console
        static bool txt = true; // Output to txt file

        static StreamWriter writer;
        static TextWriter oldOut;
        static Random r;
        static StreamReader sr;
        static void Main(string[] args)
        {
            sr = File.OpenText($"./readfiles/input.txt");
            bool endoffile = false;
            while (!endoffile)
            {
                endoffile = ReadInput();
                DoSimulation();
            }
            Console.WriteLine("Done!");
            Console.ReadLine();
        }

        // Multiple rounds of one specific (input given) version of SP.
        private static void DoSimulation()
        {
            int nrofparameters = 1;
            for (int k = 0; k < nrofparameters; k++)
            {
                // initialize the values for the observation
                if(version == "original")
                {
                    observation = new Observation(version, boundary);
                }
                else
                {
                    param = waarden[k];
                    nrofparameters = waarden.Length;
                    observation = new Observation(version, param, boundary);
                }

                if (txt)
                    OutputToTxt();

                r = new Random();
                // title of the txtfile:
                Console.WriteLine($"Cutoffpoint,␣selections,␣correct");

                // variation of cutoffpoint
                for (int j = 1; j <= n; j++)
                {
                    // reset values
                    wins = 0;
                    losses = 0;
                    DM = new decisionmaker(j, best, version);

                    for (int i = 0; i < rounds; i++)
                    {
                        solveSecretaryProblem();

                    }
                    Console.WriteLine($"{j},{(wins␣+␣losses)},{wins}");
                }

                if (txt)
                    Closetxt();
            }
        }
        private static void Closetxt()
        {
            Console.SetOut(oldOut);
            Console.WriteLine("Simulation␣round␣finished.");
            writer.Close();
        }


        // Name the txtfile according to the version of SP
        private static void OutputToTxt()
        {
            FileStream outputstream;
            oldOut = Console.Out;
            string path;
            switch (version)
            {
                case "normal␣distribution":
                    path =
                    $"./txtfiles/{n}_{rounds}_{version}_stddev-{param}_b-{boundary}.txt";
                    break;
                case "fixed␣intervals":
                    path =
                    $"./txtfiles/{n}_{rounds}_{version}_ivs-{param}_b-{boundary}.txt";
                    break;
                case "pareto␣distribution":
                    path =
                    $"./txtfiles/{n}_{rounds}_{version}_alpha-{param}_b-{boundary}.txt";
                    break;
                default: // original
                    path = $"./txtfiles/{n}_{rounds}_{version}_b-{boundary}.txt";
                    break;
            }
            try
            {
                outputstream =
                new FileStream(path, FileMode.CreateNew, FileAccess.Write);
                writer = new StreamWriter(outputstream);
                Console.SetOut(writer);
            }
            catch
            {
                // When file exists already, the results are printed to the console.
                Console.WriteLine("Caution,␣txt␣file␣exists␣already.␣Path␣is:" + path);
            }
        }

        private static void solveSecretaryProblem()
        {
            Observation max = new Observation(version);

            double currentSecretary = 0;
            DM.reset();

            // n secretaries are presented one by one.
            for (int i = 1; i <= n; i++)
            {
                // the observation is generated according to the version.
                observation.generateObservationInfo(r, i);
                currentSecretary = observation.Value;

                if (print)
                    Console.WriteLine(currentSecretary);
                // The actual best secretary is saved.
                if (currentSecretary > max.Value)
                {
                    max.Value = currentSecretary;
                    max.Number = i;
                }

                // As long as the DM has not chosen, secretaries will be presented.
                if (!DM.Chosen)
                {
                    DM.strategy(observation);
                }
            }

            // The results are gathered and printed.
            //No choose:
            if (!DM.Chosen)
            {
                if (print)
                    Console.WriteLine("No␣secretary␣was␣chosen.␣Lose.");
                nochoose++;
            }

            //Loss, maximum is compared to secretary hired by the DM:
            else if (max.Number != DM.hiredSP.Number)
                {
                    if (print)
                        Console.WriteLine("The␣best␣secretary␣was␣number␣" +
                        max.Number + "␣with␣value␣" + max.Value +
                        ".\nDM␣chose␣number␣" + DM.hiredSP.Number +
                        "with␣value␣" + DM.hiredSP.Value + ".");
                    losses++;
                }
            else // Win:
                {
                    if (print)
                        Console.WriteLine("Congrats,␣you've␣picked␣the␣best␣one,␣number␣"
                        + DM.hiredSP.Number + "␣with␣value␣" + DM.hiredSP.Value);
                    wins++;
                }
            }
        }
    }
```

```
using System;
using System.Windows.Forms.DataVisualization.Charting;
using Accord.Statistics.Distributions.Univariate;


namespace SPsimulations
{
    internal class Observation
    {
        private string version;
        private int boundary;
        private int intervalsize;
        private int stdDev;
        private double alpha;

        private double value; //actual
        private int number;
        public int Lowerborder; //apparent:
        public int Mean;

        public Observation(string v) // for the maximum
        {
            version = v;
        }

        public Observation(string v, int b) // original
        {
            version = v;
            boundary = b;
        }

        public Observation(string v, int param, int b)
        {
            version = v;
            boundary = b;
            switch(version)
            {
                case "fixed␣intervals":
                    intervalsize = param;
                    break;
                case "normal␣distribution":
                    stdDev = param;
                    break;
                default: // version is "pareto distribution"
                    alpha = param;
                    break;
            }
        }
        // the apparent and actual value of the secretary is generated.
        public void generateObservationInfo(Random r, int i)
        {
            number = i;

            switch (version)
            {
                case "original":
                    value = r.Next(boundary);
                    break;
                case "fixed␣intervals":
                    Lowerborder = r.Next(boundary);
                    value = r.Next(Lowerborder, Lowerborder + intervalsize);
                    break;
                case "normal␣distribution":
                    Mean = r.Next(boundary);
                    NormalDistribution nd = new NormalDistribution(Mean, stdDev);
                    value = nd.Generate(r);
                    break;
                default: // "pareto distribution":
                    Lowerborder = r.Next(1, boundary);
                    ParetoDistribution pd = new ParetoDistribution(Lowerborder, alpha);
                    value = pd.Generate(r);
                    Mean = (int)pd.Mean;
                    break;
            }
        }

        public Observation(Observation o) // copy observation
        {
            version = o.version;
            value = o.value;
            intervalsize = o.intervalsize;
            Lowerborder = o.Lowerborder;
            number = o.number;
            stdDev = o.stdDev;
            Mean = o.Mean;
        }

        public double Value
        {
            get
            {
                return value;
            }
            set
            {
                this.value = value;
            }
        }
        public int Upperborder
        {
            get
            {
                return Lowerborder + intervalsize;
            }
        }

        public string Version
        {
            get
            {
                return version;
            }
        }

        public override string ToString()
        {
            return "Secretary␣" + number + "␣with␣value␣" + value;
        }

        public int Number
        {
            get
            {
                return number;
            }
            set
            {
                this.number = value;
            }
        }
    }
}
```

```
using System;

namespace SPsimulations
{
    internal class decisionmaker
    {
        private string best;
        private string version;

        private int cutoff;
        private bool chosen;
        private int seen; // the number of seen applicants
        public Observation hiredSP;
        private Observation max;
        Func<Observation, bool> bestSoFar;

        public decisionmaker(int coff, string b, string v)
        {
            cutoff = coff;
            best = b;
            version = v;
            hiredSP = new Observation(version);
            max = new Observation(version);
            defineBest();
        }

        //according to the version, the function of who is best is defined.
        private void defineBest()
        {
            switch (version)
            {
                case "pareto␣distribution":
                    bestSoFar = observation => observation.Lowerborder > max.Lowerborder;
                    break;
                case "normal␣distribution":
                    bestSoFar = observation => observation.Mean > max.Mean;
                    break;
                case "fixed␣intervals":
                        switch (best)
                        {
                            case "optimistic":
                                bestSoFar = observation
                                => observation.Upperborder > max.Lowerborder;
                                break;
                            case "pessimistic":
                                bestSoFar = observation
                                => observation.Lowerborder > max.Upperborder;
                                break;
                            default: // lower border
                                bestSoFar = observation
                                => observation.Lowerborder > max.Lowerborder;
                                break;
                        }
                    break;
                default: // original/
                    bestSoFar = observation => observation.Value > max.Value;
                    break; } }
        public bool Chosen
        {
            get
            {
                return chosen;
            }
        }

        public void strategy(Observation observation)
        {
            seen++;
            if (bestSoFar(observation))
            {
                max = new Observation(observation);
                if(seen > cutoff) // current secretary is chosen.
                {
                    hiredSP = new Observation(observation);
                    chosen = true;
                }
            }
        }

        public void reset()
        {
            chosen = false;
            hiredSP = new Observation(version);
            seen = 0;
            max = new Observation(version);
        }
    }
}
```

```
using System;
using System.IO;

namespace SPsimulations
{
    public partial class Program
    {
        // This reads the input from a textfile in the format of the following example:
   /*      n=40
           rounds=100000
           version=normal distribution
           best=mean
           boundary=10000
           waarden parameter=5 10 20 40 80
  */
        public static bool ReadInput()
        {
            string[] values;
            string ln;

            while ((ln = sr.ReadLine()) != null)
            {
                try
                {
                    values = ln.Split('=');
                    //separated by empty line or "[]" for multiple inputs in one file:
                    if (!(ln.StartsWith("[") || ln.Length == 0))
                        assignvalue(values[0], values[1]);
                    else
                        return false;
                }
                catch
                {
                    throw new Exception("Input␣is␣not␣in␣right␣format.␣Please␣adapt␣it.");
                }

            }
            return true;
        }
        public static void assignvalue(string param, string val)
        {
            string[] values; int nr;
            switch (param) // parameters are assigned.
            {
                case "n":
                    n = int.Parse(val);
                    break;
                case "rounds":
                    rounds = int.Parse(val); ;
                    break;
                case "version":
                    version = val;
                    break;
                case "best":
                    best = val;
                    break;
                case "boundary":
                    boundary = int.Parse(val);
                    break;
                default: //"waarden parameter":
                    nr = val.Split().Length;
                    values = new string[nr];
                    waarden = new int[nr];
                    values = val.Split();
                    for (int i = 0; i < nr; i++)
                        waarden[i] = int.Parse(values[i]);
                    break;
            }
        }
    }
}
```