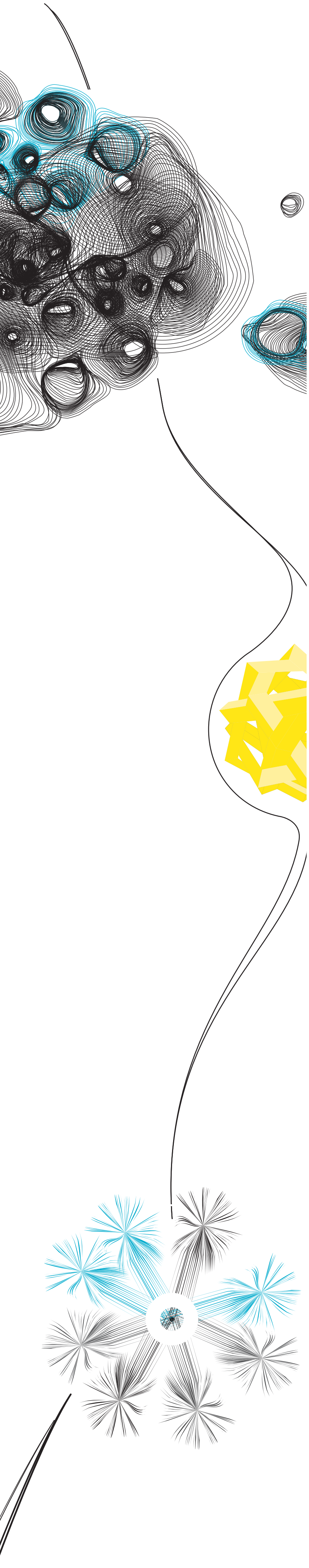


Thesis Applied Mathematics
Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS)

Optimization of the BUbiNG web crawler

Anne Buijsrogge

Assessment committee:
Prof. Dr. R.J. Boucherie
Dr. N. Litvak
Dr. E.A. van Doorn

September 16, 2014

UNIVERSITY OF TWENTE.

Abstract

This research is about the performance and optimization of the BUbiNG web crawler, an open-source web crawler developed by Paulo Boldi et al [2]. This web crawler aims to have high throughput in terms of crawled pages per unit time. The data structures of the web crawler that we consider are the sieve, the workbench and the workbench virtualizer. The goal is to have as many hosts as possible in the workbench, which is the crucial datum in order to have high throughput of the web crawler. In order to improve the number of hosts in the workbench, the workbench virtualizer can be used for URLs that are already extracted from the sieve. In case the workbench virtualizer is not used, we derived analytical results when assuming that the host sizes are homogeneous. When the host sizes are heterogeneous we find that several hosts dominate the workbench, resulting in a low throughput of the web crawler. To overcome this problem, the workbench virtualizer is used. We found that no natural decision policies for the workbench virtualizer improve the throughput of the web crawler. Instead, using the workbench virtualizer for the hosts that dominate the workbench results in a decision policy that overcomes these hosts from dominating the workbench.

Contents

Chapter 1

Introduction

A web crawler is a system that systematically downloads a large number of web pages from the world wide web. Web crawlers start with a list of URLs to visit, these URLs are called seeds. When the web crawler visits a given seed, all the hyperlinks on the page are identified and are added to the list of URLs to visit.

1.1 The BUbiNG Web Crawler

The BUbiNG web crawler is an open-source web crawler, developed by Paulo Boldi et al. [2]. This web crawler aims to guarantee high throughput, to overcome the limits of single-machine tools and at the same time to scale linearly with the amount of resources available, i.e., the throughput can be scaled linearly just by adding resources.

In the development of a web crawler, politeness limits should be taken into account. In [1] Paulo Boldi et al. describe politeness as "*A parallel crawler should never try to fetch more than one page at a time from a given host. Moreover, a suitable delay should be introduced between two subsequent requests to the same host.*". Fetching more than one page at a time from a given host overloads the server of the host. Due to this politeness limits it is impossible to crawl more than one URL from the same host at the same time. Therefore the throughput of the web crawler is maximized when there is the possibility to crawl from as many hosts as possible. Moreover, according to the designers of the BUbiNG web crawler, "*because of politeness, the number of distinct hosts currently being visited is the crucial datum that establishes how fast or slow the crawl is going to be*", see [2].

In the BUbiNG web crawler, a decision process takes place. The parts of the crawler that play a role in the decision process are described shortly.

- The *sieve* is the data structure where URLs to be crawled are kept.
- The *workbench* is the data structure that represents the URLs already got from the sieve.
- The *workbench virtualizer* is a second data structure which is a sequence of (virtual) queues. It contains the URLs already extracted from the sieve, but they are not yet put in the workbench.
- The *distributor* is the thread that has to make a decision. The distributor orchestrates the movement of URLs out of the sieve, either to the workbench or to the workbench virtualizer, and loads as necessary URLs from the workbench virtualizer into the workbench.

A schematic overview of the relation between the sieve, workbench and workbench virtualizer is given in Figure 1.1.

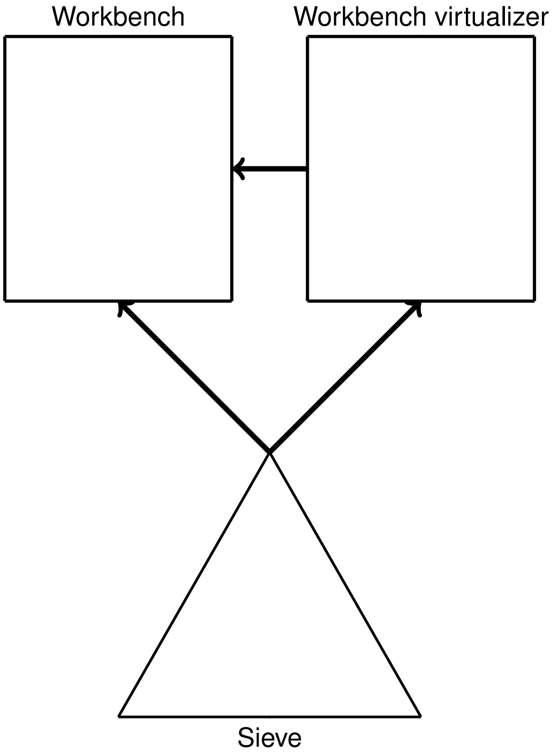


Figure 1.1: *A schematic overview of the parts of the web crawler that play a role in the decision process.*

The distributor fills the workbench with URLs, coming either from the sieve or from the workbench virtualizer. The distributor's goal is to assign the URLs to the workbench, by deciding whether to accept a URL from the sieve or from the workbench virtualizer, in such a way that the throughput is maximized. The throughput is maximized by maximizing the number of hosts the URLs belong to, considering the URLs that are in the workbench. Notice that if the distributer decides not to accept a URL from the sieve, this URL goes to the workbench virtualizer. The decisions that the distributor has to make can be translated to a one-player game with a set of rules. The player has to make the decisions and the goal of the game is to make the decisions in such a way that an objective is maximized. The objective is to maximize the number of hosts the URLs belong to in the workbench. For more details about the game the reader is referred to Chapter 2.

1.2 Research Question and Overview of the Thesis

The BUbiNG web crawler is a recently developed web crawler and the decision process of this web crawler has not been studied before. Moreover, the designers of the web crawler manage to get the throughput high in the beginning of the crawling process, but after a while the throughput starts to decay. The goal of this thesis is to make an improvement for the throughput of the BUbiNG web crawler, more specifically:

Can we optimize the decision process in the BUbiNG web crawler to maximize the number of hosts the URLs belong to in the workbench?

We start with the modeling of the game without the workbench virtualizer. Next we use it as a reference point to analyze strategies for the use of the workbench virtualizer. When there is no workbench virtualizer, no decisions have to be made and part of the problem studied is related to the birthday problem. The birthday problem is a well-known problem and it occurs frequently in literature after it has been proposed by von Mises in 1939. His question was: "*How many people must be in a room before the probability that some share a birthday, ignoring the year and ignoring leap days, becomes at least 50* *percent?*" Assuming that every day is equally likely for a birthday and excluding leap years, it is well known that the answer to this question is 23 people. The exact relation to the birthday problem is explained in Chapter 3.

This thesis is structured as follows. In Section 1.3 the related research is described. Secondly, the game is described more explicitly in Chapter 2. In Chapters 3 and 4 the game is modeled and analyzed without the use of the workbench virtualizer. Chapter 5 shows the numerical results in case the workbench virtualizer is not used. Data analysis can be found in Chapter 6. Then, in Chapters 7 and 8 strategies are implemented for the use of the workbench virtualizer. Finally, the conclusions are formulated and further research opportunities are discussed in Chapter 9.

1.3 Related work

This is the first research about the optimization of the decision process in the BUbiNG web crawler. To the best of our knowledge, no similar problem exists in literature. We describe two related problems that are used in our analysis.

To analyze the web crawler without the workbench virtualizer, we use a closed queueing network. Since the number of URLs in the workbench remains the same, the stochastic process that describes the dynamics to the workbench is closely related to closed queueing systems. In this closed queueing system the jobs are the URLs to be crawled, whereas each server contains a queue of URLs belonging to the same host. A job is served when the URL is crawled.
Closed queueing networks have been thoroughly studied by S. Lagershausen in [9], although one of their assumptions made for the closed queueing networks is that the servers of the queueing netwerk are connected in series and that every job receives service in the same order. For the closed queueing network studied in this thesis it is not known beforehand to which server the job goes after it has been served. Note that when in our case a job has been served, i.e. a URL has been crawled, it is removed from the workbench and a new URL is drawn from the sieve. Another research by K. Trivedi and R. Wagner [14] considers a decision model for closed queueing networks. In their model each server in the closed queueing network has the capability of processing b_i work units per time. The objective is to maximize the systems throughput, however, the speed of the servers, b_i, are their decision variables. In the closed queueing network studied in this thesis the goal is to be in certain states of the queueing system, because then the throughput of the web crawler is maximized.

At the beginning of the web crawling, in the initial phase, the workbench needs to be filled with a certain number of URLs. The number of different hosts in the workbench is interesting and this is related to the birthday problem. This short overview from the literature regarding the birthday problem is by no means complete. Related to this research is the generalization of the birthday problem to a total number of x birthdays by F.H. Matis [11]. In that paper the number of birthdays is generalized as well as the percentage, i.e., the probability that some share a birthday becomes at least q percent.
The birthday problem can also be generalized to an occupancy problem, see for example Gnedin et al. [5]. In the classical occupancy scheme, balls are thrown independently at a fixed infinite series of boxes, with probability p_j of hitting the *j*th box. When the boxes are considered to be the hosts and the balls as the URLs, we consider the number of different hosts for a fixed amount of URLs.
Another aspect is the uniformity assumption of the birthday problem that, among others, has been discussed by W. Knight and D. Bloom in [8]. The initial phase is related to the birthday problem when assuming uniformity to our 'birthdays' and therefore these studies about the uniformity of the birthday problem are rather interesting. In reality every birthday is not equally likely and in [8] it is shown that a non-uniform distribution of the birthdays increases the probability of sharing a birthday.
The birthday problem has also been extended to the probability that in a group of n randomly chosen people r people have the same birthday by E.H. Mckinney [12] and the probability that two birthdays fall within d adjacent days by J.I.Naus [13]. In the latter case the birthdays do not have to be on the same day, but they can be in a range of d adjacent days. These two examples do not necessarily apply to the problem studied in this thesis, but give an idea of how widely studied the birthday problem is.

Chapter 2

The Game Formalism of the BUbiNG Web Crawler

Below we describe the parts of the web crawler that play a role in the decision process. A more detailed description of the BUbiNG web crawler can be found in [2].

- The sieve is the data structure where URLs to be crawled are kept. This data structure is referred to as the *source*.
- The workbench is the data structure that represents the URLs already got from the sieve. The workbench is referred to as the *online structure*.
- The workbench virtualizer is a second data structure which is a sequence of (virtual) queues. It contains the URLs already extracted from the sieve, but they are not yet put in the workbench. This data structure is referred to as the *offline structure*.
- The distributor is the thread that has to make a decision. The distributor orchestrates the movement of URLs out of the sieve, either to the workbench or to the virtual queues, and loads as necessary URLs from the virtual queues into the workbench.

The distributor decides which URL to put in the online structure, the URL is either coming from the source or from the offline structure. The distributor wants to assign the URLs to the online structure in such a way that the number of hosts the URLs belong to is maximized.

2.1 The Game

The decision process can be described as a game as follows:

- The distributor is the *player* in the game.
- A *ball* is a URL to be crawled.
- The *color* of the ball represents the host the URL belongs to.
- Each ball of a same color is assigned a *number* consecutively.
- Removing a ball from the online structure is equivalent to crawling a URL.

The game board, the rules and the goal of the game are introduced in this chapter. Additionally, an example is given.

2.1.1 The Game Board

The parts of the web crawler that play a role in the decision process are now described in the game board.

The *source* produces the balls, where each ball has a color and a number. The balls with the same color are numbered consecutively. Every color has it's own probability and the probabilities are not the same for every color.

The *online structure* is a data structure that keeps the balls. It can keep up to, a fixed number, K balls at a time. The balls in the online structure are divided by color and each color is ordered by number.

The *offline structure* is a sequence of z queues, where some of them may be empty. Each queue contains some balls with possibly a mix of colors.

2.1.2 The Game Rules

The rules consist of several steps. The rules of the game are summarized in a flow chart, see Figure 2.1.

1. If the online structure is non-empty, one of the colors present in the online structure is chosen uniformly at random. The first ball of that color is removed from the online structure and disappears from the game.
2. The player has two choices:
    a) take the first ball of a non-empty queue in the offline structure (which is only possible if the offline structure is non-empty) and put it in the online structure in the appropriate place, or,

    b) ask a new ball from the source. If there is already some ball of the same color as the one extracted present in the offline structure, the player is forced to choose $ii)$ below. Otherwise the player can choose between the following two alternatives:

        i) put the ball in the online structure (in the appropriate place), or,

        ii) put the ball in the offline structure, at the end of one of the queues. There is one restriction: if k is the largest index of a queue containing a ball of the same color as the one extracted, the ball must be placed in some queue $Q[h]$ for some $h \geq k$.
3. Step 2 is repeated until there are exactly K balls in the online structure.

2.1.3 The Goal of the Game

The goal of the game is to get the largest possible number of colors in the online structure on average. This is achieved by finding a strategy for the decisions that have to be made. Due to the politeness limits, the web crawler has a high throughput when there is a possibility to crawl URLs from as may hosts as possible. The strategy of the player is based on some limited knowledge, namely:

- the player doesn't know what color the source will produce, not even knows the probability of each color to be drawn from the source nor the number of existing colors,
- there is a bound on the number of non-empty queues that can be used in the offline structure,
- searching in the queues in the offline structure and changing the positions of the balls in a queue in the offline structure is computationally demanding, because a disk head has to move physically.

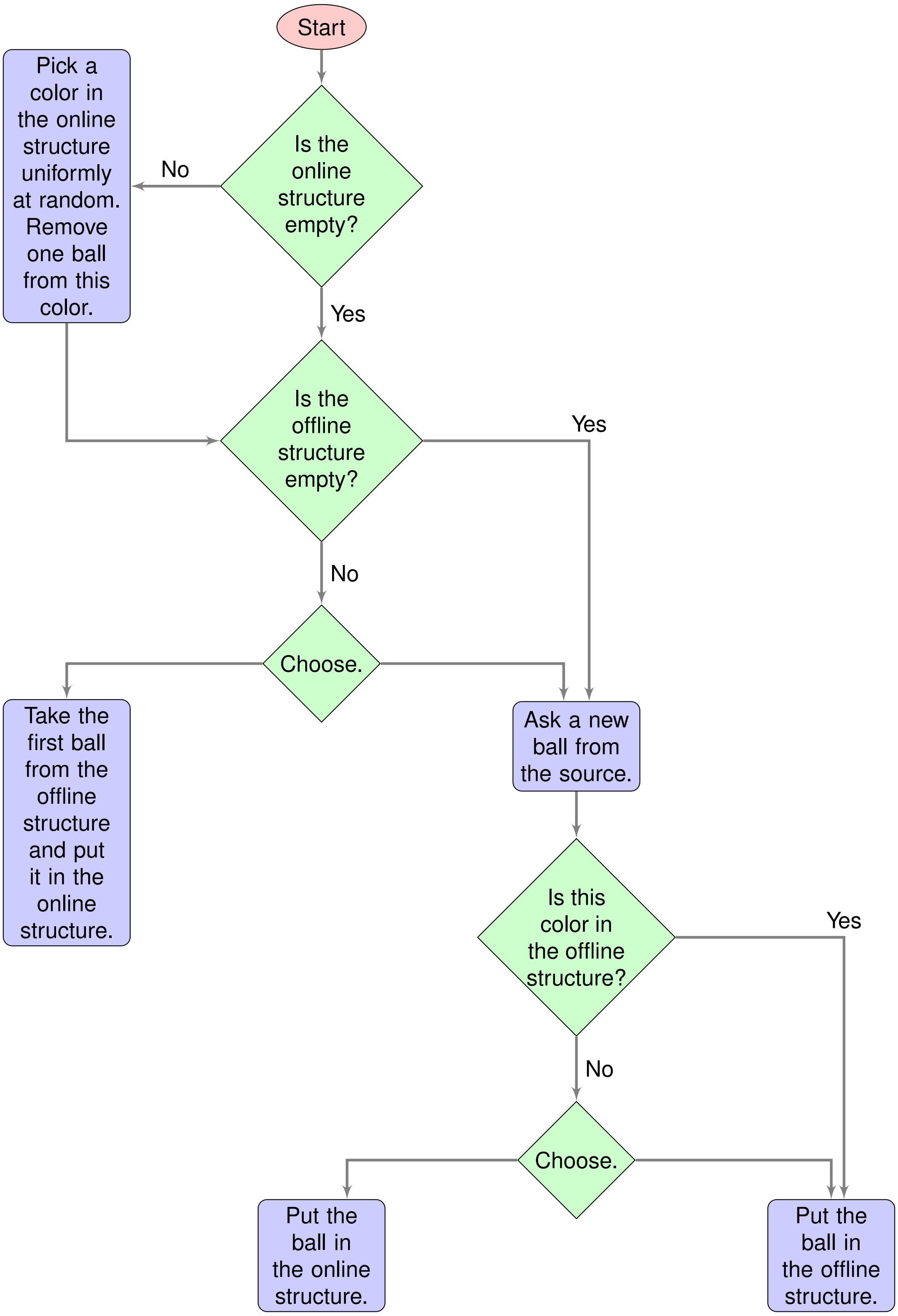

Start
Is the online structure empty?
No
Pick a color in the online structure uniformly at random. Remove one ball from this color.
Yes
Is the offline structure empty?
Yes
No
Choose.
Take the first ball from the offline structure and put it in the online structure.
Ask a new ball from the source.
Is this color in the offline structure?
Yes
No
Choose.
Put the ball in the online structure.
Put the ball in the offline structure.

Figure 2.1: *Flow chart of steps* 1 *and* 2 *of the game rules.*

2.2 An Example

Consider the following example, see Figure 2.2. It is assumed that the number of positions in the online structure, K, equals 8. The rules of the game are followed for this example, this is explained below the figure.

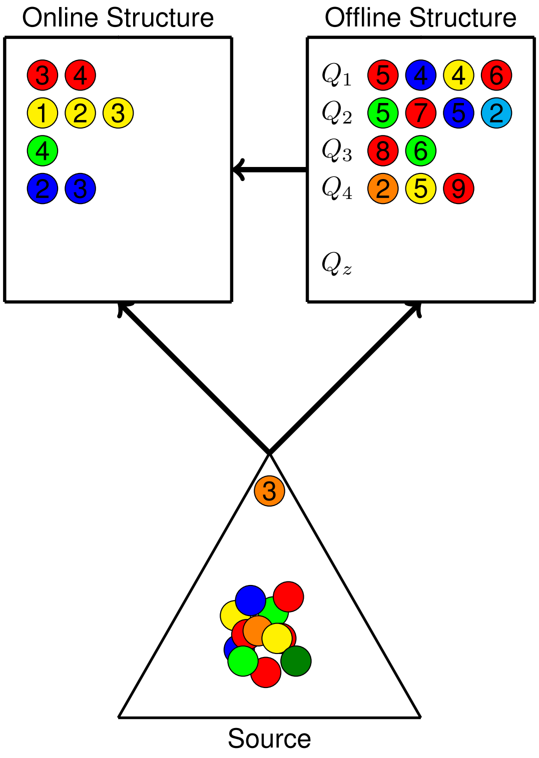


Figure 2.2: *An illustration of the game.*

1. Suppose that a yellow ball is removed, this is Yellow$\#1$
2. The player has to choose whether to do a or b.
   a) If a is chosen, ball Red$\#5$ is put in the online structure at the end of the queue with red balls.
   b) If b is chosen, there is no choice for the player since the ball from the source has a color which is already presented in the offline structure. Therefore the player is forced to put the ball in the offline structure at the end of queue i, $i \geq 4$.
3. Step 2 should be repeated, since the number of balls present in the online structure is smaller than K.

The example shows that it might happen that a color is not present anymore in the online structure, while it is still present in the offline structure. In the example this is the case for the orange balls. The result is that when a ball is drawn from the source that has a color that is not present in the online structure, in this case orange, the rules force the player to put the orange ball into the offline structure, while for the player it would have been best to put the orange ball in the online structure.

Furthermore the example shows that if the player wants to have a ball with a color that is not present in the online structure, for example cyan, the player either needs to accept a ball from the offline structure in the next 8 steps or hope to retrieve a ball from the source that has a color that is not present in the online structure nor in the offline structure..

Chapter 3

Initial Phase

The game that has been described in Chapter 2 is modeled and analyzed in both this chapter and Chapter 4 without the use of the offline structure. This means that the player of the game has no decision to make and that the rules of the game are the following:

1. If the online structure is non-empty, one of the colors present in the online structure is chosen uniformly at random. The first ball of that color is removed from the online structure and disappears from the game.
2. The player has no choice but to ask a new ball from the source. This ball goes into the online structure (in the appropriate place).

Before this procedure can start, the online structure needs to be filled with K balls. Since there is no offline structure, K balls are drawn from the source and are put in the online structure in the appropriate place. This initial phase is modeled and analyzed in this chapter, in Chapter 4 the implementation of the game without decisions is modeled and analyzed. When modeling the game, the assumption is made that the total number of colors equals C.

3.1 The Birthday Problem

Although it might not be obvious at first sight, the initial phase is related to the birthday problem. The relation is the following: suppose that the balls are the people and the colors are the birthdays. When the balls are put in the online structure, people with birthdays are put together in a room. In the classical birthday problem it is assumed that the birthdays correspond to a uniform distribution, see for example E. H. McKinney, [12]. In our case we assume that the colors can have a general distribution.

In order to analyze the initial phase, let $N(i,t)$ be the number of colors that have i balls in the online structure at time t. The number of different colors in the online structure at time t is equal to $\sum_{j=1}^{K} N(j,t)$. Remember that there is a total number of C colors and define $N(0,t) = C - \sum_{j=1}^{K} N(j,t)$ as the number of colors that are not represented in the online structure.

3.2 Number of Colors after drawing K Balls

In order to get an expression for the expected number of colors in the online structure after drawing K balls, let Z_i define a random variable:

$$Z_i = \begin{cases} 1, & \text{if there is no ball of color } i \text{ in the online structure,} \\ 0, & \text{otherwise.} \end{cases}$$

Let $p = (p_1, ..., p_C)$ be the probability vector such that p_i is the probability to draw a ball of color i from the source. The expression for the expected number of colors in the online structure is formulated in the first lemma.

Lemma 3.2.1 (Expected number of colors at $t = K$)**.** *The expected number of colors in the online structure at time* $t = K$ *is*

$$\mathbb{E}\left[\sum_{j=1}^{K} N(j,K)\right] = C - \sum_{i=1}^{C}(1-p_i)^K. \tag{3.1}$$

Proof. By definition of Z_i, $\mathbb{E}\left[N(0,K)\right] = \sum_{i=1}^{C}\mathbb{E}\left[Z_i\right]$. A queue is empty after K steps with probability $(1-p_i)^K$, where p_i is the probability for color i to be drawn from the source. Hence, $\mathbb{E}\left[Z_i\right] = (1-p_i)^K$ and this gives an expression for $\mathbb{E}\left[N(0,K)\right]$:

$$\mathbb{E}\left[N(0,K)\right] = \sum_{i=1}^{C}(1-p_i)^K.$$

□

Note that this lemma can be extended such that it holds for every $t \leq K$.

Corollary 3.2.1 (Expected number of colors for uniform distribution of p at $t = K$)**.** *When* p *corresponds to a uniform distribution, the expected number of colors in the online structure at time* $t = K$ *is*

$$\mathbb{E}\left[\sum_{j=1}^{K} N(j,K)\right] = C\left(1 - \left(1 - \frac{1}{C}\right)^K\right).$$

If the number of colors in the online structure, $\sum_{j=1}^{K} N(j,t)$, is known, it is also possible to calculate the probability to draw a color from the source that is not present in the online structure, $P_{new} = \frac{C-\sum_{j=1}^{K} N(j,K)}{C}$, and the probability to draw a color from the source that is present in the online structure, $P_{old} = \frac{\sum_{j=1}^{K} N(j,K)}{C}$. Limiting behavior when p corresponds to a uniform distribution is the following:

$$\begin{aligned}\sum_{j=1}^{K}\mathbb{E}\left[N(j,K)\right] &= C\left(1 - \left(1 - \frac{1}{C}\right)^K\right),\\ &= C\left(1 - \left(1 - \frac{K}{C} + \mathcal{O}\left(\frac{K}{C^2}\right)\right)\right),\\ &= K - \mathcal{O}\left(\frac{K}{C}\right).\end{aligned}$$

This means that with high probability every position in the online structure contains a different color at $t = K$. If $C = K$ the following relation is obtained:

$$\begin{aligned}\sum_{j=1}^{K}\mathbb{E}\left[N(j,K)\right] &= K\left(1 - \left(1 - \frac{1}{K}\right)^K\right),\\ &= K\left(1 - \frac{1}{e} + \frac{1}{2Ke} + \mathcal{O}\left(\frac{1}{K^2}\right)\right),\\ &= 0.6321K + \frac{1}{2e} + \mathcal{O}\left(\frac{1}{K}\right).\end{aligned} \tag{3.2}$$

Note that this analysis for a uniform distribution of p is a generalized version of calculating the average number of unique birthdays in a group, see S. Goldberg [6] for the calculations with n people and 365 birthdays.

Using Lemma 3.2.1 the upcoming theorem can be shown.

Theorem 3.2.1 (Maximization of the expected number of colors at $t = K$)**.** *The expected number of colors in the online structure at time* $t = K$ *is maximized when the probabilities for each color to be drawn from the source correspond to a uniform distribution.*

Proof. The goal is to maximize $C - \sum_{i=1}^{C}(1-p_i)^K$ subject to $\sum_{i=1}^{C} p_i = 1$. According to the method of Lagrange multipliers $\Lambda(p_1, ..., p_C, \lambda) = C - \sum_{i=1}^{C}(1-p_i)^K + \lambda\left(\sum_{i=1}^{C} p_i - 1\right)$. Note that derivatives of $\sum_{i=1}^{C}(1-p_i)$ and $\sum_{i=1}^{C} p_i$ with respect to p_i exist for all p_i. It follows that

$$\begin{aligned}\frac{d\Lambda}{d\lambda} &= \sum_{i=1}^{C} p_i - 1 = 0, \\ \frac{d\Lambda}{dp_i} &= (K-1)(1-p_i)^{K-1} + \lambda = 0, \qquad \forall p_i.\end{aligned}$$

Therefore $\lambda = -(K-1)(1-p_i)^{K-1}\ \forall p_i$ and so $p_i = \frac{1}{C}\ \forall p_i$. □

In the next chapter it is shown what happens to $\sum_{j=1}^{K} \mathbb{E}\left[N(j,t)\right]$, $t \geq K$.

Chapter 4

Dynamics: no Offline Structure

After the online structure has been filled with K balls, the game is modeled without the offline structure. The model is a closed queueing network with K customers in the network. Again it is assumed that the total number of colors equals C. When the online structure is filled with K balls, the queueing model is analyzed with the help of recurrence relations. Every time step one ball of a color is removed from the online structure uniformly at random and one ball is drawn from the source according to the probability distribution $p = (p_1, ..., p_C)$, where p_i is the probability to draw a ball from the source with color i. This ball is added to the online structure. We are interested in the number of colors that is present in the online structure.

4.1 Closed Queueing Network

Let $(q_1, ..., q_C)$ be the state of the Markov chain, were q_i is the number of balls of color i in the online structure. The balance equations for the closed queueing network are:

$$P(q_1, ..., q_C) \quad = \quad \sum_{i,j:q_j \geq 1}^{C} P((q_1, ..., q_C) + e_i - e_j) \frac{1}{NNZ(q + e_i - e_j)} p_j,$$

where $NNZ(x)$ is the number of non-zeros in the vector x. The probability to draw a color from the source that is not present in the online structure, P_{new}, is:

$$P_{new} \quad = \quad \sum_{k:(q-e_j)_k = 0} p_k.$$

In particular, when all colors are equally likely to be drawn from the source, so $p_i = \frac{1}{C}$, $i = 1, ..., C$, it holds that $P_{new} = \frac{C - NNZ(q-e_j)}{C}$. If C tends to infinity, this fraction tends to 1. Intuitively, we may expect that there are almost K different colors in the online structure when C is much larger than K.

Let $N(i,t)(q(t)) = |\{c : q_c(t) = i\}|$, where c is a color, be the number of queues of length i at time t. For ease of notation we write $N(i,t)$ instead of $N(i,t)(q(t))$. Note that this notation corresponds to the notation introduced in Chapter 3. Remember that the number of non-zero queues is equal to $\sum_{j=1}^{K} N(j,t)$, this is the number of different colors in the online structure at time t, and the number of colors that are not represented in the online structure is defined by $N(0,t) = C - \sum_{j=1}^{K} N(j,t)$. Note that $\sum_{j=0}^{K} jN(j,t) = K$, this is the total number of positions in the online structure.

4.2 Homogeneous Colors

To analyze the expected number of colors in stationarity, recurrence relations are used. For the dynamics it is assumed that the probability for each color to be drawn from the source corresponds to a uniform distribution. If there are $N(i,t)$ queues of length i at time t, $t \geq K$, there are several of possibilities for $N(i,t+1)$. The cases $i=0$, $i=1$ and $i=K$ are discussed separately. Recall that $N(0,t)$ is the number of colors *not* present in the online structure. The first case that is discussed is $i=0$.

- $N(0,t+1) = N(0,t) - 1$
  This happens when a ball is removed from a queue that has length greater than 1 and afterwards a ball is drawn from the source that has a color that is not present in the online structure.

- $N(0,t+1) = N(0,t)$
  The number of empty queues remains the same if:
  - a ball is removed from a queue with length greater than 1 and afterwards a ball is added to the online structure that has a color that is already present in the online structure, or,
  - a ball is removed from a queue of length 1 and afterwards a ball is added that has a color is not present in the online structure.

- $N(0,t+1) = N(0,t) + 1$
  If a ball is removed from a queue with length 1 and a ball is added that has a color that is already present in the online structure, then the number of colors not represented in the online structure increases. Note that the number of colors already present in the online structure is decreased by 1 since a ball from a queue with length 1 has been removed.

All these cases are summarized in the following equation:

$$N(0,t+1) \quad = \quad \begin{cases} N(0,t)-1, & \text{w.p.} \quad \left(1 - \frac{N(1,t)}{\sum_{j=1}^{K} N(j,t)}\right) \frac{C - \sum_{j=1}^{K} N(j,t)}{C}, \\ N(0,t), & \text{w.p.} \quad \left(1 - \frac{N(1,t)}{\sum_{j=1}^{K} N(j,t)}\right) \frac{\sum_{j=1}^{K} N(j,t)}{C} + \\ & \qquad \frac{N(1,t)}{\sum_{j=1}^{K} N(j,t)} \left(1 - \frac{\sum_{j=1}^{K} N(j,t) - 1}{C}\right), \\ N(0,t)+1, & \text{w.p.} \quad \frac{N(1,t)}{\sum_{j=1}^{K} N(j,t)} \frac{\sum_{j=1}^{K} N(j,t) - 1}{C}. \end{cases}$$

Note that there are only three options, because it is impossible that $N(0,t+1) = N(0,t) - K$ or $N(0,t+1) = N(0,t) + K$, with $K \geq 2$. The next special case is $i=1$.

- $N(1,t+1) = N(1,t) - 2$
  When a ball is removed from a queue with length one and afterwards a ball is drawn from the source from which the queue length was one, the number of queues with length one is decreasing by two.

- $N(1,t+1) = N(1,t) - 1$
  The number of queues with length one decreases by 1 if:
  - a ball is removed from a queue with length one and thereafter a ball is drawn from the source that does not have a color that is not present in the online structure nor a color that corresponds to a queue with length 1, or,
  - a ball is removed from a queue with at least length three and afterwards a ball is drawn with a color corresponding to a queue with length 1.

- $N(1,t+1) = N(1,t)$
  The number of queues with length one remains the same if:
  - a ball is removed with a color corresponding to a queue with length one and afterwards a ball is added to the online structure that has a color that is not present in the online structure, or,
  - a ball is removed from a queue that contains at least three balls and a ball is added of a color with at least two balls in the online structure, or,

    - a ball is removed from a queue with length two and afterwards a ball is added to a queue with length one.

- $N(1,t+1) = N(1,t) + 1$
  An increase of the number of queues with length one happens if:

    - a ball with a color corresponding to a queue with a length of at least three is removed followed by a draw of a ball from the source that has a color that is not present in the online structure, or,
    - the number of queues with length two decreases and thereafter a ball is drawn from the source that has a color that corresponds to a queue with length greater than 1.

- $N(1,t+1) = N(1,t) + 2$
  If a ball is removed that has a color corresponding to a queue with length two and afterwards a ball is drawn from the source that has a color that is not present in the online structure, then the number of queues with length one increases by two.

All these possibilities lead to the following equation:

$$N(1,t+1) \quad = \quad \begin{cases} N(1,t)-2, & \text{w.p.} \quad \frac{N(1,t)}{\sum_{j=1}^{K} N(j,t)} \frac{N(1,t)-1}{C}, \\[2ex] N(1,t)-1, & \text{w.p.} \quad \frac{N(1,t)}{\sum_{j=1}^{K} N(j,t)} \frac{\sum_{j=2}^{K} N(j,t)}{C} + \left(1 - \frac{N(1,t)+N(2,t)}{\sum_{j=1}^{K} N(j,t)}\right) \frac{N(1,t)}{C}, \\[2ex] N(1,t), & \text{w.p.} \quad \frac{N(1,t)}{\sum_{j=1}^{K} N(j,t)} \frac{C-\sum_{j=1}^{K} N(j,t)+1}{C} + \\ & \quad \left(1 - \frac{N(1,t)+N(2,t)}{\sum_{j=1}^{K} N(j,t)}\right) \frac{\sum_{j=2}^{K} N(j,t)}{C} + \frac{N(2,t)}{\sum_{j=1}^{K} N(j,t)} \frac{N(1,t)+1}{C}, \\[2ex] N(1,t)+1, & \text{w.p.} \quad \left(1 - \frac{N(1,t)+N(2,t)}{\sum_{j=1}^{K} N(j,t)}\right) \frac{C-\sum_{j=1}^{K} N(j,t)}{C} + \frac{N(2,t)}{\sum_{j=1}^{K} N(j,t)} \frac{\sum_{j=2}^{K} N(j,t)-1}{C}, \\[2ex] N(1,t)+2, & \text{w.p.} \quad \frac{N(2,t)}{\sum_{j=1}^{K} N(j,t)} \frac{C-\sum_{j=1}^{K} N(j,t)}{C}. \end{cases}$$

The reasoning for $i = 2, .., K-1$ is very similar to the case that $i = 1$. Since some differences occur, the possibilities are described still.

- $N(i,t+1) = N(i,t) - 2$
  When a ball is removed from a queue with length i and afterwards a ball is drawn from the source that has a color from which the queue length was i, the number of queues with length i is decreasing by two.

- $N(i,t+1) = N(i,t) - 1$
  The number of queues with length i decreases by 1 if:

    - a ball is removed from a queue with length i and afterwards a ball is drawn with a color that does not corresponds to a queue with length $i-1$ and length i, or,
    - a ball is removed from a queue that does not have length i nor $i+1$ and afterwards a ball is drawn with a color corresponding to a queue with length i.

- $N(i,t+1) = N(i,t)$
  The number of queues with length i remains the same if:

    - a ball is removed with a color corresponding to a queue with length i and afterwards a ball is added with a color corresponding to a queue with length $i-1$, or,
    - a ball is removed from a queue that does not have length i nor length $i+1$ and afterwards a ball is added to a queue that does not have length $i-1$ nor length i. Note that it can happen that a ball is removed from a queue with length $i-1$. If this case the probability that a ball is added to a queue that does not have length $i-1$ nor length i is different than when a ball is removed from a queue that does not have length $i-1$, i and $i+1$. Or,

- a ball is removed from a queue with length $i+1$ and afterwards a ball is added to a queue with length i.

- $N(i,t+1) = N(i,t)+1$
  An increase of the number of queues with length one happens if:
  - a ball with a color corresponding to a queue that does not have length i nor length $i+1$ is removed from the online structure, followed by a draw of a ball from the source corresponding to a queue with length $i-1$. Note that it can happen that a ball is removed from a queue with length $i-1$. If this is the case, the probability that a ball is added to a queue that has length $i-1$ is different than when a ball is removed from a queue that does not have length $i-1$, i and $i+1$. Or,
  - the number of queues with length $i+1$ decreases and thereafter a ball is drawn from the source that has a color that corresponds to a queue that does not have length $i-1$ nor length i.

- $N(i,t+1) = N(i,t)+2$
  If a ball is removed corresponding to a queue with length $i+1$ and afterwards a ball is drawn from the source with a color corresponding to a queue with length $i-1$, then the number of queues with length i increases by two.

The possibilities lead to the following equation:

$$N(i,t+1) \quad = \quad \begin{cases} N(i,t)-2 & \text{w.p.} \quad \frac{N(i,t)}{\sum_{j=1}^{K} N(j,t)} \frac{N(i,t)-1}{C}, \\ N(i,t)-1 & \text{w.p.} \quad \frac{N(i,t)}{\sum_{j=1}^{K} N(j,t)} \left(1 - \frac{N(i-1,t)+N(i,t)}{C}\right) + \\ & \left(1 - \frac{N(i,t)+N(i+1,t)}{\sum_{j=1}^{K} N(j,t)}\right) \frac{N(i,t)}{C}, \\ N(i,t) & \text{w.p.} \quad \frac{N(i,t)}{\sum_{j=1}^{K} N(j,t)} \frac{N(i-1,t)+1}{C} + \\ & \left(1 - \frac{N(i-1,t)+N(i,t)+N(i+1,t)}{\sum_{j=1}^{K} N(j,t)}\right) \left(1 - \frac{N(i,t)+N(i-1,t)}{C}\right) + \\ & \frac{N(i-1,t)}{\sum_{j=1}^{K} N(j,t)} \left(1 - \frac{N(i,t)+N(i-1,t)-1}{C}\right) + \\ & \frac{N(i+1,t)}{\sum_{j=1}^{K} N(j,t)} \frac{N(i,t)+1}{C}, \\ N(i,t)+1 & \text{w.p.} \quad \left(1 - \frac{N(i-1,t)+N(i,t)+N(i+1,t)}{\sum_{j=1}^{K} N(j,t)}\right) \frac{N(i-1,t)}{C} + \frac{N(i-1,t)}{\sum_{j=1}^{K} N(j,t)} \frac{N(i-1,t)-1}{C} + \\ & \frac{N(i+1,t)}{\sum_{j=1}^{K} N(j,t)} \left(1 - \frac{N(i-1,t)+N(i,t)+1}{C}\right), \\ N(i,t)+2 & \text{w.p.} \quad \frac{N(i+1,t)}{\sum_{j=1}^{K} N(j,t)} \frac{N(i-1,t)}{C}. \end{cases}$$

Note that for $i > \frac{K}{2}$, $N(i,t)$ can only be 0 or 1. Moreover, only one $N(i,t)$, $i > \frac{K}{2}$ can be equal to 1. The last case that is discussed is $i = K$.

- $N(K,t+1) = N(K,t)-1$
  When a ball is removed from a queue with length K and afterwards a ball is added to a queue that does not have length $K-1$, then the number of queues with length K decreases by one.

- $N(K,t+1) = N(K,t)$
  The number of queues with length K remains the same if:
  - a ball is removed from a queue with length K and afterwards a ball is added to a queue with length $K-1$, or,

  - a ball is removed from a queue that does not have length K and after that a ball is drawn from the source that corresponds to a queue with length not equal to $K-1$. Note that when a ball is removed from a queue that does not have length K, it might happen that a ball is removed from a queue with length $K-1$. In the latter case the probability that a ball is added to a queue with length not equal to $K-1$ is different than when a ball is removed from a queue that does not have length $K-1$ nor length K.

- $N(K, t+1) = N(K, t) + 1$
  Suppose that a ball is removed from a queue with length not equal to $K-1$ and afterwards a ball is drawn from the source corresponding to a queue with length $K-1$, then the number of queues with length K is increased by one. Note that when a ball is removed from a queue that does not have length $K-1$, it can happen dat a ball is removed from a queue with length K. In that case, the probability that a ball is added to a queue with length $K-1$ is different than when a ball is removed from a queue that does not have length $K-1$ nor length K.

This is summarized in the following equation:

$$N(K,t+1) \quad = \quad \begin{cases} N(K,t)-1, & \text{w.p.} \quad \frac{N(K,t)}{\sum_{j=1}^{K} N(j,t)} \left(1 - \frac{N(K-1,t)+1}{C}\right), \\ N(K,t), & \text{w.p.} \quad \frac{N(K,t)}{\sum_{j=1}^{K} N(j,t)} \frac{N(K-1,t)+1}{C} + \\ & \quad \left(1 - \frac{N(K-1,t)+N(K,t)}{\sum_{j=1}^{K} N(j,t)}\right) \left(1 - \frac{N(K-1,t)}{C}\right) + \\ & \quad \frac{N(K-1,t)}{\sum_{j=1}^{K} N(j,t)} \left(1 - \frac{N(K-1,t)-1}{C}\right), \\ N(K,t)+1, & \text{w.p.} \quad \left(1 - \frac{N(K-1,t)+N(K,t)}{\sum_{j=1}^{K} N(j,t)}\right) \frac{N(K-1,t)}{C} + \frac{N(K-1,t)}{\sum_{j=1}^{K} N(j,t)} \frac{N(K-1,t)-1}{C}. \end{cases}$$

Now it is possible to proof the following theorem.

Theorem 4.2.1 (Expected number of colors in stationary regime with homogeneous colors.)**.** *When there are K positions in the online structure, the total number of colors is C and the probability for each color to be drawn from the source is $\frac{1}{C}$ it follows that the mean field approximation of $\mathbb{E}\left[\sum_{j=1}^{K} N(j,t)\right]$, the number of colors in the online structure, is $\frac{KC}{K+C-1}$.*

Proof. From the dynamics for $N(i, t+1)$, $i = 0, .., K$, these expectations follow.

$$\begin{aligned} \mathbb{E}[N(0,t+1)] &= \mathbb{E}[N(0,t)] + \mathbb{E}\left[\frac{N(1,t)}{\sum_{j=1}^{K} N(j,t)}\right] + \frac{\sum_{j=1}^{K} \mathbb{E}[N(j,t)]}{C} - \frac{1}{C}\mathbb{E}\left[\frac{N(1,t)}{\sum_{j=1}^{K} N(j,t)}\right] - 1 \quad (4.1) \\ \mathbb{E}[N(1,t+1)] &= \mathbb{E}[N(1,t)] + \frac{2}{C}\mathbb{E}\left[\frac{N(1,t)}{\sum_{j=1}^{K} N(j,t)}\right] - \mathbb{E}\left[\frac{N(1,t)}{\sum_{j=1}^{K} N(j,t)}\right] - \frac{\mathbb{E}[N(1,t)]}{C} + \\ & \quad \mathbb{E}\left[\frac{N(2,t)}{\sum_{j=1}^{K} N(j,t)}\right] - \frac{1}{C}\mathbb{E}\left[\frac{N(2,t)}{\sum_{j=1}^{K} N(j,t)}\right] + 1 - \frac{\sum_{j=1}^{K} \mathbb{E}[N(j,t)]}{C} \\ \mathbb{E}[N(i,t+1)] &= \mathbb{E}[N(i,t)] + \frac{2}{C}\mathbb{E}\left[\frac{N(i,t)}{\sum_{j=1}^{K} N(j,t)}\right] - \mathbb{E}\left[\frac{N(i,t)}{\sum_{j=1}^{K} N(j,t)}\right] - \frac{\mathbb{E}[N(i,t)]}{C} + \\ & \quad \mathbb{E}\left[\frac{N(i+1,t)}{\sum_{j=1}^{K} N(j,t)}\right] - \frac{1}{C}\mathbb{E}\left[\frac{N(i+1,t)}{\sum_{j=1}^{K} N(j,t)}\right] + \frac{\mathbb{E}[N(i-1,t)]}{C} - \frac{1}{C}\mathbb{E}\left[\frac{N(i-1,t)}{\sum_{j=1}^{K} N(j,t)}\right] \\ \mathbb{E}[N(K,t+1)] &= \mathbb{E}[N(K,t)] + \frac{1}{C}\mathbb{E}\left[\frac{N(K,t)}{\sum_{j=1}^{K} N(j,t)}\right] - \mathbb{E}\left[\frac{N(K,t)}{\sum_{j=1}^{K} N(j,t)}\right] + \frac{\mathbb{E}[N(K-1,t)]}{C} - \\ & \quad \frac{1}{C}\mathbb{E}\left[\frac{N(K-1,t)}{\sum_{j=1}^{K} N(j,t)}\right] \end{aligned}$$

Because of stationarity $\mathbb{E}[N(i,t+1)] = \mathbb{E}[N(i,t)]$ for $i = 0, ..., K$. However, it is not possible to solve these equations and therefore a mean field approximation is applied. The random variable $N(i,t)$ is replaced by a constant $\nu(i,t)$. Applying stationarity and the mean field approximation to (4.1) results in:

$$0 = \frac{\nu(1,t)}{\sum_{j=1}^{K} \nu(j,t)} + \frac{\sum_{j=1}^{K} \nu(j,t)}{C} - \frac{1}{C} \frac{\nu(1,t)}{\sum_{j=1}^{K} \nu(j,t)} - 1.$$

Solving $\nu(1,t)$ in terms of $\sum_{j=1}^{K} \nu(j,t)$ and C gives

$$\nu(1,t) \quad = \quad \frac{\sum_{j=1}^{K} \nu(j,t) \left[C - \sum_{j=1}^{K} \nu(j,t)\right]}{C-1}.$$

Rewriting the other equations in a similar way results in the following recursive equations for $\nu(i,t)$, $i = 1, ..., K$.

$$\begin{aligned}
\nu(1,t) &= \frac{\sum_{j=1}^{K} \nu(j,t) \left[C - \sum_{j=1}^{K} \nu(j,t)\right]}{C-1} && (4.2)\\
\nu(2,t) &= \left[\frac{C + \sum_{j=1}^{K} \nu(j,t) - 2}{C-1}\right] \nu(1,t) - \frac{\sum_{j=1}^{K} \nu(j,t)}{C-1} \left[C - \sum_{j=1}^{K} \nu(j,t)\right] \\
&= \left[\frac{C + \sum_{j=1}^{K} \nu(j,t) - 2}{C-1}\right] \nu(1,t) - \frac{\sum_{j=1}^{K} \nu(j,t)}{C-1} \nu(0,t) && (4.3)\\
\nu(i+1,t) &= \left[\frac{C + \sum_{j=1}^{K} \nu(j,t) - 2}{C-1}\right] \nu(i,t) - \frac{\sum_{j=1}^{K} \nu(j,t) - 1}{C-1} \nu(i-1,t) && (4.4)\\
\nu(K,t) &= \frac{\sum_{j=1}^{K} \nu(j,t) - 1}{C-1} \nu(K-1,t)
\end{aligned}$$

Equation (4.4) holds for $i = 2, .., K-2$. The claim is that the solution of these equations is $\nu(i,t) = \left[C - \sum_{j=1}^{K} \nu(j,t)\right] \frac{\sum_{j=1}^{K} \nu(j,t)}{C-1} \left(\frac{\sum_{j=1}^{K} \nu(j,t)-1}{C-1}\right)^{i-1}$ and this is shown by induction. From (4.2) and (4.3) it is known that the desired relation holds for $i = 1$ and $i = 2$.

$$\begin{aligned}
\nu(1,t) &= \frac{\sum_{j=1}^{K} \nu(j,t) \left[C - \sum_{j=1}^{K} \nu(j,t)\right]}{C-1} \\
\nu(2,t) &= \frac{\sum_{j=1}^{K} \nu(j,t)}{C-1} \frac{\sum_{j=1}^{K} \nu(j,t) - 1}{C-1} \left[C - \sum_{j=1}^{K} \nu(j,t)\right]
\end{aligned}$$

Now suppose that for $\nu(i-1,t)$ and $\nu(i,t)$ it holds that

$$\begin{aligned}
\nu(i-1,t) &= \left[C - \sum_{j=1}^{K} \nu(j,t)\right] \frac{\sum_{j=1}^{K} \nu(j,t)}{C-1} \left(\frac{\sum_{j=1}^{K} \nu(j,t) - 1}{C-1}\right)^{i-2}, \\
\nu(i,t) &= \left[C - \sum_{j=1}^{K} \nu(j,t)\right] \frac{\sum_{j=1}^{K} \nu(j,t)}{C-1} \left(\frac{\sum_{j=1}^{K} \nu(j,t) - 1}{C-1}\right)^{i-1},
\end{aligned}$$

then it should be true that $\nu(i+1,t) = \left[C - \sum_{j=1}^{K} \nu(j,t)\right] \frac{\sum_{j=1}^{K} \nu(j,t)}{C-1} \left(\frac{\sum_{j=1}^{K} \nu(j,t)-1}{C-1}\right)^{i}$. From equation

(4.4) it follows that:

$$\begin{aligned}
\nu(i+1,t) &= \left[1+\frac{\sum_{j=1}^K \nu(j,t)-1}{C-1}\right]\nu(i,t)-\frac{\sum_{j=1}^K \nu(j,t)-1}{C-1}\nu(i-1,t),\\
&= \left[1+\frac{\sum_{j=1}^K \nu(j,t)-1}{C-1}\right]\left[C-\sum_{j=1}^K \nu(j,t)\right]\frac{\sum_{j=1}^K \nu(j,t)}{C-1}\left(\frac{\sum_{j=1}^K \nu(j,t)-1}{C-1}\right)^{i-1}-\\
&\quad \frac{\sum_{j=1}^K \nu(j,t)-1}{C-1}\left[C-\sum_{j=1}^K \nu(j,t)\right]\frac{\sum_{j=1}^K \nu(j,t)}{C-1}\left(\frac{\sum_{j=1}^K \nu(j,t)-1}{C-1}\right)^{i-2},\\
&= \left[C-\sum_{j=1}^K \nu(j,t)\right]\frac{\sum_{j=1}^K \nu(j,t)}{C-1}\left(\frac{\sum_{j=1}^K \nu(j,t)-1}{C-1}\right)^{i},
\end{aligned}$$

for $i=2,...,K-2$. Obviously, since the relation holds for $i=K-2$ it also folds for $i=K-1$. So it is true that:

$$\begin{aligned}
\nu(i,t) &= \left[C-\sum_{j=1}^K \nu(j,t)\right]\frac{\sum_{j=1}^K \nu(j,t)}{C-1}\left(\frac{\sum_{j=1}^K \nu(j,t)-1}{C-1}\right)^{i-1},\\
&= \left[C-\sum_{j=1}^K \nu(j,t)\right]\frac{\sum_{j=1}^K \nu(j,t)}{C-1}\frac{C-1}{\sum_{j=1}^K \nu(j,t)-1}\left(\frac{\sum_{j=1}^K \nu(j,t)-1}{C-1}\right)^{i}.
\end{aligned}$$

This relation holds for $i=1,...,K$, by definition $\nu(0,t)=C-\sum_{j=1}^K \nu(j,t)$. Note that $\sum_{i=0}^K i\nu(i,t)=K$, so

$$K=\sum_{i=0}^K i\nu(i,t)=\left[C-\sum_{j=1}^K \nu(j,t)\right]\frac{\sum_{j=1}^K \nu(j,t)}{\sum_{j=1}^K \nu(j,t)-1}\sum_{i=0}^K i\left(\frac{\sum_{j=1}^K \nu(j,t)-1}{C-1}\right)^{i}.$$

Applying $\sum_{i=0}^\infty ix^i=\frac{x}{(x-1)^2}$, $|x|<1$ is possible because K is large and $\frac{\sum_{j=1}^K \nu(j,t)-1}{C-1}<1$. Since there are K positions in the online structure it follows $\sum_{j=1}^K \nu(j,t)\neq C$ and therefore the fraction can not be equal to 1. Note that if $C=K$ then $\sum_{j=1}^K \nu(j,t)\neq K$, because if this would be true then $\nu(1,t)=K$. By (4.2) this results in $\nu(1,t)=0$, a contradiction. Note that by replacing the finite sum by an infinite sum the residual term is

$$\begin{aligned}
\sum_{i=K+1}^\infty i\left(\frac{\sum_{j=1}^K \nu(j,t)-1}{C-1}\right)^{i} &= \left(\frac{\sum_{j=1}^K \nu(j,t)-1}{C-1}\right)^{K+1}\sum_{i=0}^\infty (i+K+1)\left(\frac{\sum_{j=1}^K \nu(j,t)-1}{C-1}\right)^{i},\\
&= (K+1)\left(\frac{\sum_{j=1}^K \nu(j,t)-1}{C-1}\right)^{K+1}\sum_{i=0}^\infty \left(\frac{\sum_{j=1}^K \nu(j,t)-1}{C-1}\right)^{i}+\\
&\quad \left(\frac{\sum_{j=1}^K \nu(j,t)-1}{C-1}\right)^{K+1}\sum_{i=0}^\infty i\left(\frac{\sum_{j=1}^K \nu(j,t)-1}{C-1}\right)^{i},
\end{aligned}$$

which is exponentially decreasing in K. So we have

$$\begin{aligned}
K &= \left[C - \sum_{j=1}^{K} \nu(j,t)\right] \frac{\sum_{j=1}^{K} \nu(j,t)}{\sum_{j=1}^{K} \nu(j,t) - 1} \left[\frac{\frac{\sum_{j=1}^{K} \nu(j,t)-1}{C-1}}{\left(\frac{\sum_{j=1}^{K} \nu(j,t)-1}{C-1} - 1\right)^2} - \mathcal{O}\left(\left(\frac{\sum_{j=1}^{K} \nu(j,t) - 1}{C-1}\right)^{K+1}\right)\right], \\
&= \left[C - \sum_{j=1}^{K} \nu(j,t)\right] \frac{\sum_{j=1}^{K} \nu(j,t)}{\sum_{j=1}^{K} \nu(j,t) - 1} \frac{\sum_{j=1}^{K} \nu(j,t) - 1}{C-1} \left(\frac{C-1}{\sum_{j=1}^{K} \nu(j,t) - C}\right)^2 - \\
&\quad \left[C - \sum_{j=1}^{K} \nu(j,t)\right] \frac{\sum_{j=1}^{K} \nu(j,t)}{\sum_{j=1}^{K} \nu(j,t) - 1} \mathcal{O}\left(\left(\frac{\sum_{j=1}^{K} \nu(j,t) - 1}{C-1}\right)^{K+1}\right), \\
&= \frac{(C-1)\sum_{j=1}^{K} \nu(j,t)}{C - \sum_{j=1}^{K} \nu(j,t)} - \left[C - \sum_{j=1}^{K} \nu(j,t)\right] \frac{\sum_{j=1}^{K} \nu(j,t)}{\sum_{j=1}^{K} \nu(j,t) - 1} \mathcal{O}\left(\left(\frac{\sum_{j=1}^{K} \nu(j,t) - 1}{C-1}\right)^{K+1}\right),
\end{aligned}$$

which results in the approximation of $\sum_{j=1}^{K} \nu(j,t)$ being $\frac{KC}{K+C-1}$. Note that there is a devision by $C - \sum_{j=1}^{K} \nu(j,t)$, but $\sum_{j=1}^{K} \nu(j,t) \neq C$, as has been explained before. □

By definition of P_{new}, $P_{new} = \frac{C - \sum_{j=1}^{K} \nu(j,t)}{C}$, the probability to draw a ball from the source that is not present in the online structure approximates $\frac{C-1}{K+C-1}$. This theorem has some immediate results.

Corollary 4.2.1 (Expected number of queues with length i with equally likely colors.)**.** *When there are K positions in the online structure, the total number of colors is C and the probability for each color to be drawn from the source us $\frac{1}{C}$ it follows that $\mathbb{E}[N(i,t)]$ approximates $\frac{KC^2(K-1)^{i-1}}{(K+C-1)^{i+1}}$.*

Corollary 4.2.2 (Limiting behavior with equally likely colors.)**.** *When there are K positions in the online structure, the total number of colors C tends to infinity and the probability for each color to be drawn from the source is $\frac{1}{C}$ it follows that $\sum_{j=1}^{C} N(j,t) = K$ and $P_{new} = 1$.*

Corollary 4.2.3. *Suppose there are K positions in the online structure, the total number of colors is C and the probability for each color to be drawn from the source is $\frac{1}{C}$. If both K and C are multiplied with the same constant a, then $\sum_{j=1}^{C} N(j,t)$ approximates $a\frac{KC}{K+C}$ and P_{new} approximates $\frac{C}{K+C}$.*

Proof. It has been shown that

$$\begin{aligned}
\sum_{j=1}^{C} N(j,t) &= \frac{KC}{K+C-1}, \\
&= \frac{KC}{K+C}\left[\frac{1}{1 - \frac{1}{K+C}}\right], \\
&= \frac{KC}{K+C}\left[1 + \mathcal{O}\left(\frac{1}{K+C}\right)\right].
\end{aligned}$$

Multiplying both K and C by a gives

$$\begin{aligned}
\sum_{j=1}^{C} N(j,t) &= \frac{a^2KC}{aK + aC}\left[1 + \mathcal{O}\left(\frac{1}{aK + aC}\right)\right], \\
&= a\frac{KC}{K+C}\left[1 + \mathcal{O}\left(\frac{1}{K+C}\right)\right].
\end{aligned}$$

For P_{new} it has been shown that

$$\begin{aligned}
P_{new} &= \frac{C-1}{K+C-1}, \\
&= \frac{C}{K+C}\left[\frac{1}{1-\frac{1}{K+C}}\right] - \frac{1}{K+C}\left[\frac{1}{1-\frac{1}{K+C}}\right], \\
&= \frac{C}{K+C}\left[1+\mathcal{O}\left(\frac{1}{K+C}\right)\right] - \frac{1}{K+C}\left[1+\mathcal{O}\left(\frac{1}{K+C}\right)\right].
\end{aligned}$$

Multiplying both K and C by a gives

$$\begin{aligned}
P_{new} &= \frac{aC}{aK+aC}\left[1+\mathcal{O}\left(\frac{1}{aK+aC}\right)\right] - \frac{1}{aK+aC}\left[1+\mathcal{O}\left(\frac{1}{aK+aC}\right)\right], \\
&= \frac{C}{K+C}\left[1+\mathcal{O}\left(\frac{1}{K+C}\right)\right] - \frac{1}{a}\frac{1}{K+C}\left[1+\mathcal{O}\left(\frac{1}{K+C}\right)\right].
\end{aligned}$$

The statement follows. □

Corollary 4.2.4 (Expected number of colors in stationary when $C = K$ with equally likely colors.). *When there are K positions in the online structure, the total number of colors is K and the probability for each color to be drawn from the source is $\frac{1}{K}$ it follows that $\mathbb{E}\left[N(i,t)\right]$ approximates $\frac{K+1}{2^{i+1}}$, the number of colors in the online structure approximates $\frac{K}{2}+\frac{1}{4}$ and P_{new} approximates $\frac{1}{2}$.*

Proof. Using the obtained relations from the theorem and applying $C = K$ yields:

$$\begin{aligned}
\nu(i,t) &= \frac{KC^2(K-1)^{i-1}}{(K+C-1)^{i+1}}, \\
&= \frac{K^3(K-1)^{i-1}}{(2K-1)^{i+1}}, \\
&= K\left(\frac{K}{2K-1}\right)^2\left(\frac{K-1}{2K-1}\right)^{i-1}, \\
&= K\left(\frac{K}{2K}\left[\frac{1}{1-\frac{1}{2K}}\right]\right)^2\left(\frac{K-1}{2(K-1)}\left[\frac{1}{1+\frac{1}{2(K-1)}}\right]\right)^{i-1}, \\
&= K\left(\frac{1}{2}\left[1+\frac{1}{2K}+\mathcal{O}\left(\frac{1}{K^2}\right)\right]\right)^2\left(\frac{1}{2}\left[1-\mathcal{O}\left(\frac{1}{K}\right)\right]\right)^{i-1}, \\
&= \frac{K+1}{2^{i+1}}+\mathcal{O}\left(\frac{1}{K}\right), \\
\sum_{j=1}^{K}\nu(j,t) &= \frac{KC}{K+C-1}, \\
&= \frac{K^2}{2K-1}, \\
&= \frac{K^2}{2K}\left[\frac{1}{1-\frac{1}{2K}}\right], \\
&= \frac{K}{2}\left[1+\frac{1}{2K}+\mathcal{O}\left(\frac{1}{K^2}\right)\right], \\
&= \frac{K}{2}+\frac{1}{4}+\mathcal{O}\left(\frac{1}{K}\right),
\end{aligned}$$

$$\begin{aligned} P_{new} &= \frac{C-1}{K+C}, \\ &= \frac{K-1}{2K}, \\ &= \frac{1}{2} - \frac{1}{2K}. \end{aligned}$$

□

4.3 Heterogeneous Colors

In order to make a generatlization of the dynamics, let $M(i,t)$ be the set of the color numbers from which the queues have i balls at time t, so $M(i,t)(q(t)) = \{c : q_c(t) = i\}$. For ease of notation we write $M(i,t)$ instead of $M(i,t)(q(t))$. When at time t there are $N(i,t)$ queues of length i, $t \geq K$, the possibilities for $N(i,t+1)$ are the same as for the uniform distribution and therefore are not repeated. Moreover, the same special cases are introduced, i.e., $i = 0$, $i = 1$, and $i = K$. In the following subsection only the recurrence relations are obtained, proofs on the number of colors in the online structure are left for future research.

4.3.1 Recurrence Relations

Before we provide the equations, we note the following. Suppose that a ball is removed from a queue with length i. This means that a color is removed from $M(i,t)$. The probability for this to happen to a color $k \in M(i,t)$ is the same for every color in $M(i,t)$. The probability that after a color is removed from $M(i,t)$ also a color is added to $M(i,t)$ is:

$$\begin{aligned} \sum_{k \in M(i,t)} \frac{1}{\sum_{j=1}^{K} N(j,t)} \left(\sum_{j \in M(i,t)} p_j - p_k \right) &= \frac{N(i,t)}{\sum_{j=1}^{K} N(j,t)} \sum_{j \in M(i,t)} p_j - \frac{\sum_{j \in M(i,t)} p_j}{\sum_{j=1}^{K} N(j,t)}, \\ &= \frac{N(i,t)-1}{\sum_{j=1}^{K} N(j,t)} \sum_{j \in M(i,t)} p_j. \end{aligned}$$

The same can happen when a ball is removed from a queue with length $i+1$. Then a color is added to $M(i,t)$ by removing a ball from $N(i+1,t)$. If afterwards a ball is added to a queue with length i, a color is added to $M(i,t)$. The probability for this to happen is:

$$\sum_{k \in M(i+1,t)} \frac{1}{\sum_{j=1}^{K} N(j,t)} \left(\sum_{j \in M(i,t)} p_j + p_k \right) = \frac{N(i+1,t)}{\sum_{j=1}^{K} N(j,t)} \sum_{j \in M(i,t)} p_j + \frac{\sum_{j \in M(i+1,t)} p_j}{\sum_{j=1}^{K} N(j,t)}.$$

The last special case that is discussed is the following. Suppose that a ball is removed from a queue with length $i+1$. This means that a color is removed from $M(i+1,t)$ and a color is added to $M(i,t)$. Suppose that afterwards a ball is drawn from a queue that does not have length i nor length $i+1$. Then although $M(i+1,t)$ and $M(i,t)$ have changed, the colors that are in both of them did not change. So the probability for this to happen is

$$\frac{N(i+1,t)}{\sum_{j=1}^{K} N(j,t)} \left(1 - \sum_{j \in M(i+1,t)} p_j - \sum_{j \in M(i,t)} p_j \right).$$

The dynamics are given now:

$$N(0,t+1) = \begin{cases} N(0,t)-1, & \text{w.p.} \quad \left(1-\frac{N(1,t)}{\sum_{j=1}^{K} N(j,t)}\right) \sum_{j\in M(0,t)} p_j, \\ N(0,t), & \text{w.p.} \quad \left(1-\frac{N(1,t)}{\sum_{j=1}^{K} N(j,t)}\right) \left(1-\sum_{j\in M(0,t)} p_j\right) + \\ & \quad \frac{N(1,t)}{\sum_{j=1}^{K} N(j,t)} \sum_{j\in M(0,t)} p_j + \frac{\sum_{j\in M(1,t)} p_j}{\sum_{j=1}^{K} N(j,t)}, \\ N(0,t)+1, & \text{w.p.} \quad \frac{N(1,t)}{\sum_{j=1}^{K} N(j,t)} \left(1-\sum_{j\in M(0,t)} p_j\right) - \frac{\sum_{j\in M(1,t)} p_j}{\sum_{j=1}^{K} N(j,t)}, \end{cases}$$

$$N(1,t+1) = \begin{cases} N(1,t)-2, & \text{w.p.} \quad \frac{N(1,t)-1}{\sum_{j=1}^{K} N(j,t)} \sum_{j\in M(1,t)} p_j, \\ N(1,t)-1, & \text{w.p.} \quad \frac{N(1,t)}{\sum_{j=1}^{K} N(j,t)} \left(1-\sum_{j\in M(0,t)\vee M(1,t)} p_j\right) + \\ & \quad \left(1-\frac{N(1,t)+N(2,t)}{\sum_{j=1}^{K} N(j,t)}\right) \sum_{j\in M(1,t)} p_j, \\ N(1,t), & \text{w.p.} \quad \frac{N(1,t)}{\sum_{j=1}^{K} N(j,t)} \sum_{j\in M(0,t)} p_j + \frac{\sum_{j\in M(1,t)} p_j}{\sum_{j=1}^{K} N(j,t)} + \\ & \quad \left(1-\frac{N(1,t)+N(2,t)}{\sum_{j=1}^{K} N(j,t)}\right) \left(1-\sum_{j\in M(0,t)\vee M(1,t)} p_j\right) + \\ & \quad \frac{N(2,t)}{\sum_{j=1}^{K} N(j,t)} \sum_{j\in M(1,t)} p_j + \frac{\sum_{j\in M(2,t)} p_j}{\sum_{j=1}^{K} N(j,t)}, \\ N(1,t)+1, & \text{w.p.} \quad \left(1-\frac{N(1,t)+N(2,t)}{\sum_{j=1}^{K} N(j,t)}\right) \sum_{j\in M(0,t)} p_j + \\ & \quad \frac{N(2,t)}{\sum_{j=1}^{K} N(j,t)} \left(1-\sum_{j\in M(0,t)\vee M(1,t)} p_j\right) - \frac{\sum_{j\in M(2,t)} p_j}{\sum_{j=1}^{K} N(j,t)}, \\ N(1,t)+2, & \text{w.p.} \quad \frac{N(2,t)}{\sum_{j=1}^{K} N(j,t)} \sum_{j\in M(0,t)} p_j, \end{cases}$$

$$N(i,t+1) = \begin{cases} N(i,t)-2, & \text{w.p.} \quad \frac{N(i,t)-1}{\sum_{j=1}^{K} N(j,t)} \sum_{j\in M(i,t)} p_j, \\ N(i,t)-1, & \text{w.p.} \quad \frac{N(i,t)}{\sum_{j=1}^{K} N(j,t)} \left(1-\sum_{j\in M(i,t)\vee M(i-1,t)} p_j\right) + \\ & \quad \left(1-\frac{N(i,t)+N(i+1,t)}{\sum_{j=1}^{K} N(j,t)}\right) \sum_{j\in M(i,t)} p_j, \\ N(i,t), & \text{w.p.} \quad \frac{N(i,t)}{\sum_{j=1}^{K} N(j,t)} \sum_{j\in M(i-1,t)} p_j + \frac{\sum_{j\in M(i,t)} p_j}{\sum_{j=1}^{K} N(j,t)} \\ & \quad \left(1-\frac{N(i-1,t)+N(i,t)+N(i+1,t)}{\sum_{j=1}^{K} N(j,t)}\right) \left(1-\sum_{j\in M(i,t)\vee M(i-1,t)} p_j\right) + \\ & \quad \frac{N(i-1,t)}{\sum_{j=1}^{K} N(j,t)} \left(1-\sum_{j\in M(i,t)\vee M(i-1,t)} p_j\right) + \frac{\sum_{j\in M(i-1,t)} p_j}{\sum_{j=1}^{K} N(j,t)} + \\ & \quad \frac{N(i+1,t)}{\sum_{j=1}^{K} N(j,t)} \sum_{j\in M(i,t)} p_j + \frac{\sum_{j\in M(i+1,t)} p_j}{\sum_{j=1}^{K} N(j,t)}, \\ N(i,t)+1, & \text{w.p.} \quad \left(1-\frac{N(i-1,t)+N(i,t)+N(i+1,t)}{\sum_{j=1}^{K} N(j,t)}\right) \sum_{j\in M(i-1,t)} p_j + \\ & \quad \frac{N(i-1,t)-1}{\sum_{j=1}^{K} N(j,t)} \sum_{j\in M(i-1,t)} p_j + \frac{N(i+1,t)}{\sum_{j=1}^{K} N(j,t)} \left(1-\sum_{j\in M(i-1,t)\vee M(i,t)} p_j\right) - \\ & \quad \frac{\sum_{j\in M(i+1)} p_j}{\sum_{j=1}^{K} N(j,t)}, \\ N(i,t)+2, & \text{w.p.} \quad \frac{N(i+1,t)}{\sum_{j=1}^{K} N(j,t)} \sum_{j\in M(i-1,t)} p_j, \end{cases}$$

$$N(K,t+1) \quad = \quad \begin{cases} N(K,t)-1, & \text{w.p.} \quad \frac{N(K,t)}{\sum_{j=1}^{K} N(j,t)} \left(1-\sum_{j\in M(K-1,t)} p_j\right) - \frac{\sum_{j\in M(K,t)} p_j}{\sum_{j=1}^{K} N(j,t)}, \\ N(K,t), & \text{w.p.} \quad \frac{N(K,t)}{\sum_{j=1}^{K} N(j,t)} \sum_{j\in M(K-1,t)} p_j + \frac{\sum_{j\in M(K,t)} p_j}{\sum_{j=1}^{K} N(j,t)} + \\ & \qquad \left(1-\frac{N(K-1,t)+N(K,t)}{\sum_{j=1}^{K} N(j,t)}\right)\left(1-\sum_{j\in M(K-1,t)} p_j\right) + \\ & \qquad \frac{N(K-1,t)}{\sum_{j=1}^{K} N(j,t)} \left(1-\sum_{j\in M(K-1,t)} p_j\right) + \frac{\sum_{j\in M(K-1,t)} p_j}{\sum_{j=1}^{K} N(j,t)}, \\ N(K,t)+1, & \text{w.p.} \quad \left(1-\frac{N(K-1,t)+N(K,t)}{\sum_{j=1}^{K} N(j,t)}\right) \sum_{j\in M(K-1,t)} p_j + \frac{N(K-1,t)-1}{\sum_{j=1}^{K} N(j,t)} \sum_{j\in M(K-1,t)} p_j. \end{cases}$$

The dynamics result in the following expectations for the number of queues with length i:

$$\begin{aligned} \mathbb{E}\left[N(0,t+1)\right] &= \mathbb{E}\left[N(0,t)\right] + \mathbb{E}\left[\frac{N(1,t)}{\sum_{j=1}^{K} N(j,t)}\right] - \sum_{j\in M(0,t)} p_j - \mathbb{E}\left[\frac{\sum_{j\in M(1,t)} p_j}{\sum_{j=1}^{K} N(j,t)}\right], \\ \mathbb{E}\left[N(1,t+1)\right] &= \mathbb{E}\left[N(1,t)\right] + \sum_{j\in M(0,t)} p_j - \sum_{j\in M(1,t)} p_j - \mathbb{E}\left[\frac{\sum_{j\in M(2,t)} p_j}{\sum_{j=1}^{K} N(j,t)}\right] + \mathbb{E}\left[\frac{2\sum_{j\in M(1,t)} p_j}{\sum_{j=1}^{K} N(j,t)}\right] + \\ &\quad \mathbb{E}\left[\frac{N(2,t)}{\sum_{j=1}^{K} N(j,t)}\right] - \mathbb{E}\left[\frac{N(1,t)}{\sum_{j=1}^{K} N(j,t)}\right], \\ \mathbb{E}\left[N(i,t+1)\right] &= \mathbb{E}\left[N(i,t)\right] + \sum_{j\in M(i-1,t)} p_j - \sum_{j\in M(i,t)} p_j - \mathbb{E}\left[\frac{\sum_{j\in M(i+1,t)} p_j}{\sum_{j=1}^{K} N(j,t)}\right] + \mathbb{E}\left[\frac{2\sum_{j\in M(i,t)} p_j}{\sum_{j=1}^{K} N(j,t)}\right] - \\ &\quad \mathbb{E}\left[\frac{\sum_{j\in M(i-1,t)} p_j}{\sum_{j=1}^{K} N(j,t)}\right] + \mathbb{E}\left[\frac{N(i+1,t)}{\sum_{j=1}^{K} N(j,t)}\right] - \mathbb{E}\left[\frac{N(i,t)}{\sum_{j=1}^{K} N(j,t)}\right], \\ \mathbb{E}\left[N(K,t+1)\right] &= \mathbb{E}\left[N(K,t)\right] + \sum_{j\in M(K-1,t)} p_j + \mathbb{E}\left[\frac{\sum_{j\in M(K,t)} p_j}{\sum_{j=1}^{K} N(j,t)}\right] - \mathbb{E}\left[\frac{\sum_{j\in M(K-1,t)} p_j}{\sum_{j=1}^{K} N(j,t)}\right] - \mathbb{E}\left[\frac{N(K,t)}{\sum_{j=1}^{K} N(j,t)}\right]. \end{aligned}$$

In stationarity $\mathbb{E}\left[N(i,t+1)\right] = \mathbb{E}\left[N(i,t)\right]$ for $i = 0,...,K$. However, these equations can not be solved and a solution to this problem is left for future research. When the probabilities for each color to be drawn from the source correspond to a uniform distribution, the same equations are obtained as in Theorem 4.2.1. It is expected that, as was the case after K steps, it can be proven that the expected number of colors in the online structure in stationarity is maximized when the probabilities for each color to be drawn from the source correspond to a uniform distribution. Since there is no expression yet for the number of colors in the online structure for a general distribution, this is also left for future research.

4.3.2 The Role of Super Colors

Super colors are colors with a high probability to be drawn from the source, especially compared to the other colors. It turns out that the super colors dominate the online structure, see the results in Section 5.3 and Section 5.5, Figures 5.10, 5.11b, 5.16, 5.17, 5.19, 5.20 and 5.21. As a result of the super colors, the number of colors in the online structure stays low in stationarity. This is what the designers of the web crawler observe. The rate at which balls are removed from the online structure is $\mathbb{E}\left[\frac{1}{\sum_{j=1}^{K} N(j,t)}\right]$ and let $p^* = \max_i p_i$. When $p^* > \mathbb{E}\left[\frac{1}{\sum_{j=1}^{K} N(j,t)}\right] > \frac{1}{K}$, the super color dominates the online structure, since they are added more often to the online structure than they are removed from the online structure. It is expected that when $p^* < \mathbb{E}\left[\frac{1}{\sum_{j=1}^{K} N(j,t)}\right]$, the super color does not dominate the online structure anymore, because then the rate out of the online structure is greater than the rate into the online structure.

Chapter 5

Numerical Results: no Offline Structure

The results in this chapter are shown to support the analytical results that have been derived in Chapter 3 and Chapter 4. Moreover, the results are used as a reference point to analyze the strategies for the use of the offline structure.

In this section we provide numerical results for the case there is a finite number of colors C and that there are K positions in the online structure. The model has been presented in Section 4.1. Several distributions are used to invoke the frequency of colors to be drawn from the source. Note that in practice the frequency of colors corresponds to the size of the hosts. The results shown are for the uniform, exponential, normal, Poisson and power law distribution.

All the results presented in this thesis are unique due to the fact that a single simulation is performed. However, if the same simulation would be repeated the stationary behavior is equal.

5.1 Uniform Distribution

In this section the assumption is made that the probabilities for a color to be drawn from the source correspond to a uniform distribution, that is, $p_i = \frac{1}{C}\ \forall i = 1, ..., C$. For the uniform distribution a distinction is made between a total number of K colors and a total number of C colors. This is done since analytical results have been obtained for both cases in Section 4.2.

5.1.1 K Colors

Consider the case that there are K colors in the game. Figure 5.1 shows a progression of the number of colors in the online structure.

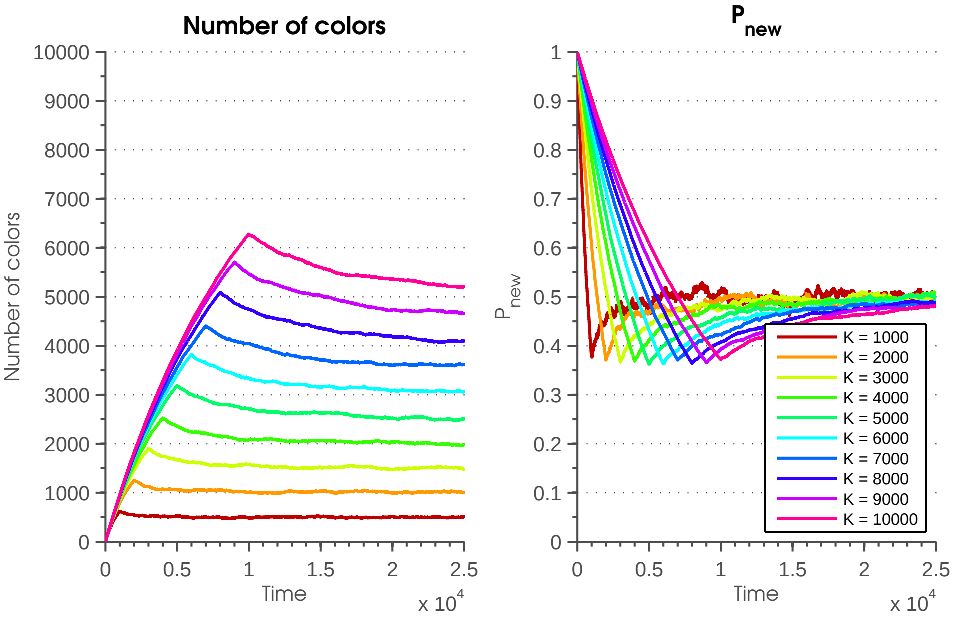


Figure 5.1: *The number of colors in the online structure (left) and the probability to draw a ball from the source that has a color that is not present in the online structure, P_{new}, (right) as a function of the time. The probability to draw a color from the source corresponds to a uniform distribution.*

At time $t = 0$ the number of colors in the online structure is zero and accordingly, the probability to draw a ball from the source with a color that is not present in the online structure equals 1. After K time steps the online structure is completely filled and it is expected that there are approximately $0.6321K$ different colors in the online structure, see (3.2). This is what the numerical result shows as well. Every time step after $t = K$ one ball is removed from the online structure and one ball is added to the online structure.

The result in Figure 5.1 also corresponds with the other analytical results that have been derived in Section 4.2. These analytical results are that P_{new} approximates $\frac{1}{2}$ and that the number of colors in the online structure in stationarity approximates $\frac{K}{2}$. The following figure shows the plots for $N(i, t)$, the number of queues with length i at time t, $i = 0, 1, 2, 3, 4, 5$, when $K = 10000$. Again, these results are in accordance with the analytical result: $\mathbb{E}\left[N(i, t)\right]$ approximates $\frac{K}{2^{i+1}}$, see Corrolary 4.2.4.

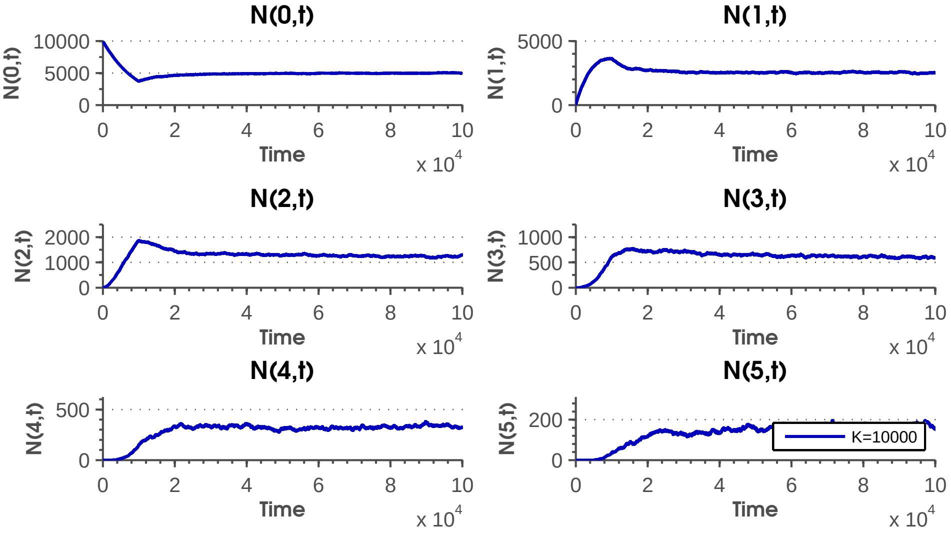


Figure 5.2: The number of queues in the online structure with length i, $i = 0, 1, 2, 3, 4, 5$ as a function of the time t. $C = K$ and the probability to draw a color from the source corresponds to a uniform distribution.

Note that $N(0,t)$ is decreasing in the beginning, this is caused by the fact that all the positions in the online structure are empty at $t = 0$. For the same reason in the plots for $N(i,t)$, $i = 1, ..., 5$ the number of colors in the queue with length i is increasing.

5.1.2 C Colors

Also in case there is a total number of C colors in the game, it turns out that the numerical results indeed correspond to the obtained analytical results.

Variation of the total number of colors

Figure 5.3 presents the numerical results using a fixed value of K and different values for C.

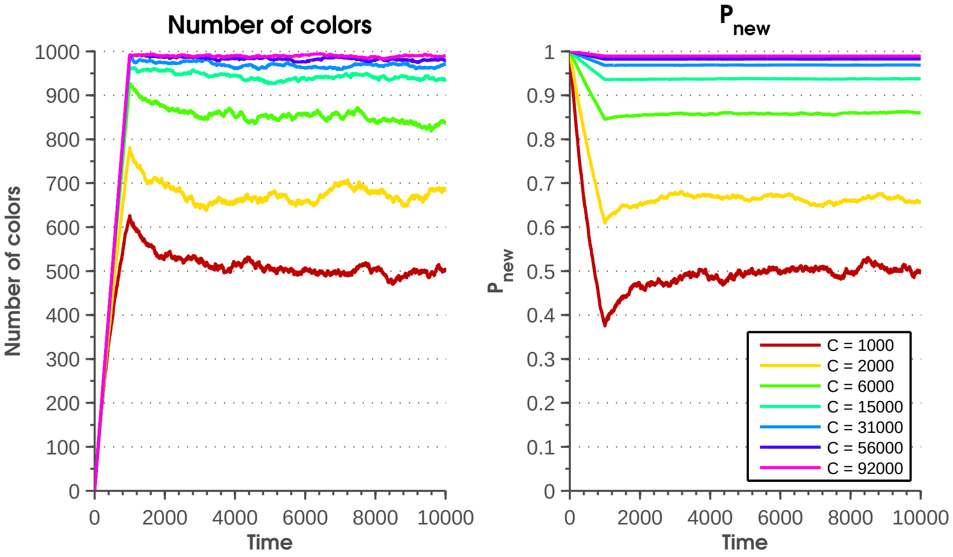


Figure 5.3: The number of colors in the online structure (left) and the probability to draw a ball from the source that has a color that is not present in the online structure, P_{new}, (right) as a function of the time for different values of C. $K = 1000$ and the probability to draw a color from the source corresponds to a uniform distribution.

The figure shows that if C increases, while K remains the same, P_{new} tends to 1. This is in accordance with the analytical results that are obtained in Section 4.2, as both P_{new} and the number of colors in the online structure are increasing in C. Furthermore it has been shown in Section 4.2 that if C tends to infinity that P_{new} tends to 1 and the number of colors in the online structure tends to K. Note that when $C = K$ the results are the same as for the case that there were K colors in the game, see Figure 5.1. The expected number of colors at $t = K$ also corresponds with the analytical result and as C tends to infinity, the expected number of colors at time K goes to K.

Variation of the number of positions in the online structure

Varying the number of positions K, when the probability vector p corresponds to a uniform distribution, is presented in Figure 5.4. Also this figure is in accordance to the analytical resuls: if $C >> K$ the probability to draw a ball from the source with a color that is not present in the online structure tends to 1.

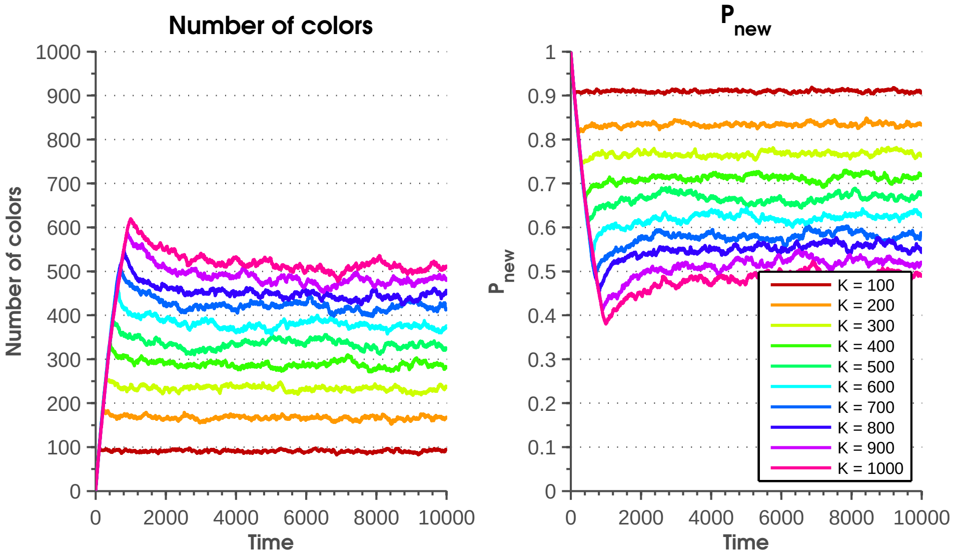


Figure 5.4: *The number of colors in the online structure (left) and the probability to draw a ball from the source that has a color that is not present in the online structure, P_{new}, (right) as a function of the time for different values of K. $C = 1000$ and the probability to draw a color from the source corresponds to a uniform distribution.*

The analytical results say that when, for example, $K = 100$ and $C = 1000$ the number of colors in the online structure is expected to be 91 in stationarity and P_{new} approximates 91%. With these parameters the expected number of colors in the online structure at time K is 95. Figure 5.4 suggests that this is indeed the case.

5.2 Exponential Distribution

In this subsection the probability vector p corresponds to an exponential distribution with mean λ. Since it is a discrete version of the exponential distribution this comes down to a geometric distribution with success probability $1 - e^{-\frac{1}{\lambda}}$.

Variation of the total number of colors

Figure 5.5 shows variation of the total number of colors C for a fixed value of K, the number of positions in the online structure, and a fixed value of the mean λ.

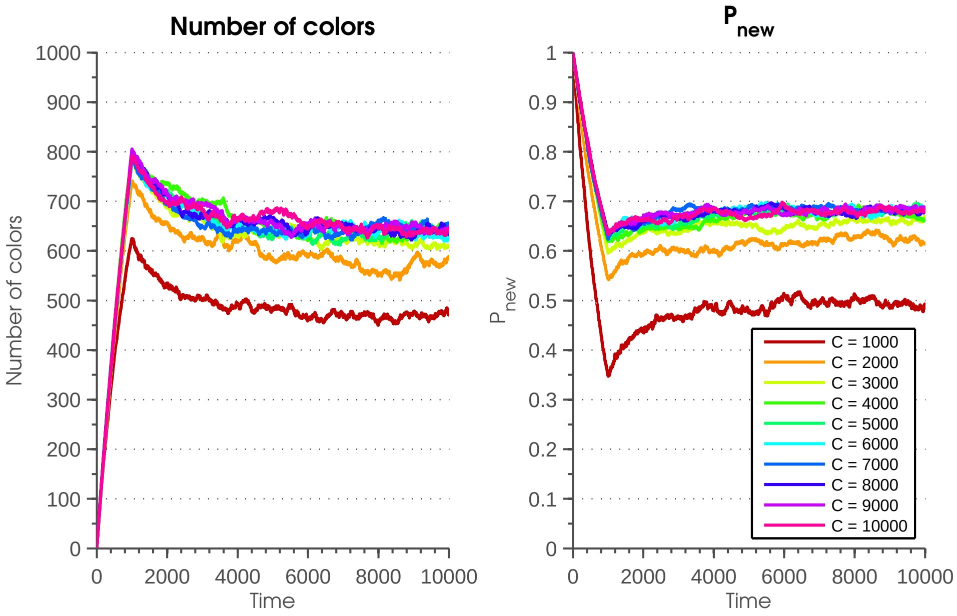


Figure 5.5: *The number of colors in the online structure (left) and the probability to draw a ball from the source that has a color that is not present in the online structure, P_{new}, (right) as a function of the time for different values of C. $K = 1000$ and the probability to draw a color from the source corresponds to an exponential distribution with mean $\lambda = 1000$.*

In case that $C = \lambda$, Figure 5.5 shows that P_{new} is about $\frac{1}{2}$ and that the number of colors in the online structure is approximately $\frac{K}{2}$. This was also the case for the uniform distribution if $C = K$. The cumulative distribution function for the uniform distribution is linear in C. The cumulative distribution function for the exponential distribution is:

$$F(c, \lambda) = \frac{1 - e^{-\frac{c}{\lambda}}}{1 - e^{-\frac{C}{\lambda}}},$$

where $c = 1, ..., C$ is the color. The normalization is applied because the sum of the elements in the probability vector p should be 1. It turns out that in case $C = \lambda$ the uniform distribution is a good approximation for the exponential distribution. This is explained below. Note that when $C = \lambda$, the cumulative distribution function is

$$F(c, \lambda) = \frac{1 - e^{-\frac{c}{\lambda}}}{1 - e^{-1}},$$

Now let $g(c)$ be the difference between $F(c, \lambda)$ and $\frac{c}{\lambda}$, the cumulative distribution function of the uniform distribution. So,

$$\begin{aligned} g(c) &= \frac{1 - e^{-\frac{c}{\lambda}}}{1 - e^{-1}} - \frac{c}{\lambda}, \\ g'(c) &= \frac{1}{\lambda}\left[\frac{e^{-\frac{c}{\lambda}}}{1 - e^{-1}} - 1\right]. \end{aligned}$$

Note that the function $g(c)$ is a concave function. The derivative is a decreasing function, because $e^{-\frac{c}{\lambda}}$ is a decreasing function and $1 - e^{-1}$ is just a constant. At $c = 0$, $g'(c) = \frac{1}{e-1}\frac{1}{\lambda} > 0$ and at $c = C$, $g'(c) = \frac{2-e}{e-1}\frac{1}{\lambda} < 0$ and it follows that $g(c)$ is a concave function and the extreme value is a maximum. Solving the equation $g'(c) = 0$ yields:

$$\begin{aligned} \frac{1}{\lambda}\left[\frac{e^{-\frac{c}{\lambda}}}{1 - e^{-1}} - 1\right] &= 0, \\ e^{-\frac{c}{\lambda}} &= 1 - e^{-1}, \\ c &= -\lambda \ln(1 - e^{-1}), \end{aligned}$$

and it follows that the top of the concave function is at $c = -\lambda \ln(1-e^{-1})$. The function value at this point is $\frac{1}{e-1} + \ln(1-e^{-1}) = 0.12$. This means that the difference between the uniform cumulative distribution function and the exponential cumulative distribution function is at most 0.12 and this could explain why the exponential distribution for p in case $C = \lambda$ gives similar results as the uniform distribution for p.

In the case $C \neq \lambda$ and when C goes to infinity, the cumulative distribution function reduces to

$$F(c, \lambda) = 1 - e^{-\frac{c}{\lambda}}.$$

Because of the discrete exponential distribution the probability that a ball has color c , where $c = 1, ..., C$, is the following:

$$\begin{aligned} P(color = c) &= e^{-\frac{1}{\lambda}(c-1)} - e^{-\frac{1}{\lambda}c}, \\ &= e^{-\frac{1}{\lambda}(c-1)}(1 - e^{-\frac{1}{\lambda}}). \end{aligned} \tag{5.1}$$

Note that there will be convergence of the number of colors in the online structure from a certain point onwards when C goes to ∞, because the colors at the tail of the distribution are very unlikely to retrieve from the source. Figure 5.5 indeed illustrates this behavior and it means that P_{new} is independent of the total number of colors C. The convergence can also be seen in the expected number of colors at time K. With a constant mean $\lambda = 1000$ and with $K = 1000$ the expected number of colors at time K converges to approximately 796.

Variation of the number of positions in the online structure

Figure 5.6 shows what happens when the number of positions in the online structure, K, is changing, while the total number of colors C and the mean of the exponential distribution λ are fixed.

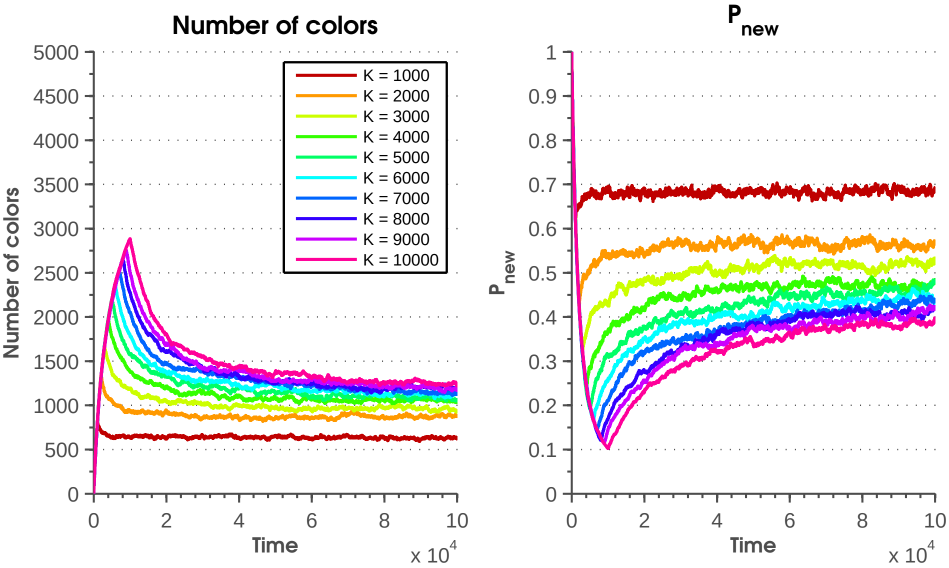


Figure 5.6: *The number of colors in the online structure (left) and the probability to draw a ball from the source that has a color that is not present in the online structure, P_{new}, (right) as a function of the time for different values of K. $C = 100000$ and the probability to draw a color from the source corresponds to an exponentail distribution with mean $\lambda = 1000$.*

If K changes, the probability to draw a color from the source remains the same, as well as the number of elements in the probability vector p that are almost zero. The difference is the number of positions in the online structure to fill. This explains the convergence from a certain value of K onwards. The behavior at $t = K$ is explained by (3.1).

Variation of the mean λ

The last figure for the probability vector that corresponds to an exponential distribution shows a variation of λ, the mean of the exponential distribution.

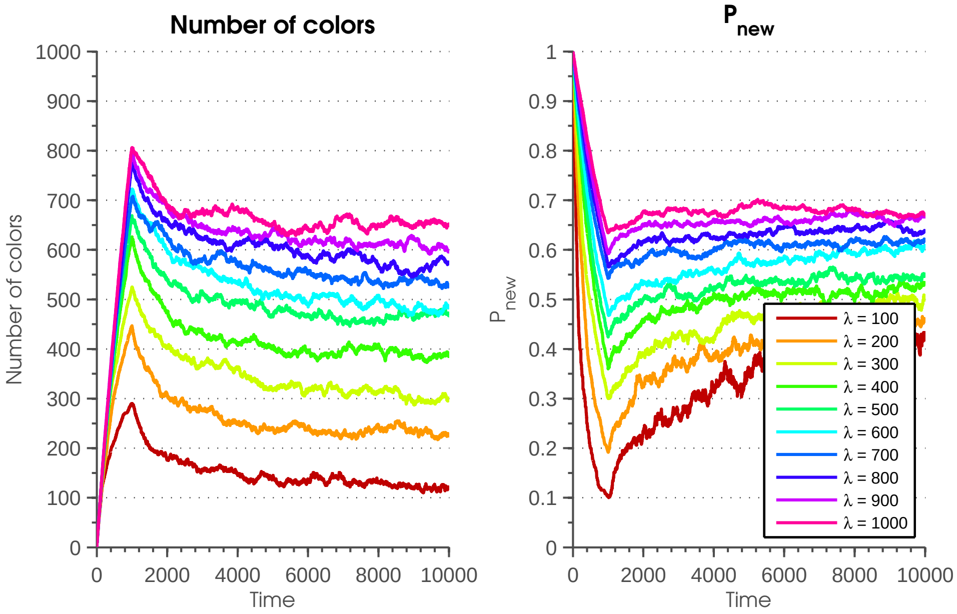


Figure 5.7: *The number of colors in the online structure (left) and the probability to draw a ball from the source that has a color that is not present in the online structure, P_{new}, (right) as a function of the time for different values of λ. $C = 100000$, $K = 1000$ and the probability to draw a color from the source corresponds to an exponential distribution with mean λ.*

When λ increases, the figures shows that the number of colors in the online structure increases, as well as P_{new}. Remember the formula of the probability that a ball is of color c, see (5.1). When C is a constant and λ is a variable then this is an increasing function, as the derivative shows:

$$\frac{dP(color = c)}{d\lambda} = e^{-\frac{c}{\lambda}} \left[c(e^{\frac{1}{\lambda}} - 1) - e^{\frac{1}{\lambda}} \right].$$

So when λ increases, the probability to draw of a ball with color c from the source increases. Note that normalization applies in order to keep a cumulative distribution function. Therefore, there are more colors in the online structure since the number of colors with probability of (almost) zero to be drawn from the source decreases.

5.3 Normal Distribution

In this subsection, the probability vector p has a normal distribution with mean μ and standard deviation σ.

Variation of the total number of colors

In Figure 5.8 the total number of colors C is changing, while the number of positions in the online structure, K, and both the mean μ and the standard deviation σ of the normal distribution are fixed.

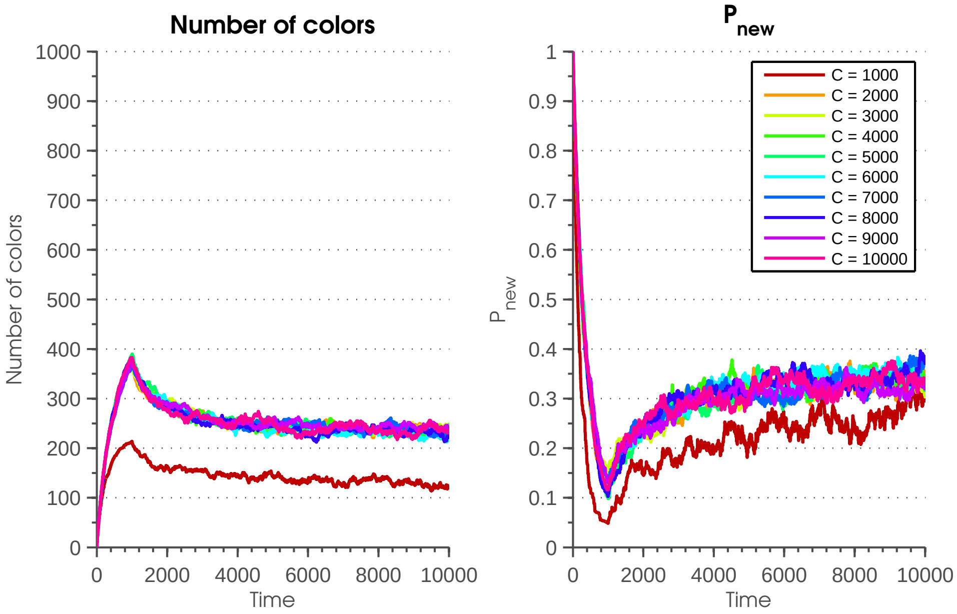


Figure 5.8: *The number of colors in the online structure (left) and the probability to draw a ball from the source that has a color that is not present in the online structure, P_{new}, (right) as a function of the time for different values of C. $K = 1000$ and the probability to draw a color from the source corresponds to a normal distribution with $\mu = 1000$ and $\sigma = 100$.*

First of all, note that Figure 5.8 shows that the behavior of the number of colors in the online structure in stationarity is very different from the case the probability vector p corresponds to a uniform distribution. Figure 5.8 also shows that the number of colors in the online structure and P_{new} are independent of the total number of colors. This is because the probability vector p corresponds to a normal distribution and it is known from the normal distribution that 99.7% of the values are within three standard deviations from the mean. Therefore there are about 6σ elements in this vector that are non-zero. So, only when $C \neq \mu$, the probability vectors p are just shifted versions of each other. When $C = \mu$ the mean lies at the edge of the range of the values, and therefore in this case there are only 3σ elements in the probability vector that are non-zero. This explains why the number of colors in the online structure is now about half the number of colors from the case $C \neq \mu$. It also explains why the number of colors in the online structure after K steps is approximately the same for all values of C, $C \neq \mu$.

Variation of the number of positions in the online structure

The next variable that changes is K, the number of positions in the online structure. The other variables, C, the total number of colors, μ, the mean of the normal distribution, and σ, the standard deviation of the normal distribution, are fixed.

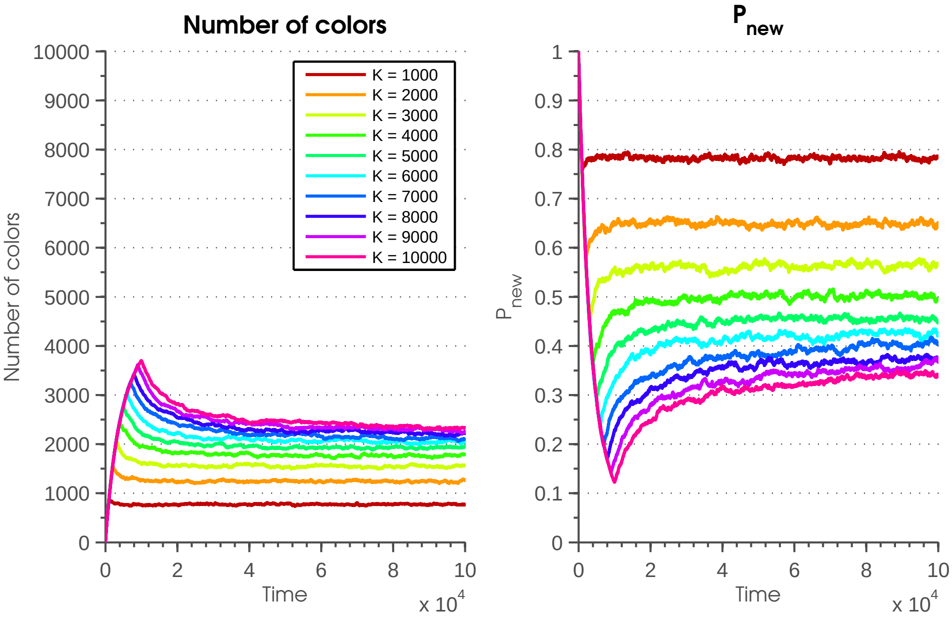


__Figure 5.9:__ The number of colors in the online structure (left) and the probability to draw a ball from the source that has a color that is not present in the online structure, P_{new}, (right) as a function of the time for different values of K. $C = 100000$ and the probability to draw a color from the source corresponds to a normal distribution with mean $\mu = 10000$ and standard deviation $\sigma = 1000$.

When K is increasing towards C, the probability of drawing a ball that has a color that is not present in the online structure is converging to the same value when $K \geq 6\sigma$. Since there are about 6σ different colors, for the other colors the probability is about 0, this results in the same number of colors in the online structure when K is increasing. Note that the number of colors in stationarity is smaller than 6σ.The number of colors in the online structure at time $t = K$ is also converging to the same value when $K \geq 6\sigma$, the reasoning is similar.

In case $K < 6\sigma$, there are more possible colors than number of positions in the online structure and therefore the number of colors in the online structure is lower than when $K \geq 6\sigma$.

Variation of the mean μ

Now the variable μ, the mean of the normal distribution, varies, while the other variables C, the total number of colors, K, the number of positions in the online structure, and σ, the standard deviation of the normal distribution, are fixed.

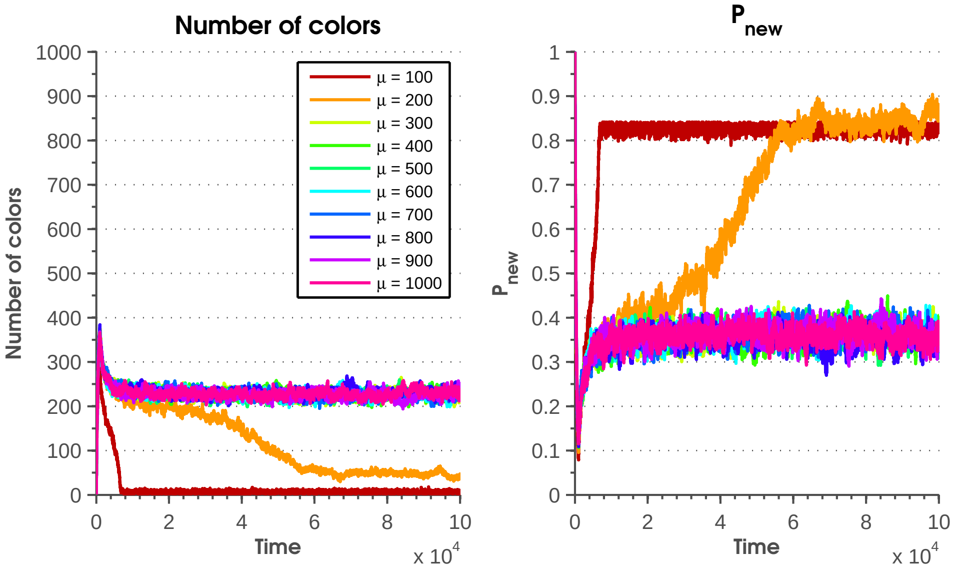


Figure 5.10: *The number of colors in the online structure (left) and the probability to draw a ball from the source that has a color that is not present in the online structuree, P_{new}, (right) as a function of the time for different values of μ. $C = 100000$, $K = 1000$ and the probability to draw a color from the source corresponds to a normal distribution with mean μ and standard deviation $\sigma = 100$.*

Figure 5.10 illustrates that the number of colors in the online structure is independent of the mean of the distribution, except for the case when $\mu = \sigma$ and $\mu = 2\sigma$. This is again due to the fact that the non-zero elements of the probability vector p are in an interval of approximately 6σ. Hence when $\mu \geq 3\sigma$, the number of colors in the online structure is independent of mean of the distribution, as well is P_{new}.

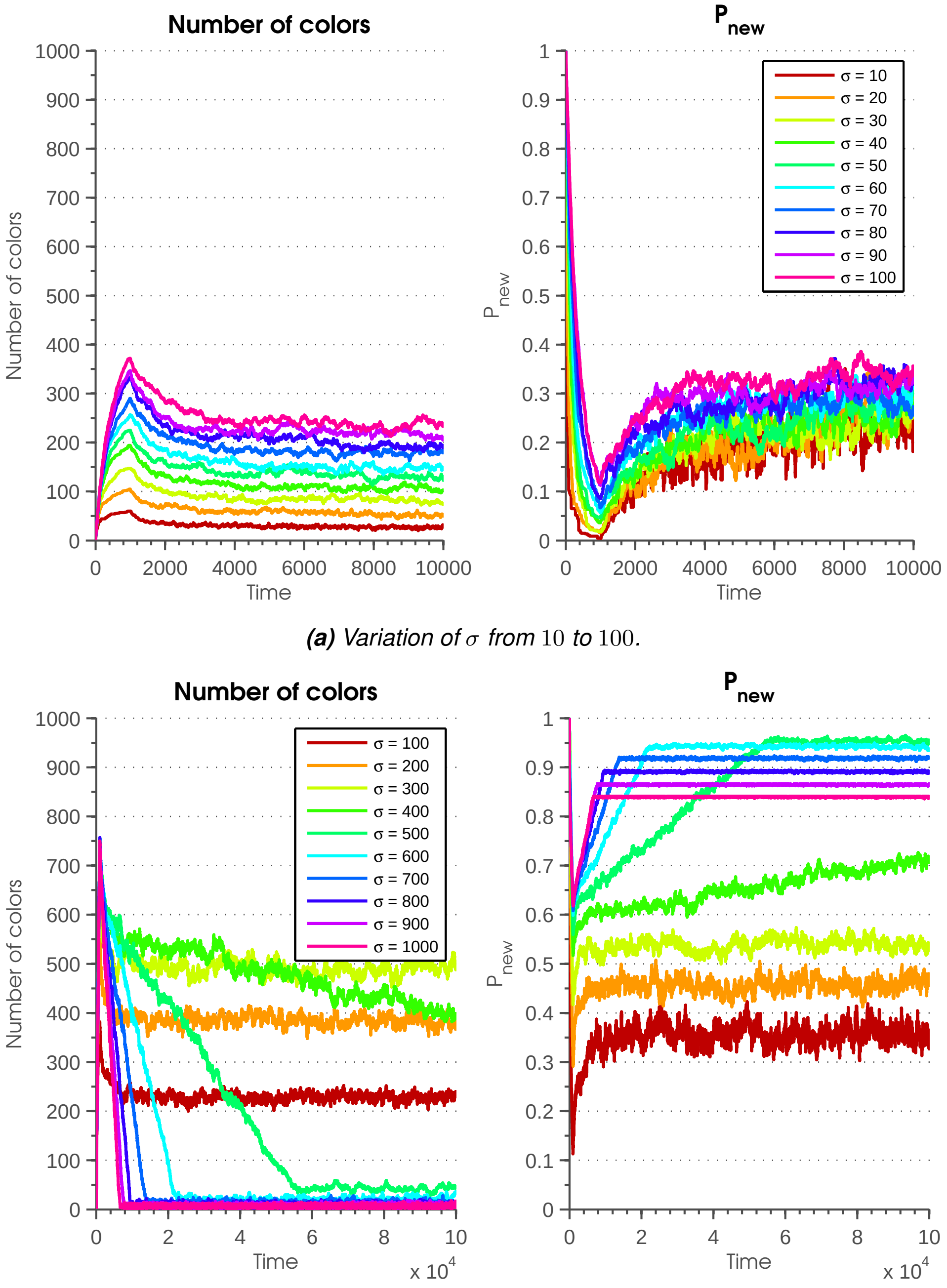
In the case $\mu = \sigma$ there is only one standard deviation at the left side of μ and so something else happens. There are a couple of colors that have, especially compared to the rest of the colors, a much higher probability to be drawn from the source and can be seen as a super color. Therefore, the number of colors in the online structure is really small. P_{new} is big instead because there is a high probability that a color is drawn from the source that is not in the online structure, but the probabilities of the other colors are relatively small. Also, the removal of the balls from the online structure is uniformly at random over the number of colors present, while the probability to draw the super color is big compared to the other colors. This results in the fact that there are a few colors dominating the online structure.

If $\mu = 2\sigma$ there are 2 standard deviations at the left side of the mean with slightly higher probabilities than at the right side of the mean. This explains the behavior in this case, since there is not one color with much higher probability than the other colors but there are some colors with a bit higher probability to be drawn from the source. Still, the removal of the balls from the online structure is uniformly at random over the number of colors present. Therefore, the same kind of behavior occurs as for $\mu = \sigma$.

When $\mu \geq 3\sigma$ the probability distributions are just shifted versions of each other and this explains the result. The expected number of colors in the online structure at $t = K$ is equal to 373 according to (3.1). The left figure shows that this is indeed the case.

Variation of the standard deviation σ

Figures 5.11a and 5.11b show some variations of σ, the standard deviation of the normal distribution. The total number of colors, C, the number of positions in the online structure, K, and the mean of the normal distribution, μ, are fixed.



***(a)** Variation of σ from 10 to 100.*

***(b)** Variation of σ from 100 to 1000.*

***Figure 5.11:** The number of colors in the online structure (left) and the probability to draw a ball from the source that has a color that is not present in the online structure, P_{new}, (right) as a function of the time for different values of σ. $C = 100000$, $K = 1000$ and the probability to draw a color from the source corresponds to a normal distribution with mean $\mu = 1000$ and standard deviation σ.*

These figures show the same behavior as with changing μ. When there is an interval of 3σ at both sides of the mean, nothing special happens. When this is not the case, for example when $\sigma = 800$, then there are some colors with a much higher probability to be drawn from the source compared to the other colors, the super colors. Still, the removal of the balls from the online structure is uniformly at random over the number of colors present in the online structure. Therefore the number of colors in the online structure is rather small, while P_{new} is big instead.

5.4 Poisson Distribution

Instead of a probability vector p that corresponds to a normal distribution, this vector now corresponds to a Poisson distribution with mean λ. It is known that when λ is large, the Poisson distribution is approximated by the normal distribution with mean λ and standard deviation $\sqrt{\lambda}$.

Variation of the total number of colors

In Figure 5.12 the total number of colors C varies. The number of positions in the online structure, K, as well as the mean of the Poisson distribution, λ, are fixed.

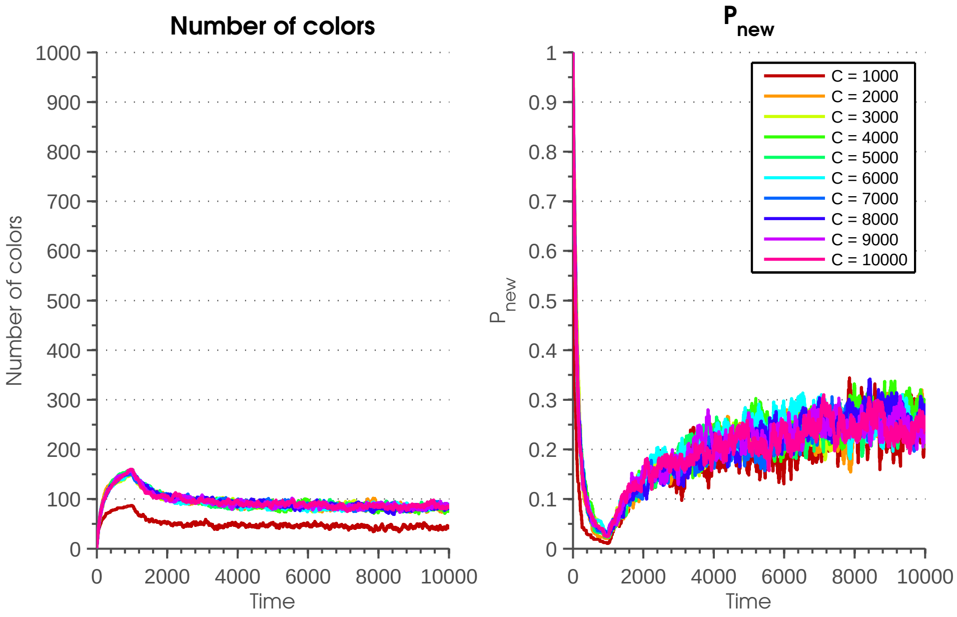


Figure 5.12: *The number of colors in the online structure (left) and the probability to draw a ball from the source that has a color that is not present in the online structure, P_{new}, (right) as a function of the time for different values of C. $K = 1000$ and the probability to draw a color from the source corresponds to a Poisson distribution with mean $\lambda = 1000$.*

This figure shows that P_{new} is independent of the total number of colors C and that the number of colors in the online structure is also independent of C. Note that $C = \lambda$ is an exception and this is discussed later on.

The Poisson distribution is approximated by the normal distribution when λ is large and from the normal distribution it is known that 99.7% of the values are within three standard deviations from the mean. So with the choice of $\lambda = 1000$ there are approximately 190 non-zero elements in p. For this reason the number of colors in the online structure stays quite low. In case when $C \neq \lambda$, for each value of C the probability vectors p are shifted versions of each other and hence P_{new} is independent of the total number of colors C and the number of colors in the online structure is independent of C as well. Due to this shifting of the probability vector, the number of colors in the online structure at time $t = K$ is approximately the same for all values of $C \neq \lambda$. Note that when, for example, $C = 1200$ there are also not yet 3 standard deviations on both sides of the mean of the distribution and therefore the stationary behavior will be slightly different.

In case $C = \lambda$ the mean of the distribution lies at the edge of the range of values, since p has length C. Since almost all the values are within three standard deviations of the mean, which is now only possible at one side of the mean, there are about half of the number of colors in the online structure compared to the case $C \neq \lambda$.

Figure 5.13 shows that indeed the Poisson distribution is approximated by the normal distribution. In this figure the probability vector p has a normal distribution with $\mu = \lambda$ and $\sigma = \sqrt{\mu}$. The parameters C, the total number of colors, and K, the total number of positions in the online structure, are the same as in Figure 5.12.

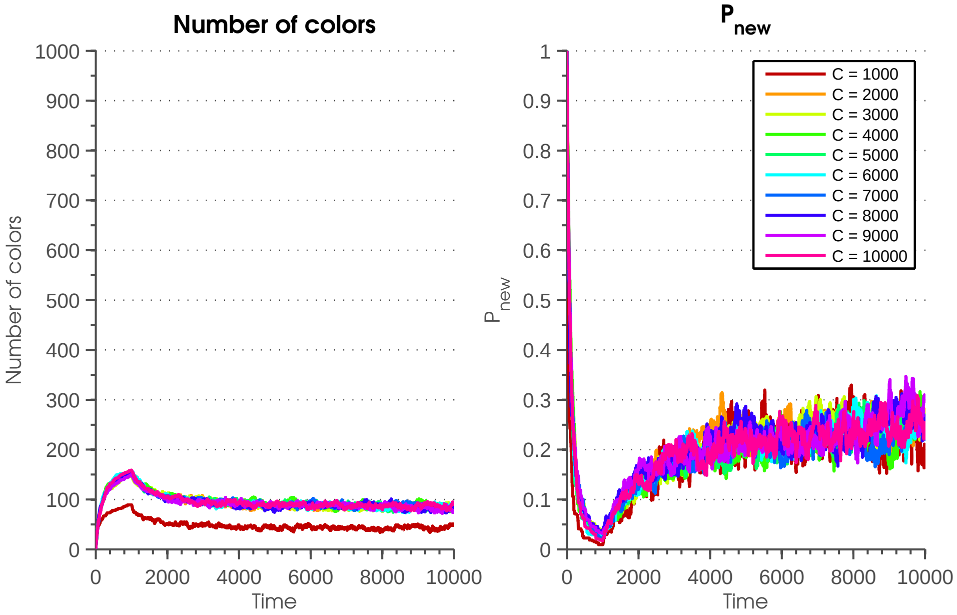


Figure 5.13: *The number of colors in the online structure (left) and the probability to draw a ball from the source that has a color that is not present in the online structure, P_{new}, (right) as a function of the time for different values of C. $K = 1000$ and the probability to draw a color from the source corresponds to a normal distribution with mean $\mu = 1000$ and standard deviation $\sigma = \sqrt{\mu}$.*

This figure is approximately the same as the figure with the probability vector that corresponds to the Poisson distribution. Some difference might occur due to the fact that it is a single simulation.

Variation of the number of positions in the online structure

The following figure shows a variation of K, the number of positions in the online structure. The total number of colors C and the mean of the Poisson distribution λ are fixed.

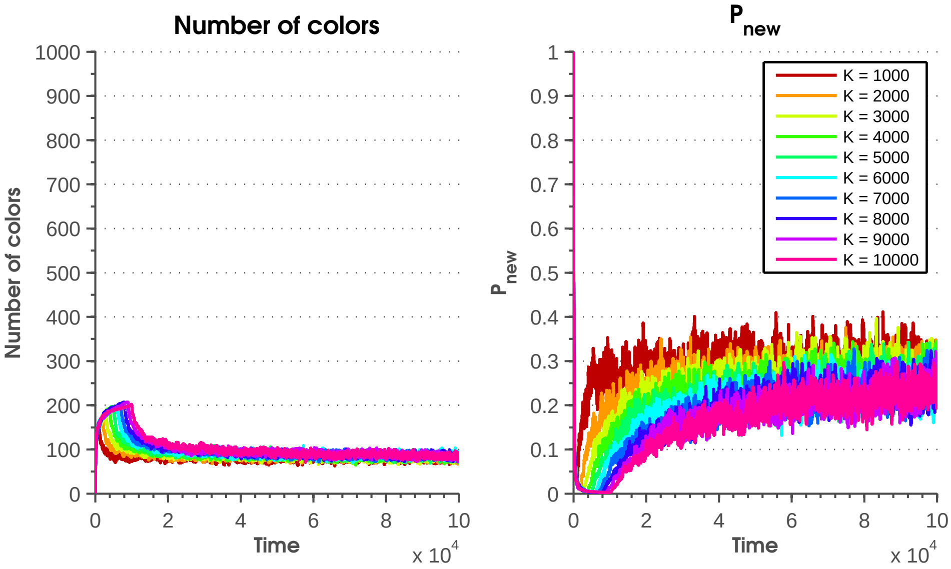


Figure 5.14: *The number of colors in the online structure (left) and the probability to draw a ball from the source that has a color that is not present in the online structure, P_{new}, (right) as a function of the time for different values of K. $C = 100000$ and the probability to draw a color from the source corresponds to a Poisson distribution with mean $\lambda = 1000$.*

Figure 5.14 shows that when K is getting closer to C the probability of drawing a ball from the source that has a color that is not present in the online structure converges to the same value for all K. The number of colors in the online structure also converges to the same value for all K. Note that in this case the number of positions in the online structure is greater than the number of colors, since there are a lot of colors with probability of zero to be drawn from the source. Besides, the number of colors with probability of (almost) zero to be drawn from the source remains the same when changing K. These observations also explain why the number of colors in the online structure at time $t = K$ are hardly dependent on K.

Variation of the mean λ

In the last figure with a Poisson distributed probability vector p there is a variation of λ, the mean of the Poisson distribution. The total number of colors C and the number of positions in the online structure K are fixed.

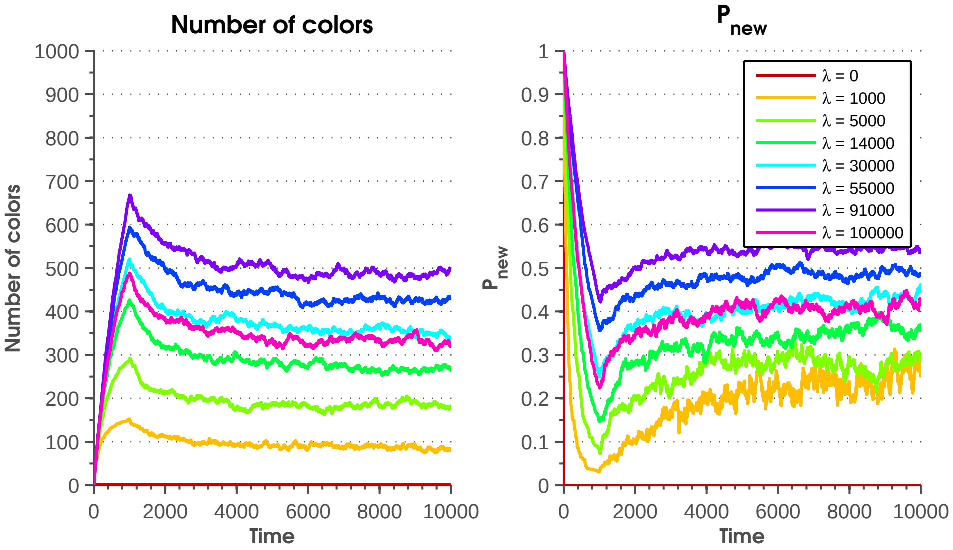


Figure 5.15: *The number of colors in the online structure (left) and the probability to draw a ball from the source that has a color that is not present in the online structure, P_{new}, (right) as a function of the time for different values of λ. $C = 100000$, $K = 1000$ and the probability to draw a color from the source corresponds to a Poisson distribution with mean λ.*

Figure 5.15 shows that when λ increases, P_{new} and the number of colors in the online structure also increase. This is indeed true because when the Poisson distribution has a higher mean, it also has a larger variance and hence a larger standard deviation. Therefore, since the Poisson distribution is approximated by the normal distribution for high values of λ, there are less colors which have a probability of almost 0 to be drawn from the source. There are two exceptions, namely the case that $\lambda = 0$ and when $\lambda = C$. When $\lambda = 0$, P_{new} of course is zero since the mean and variance are 0. For the other case, the mean of the distribution lies at the end of the interval and therefore the result is slightly different. The number of colors in the online structure after K steps is according to (3.1), e.g., when $\lambda = 55000$ the expected number of colors in the online structure at time $t = K$ equals 599.

5.5 Power Law

The probability vector p is in this section distributed according to the power law. This means that the probability for a certain color $c = 1, ..., C$ is

$$P(color = c) = \frac{c^{-\gamma}}{\sum_{n=1}^{C} n^{-\gamma}}.$$

A super color is a color that has a big probability to be drawn from the source, especially compared to the other colors. A super color dominates the online structure and it is possible that there is more than one super color.

Variation of the total number of colors

The changing variable in this section is C, the total number of colors. The number of positions in the online structure K is fixed, as well as γ.

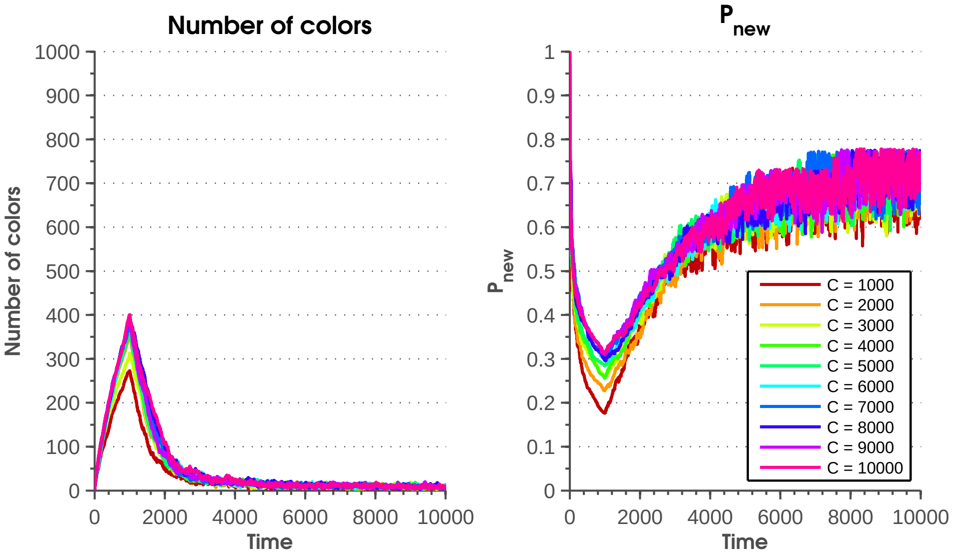


Figure 5.16: *The number of colors in the online structure (left) and the probability to draw a ball from the source that has a color that is not present in the online structure,* P_{new}*, (right) as a function of the time for different values of* C*.* $K = 1000$ *and the probability to draw a color from the source corresponds to a power law distribution with* $\gamma = 1.1$*.*

When adding K balls to the online structure, the online structure is filled with a certain amount of different colored balls. This is in accordance with (3.1), for $C = 10000$ the expected number of colors in the online structure at time $t = K$ equals 404. However, when removing a ball from the online structure this is done uniformly at random over the number of colors in the online structure while the probability to draw one of the super colors in every step is big. In the end the online structure is highly dominated by just a few colors, while the probability to draw a ball from the source that has a color that is not present in the online structure is relatively big. This is exactly what the designers of the BUbiNG web crawler observed.

Also, note that the number of colors in the online structure is lower than with the exponential distribution of the probability vector p. This is due to the choice of the mean of the exponential distribution, λ.

Variation of the number of positions in the online structure

A variation of K, the number of positions in the online structure, gives the result as shown in Figure 5.17. The total number of colors C and γ are fixed.

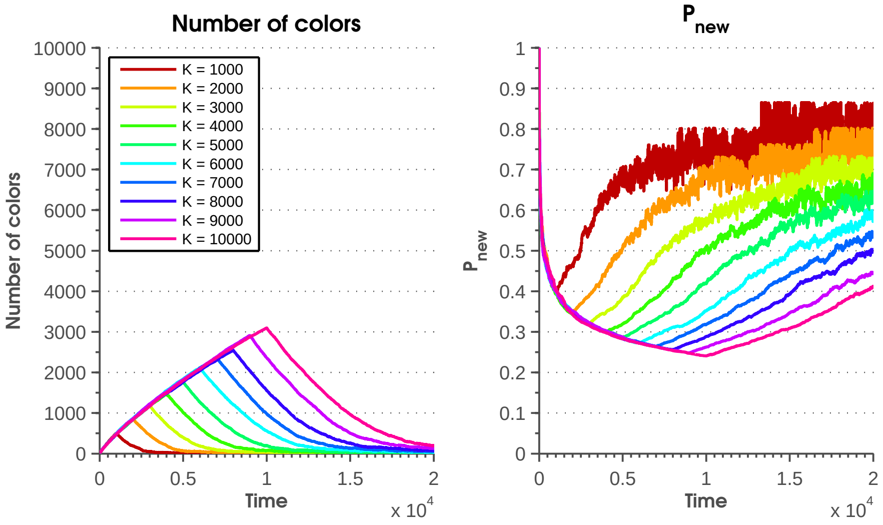


Figure 5.17: *The number of colors in the online structure (left) and the probability to draw a ball from the source that has a color that is not present in the online structure, P_{new}, (right) as a function of the time for different values of K. $C = 100000$ and the probability to draw a color from the source corresponds to a power law distribution with $\gamma = 1.1$.*

Increasing K only results in some more colors in the online structure during the initialization. Afterwards the super colors come into play and in stationarity the number of colors in the online structure is small. When γ is rather small, the probability for each color to occur tends to 1. By normalization, the cumulative distribution function is the same as the one from the uniform distribution. This is what can be seen in the Figure 5.18.

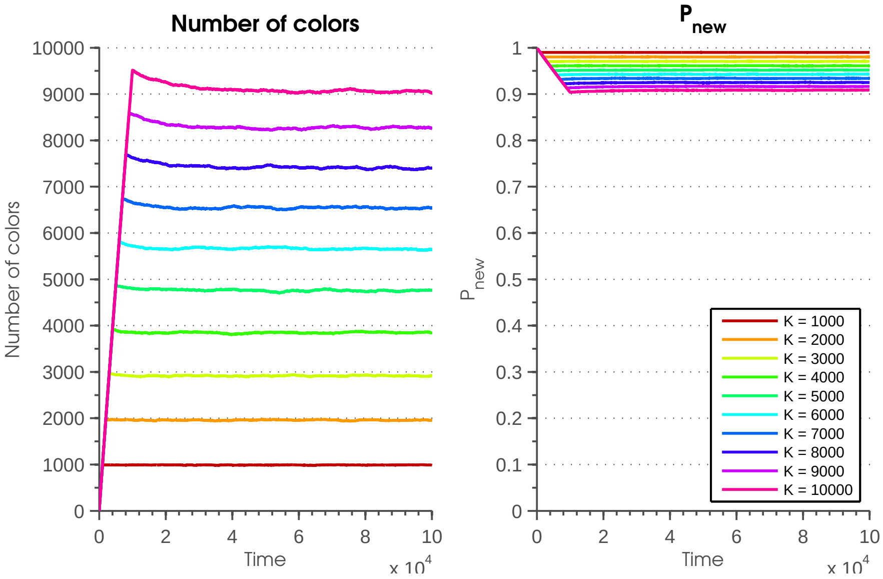


Figure 5.18: *The number of colors in the online structure (left) and the probability to draw a ball from the source that has a color that is not present in the online structure, P_{new}, (right) as a function of the time for different values of K. $C = 100000$ and the probability to draw a color from the source corresponds to a power law distribution with $\gamma = 0.1$.*

Note that the power law distribution is exactly equal to the uniform distribution when $\gamma = 0$.

Variation of γ

What happens when changing γ? The previous paragraph showed that when γ tends to 0 the same behavior occurs as when the probability vector p did correspond to a uniform distribution. The following figure shows variation of γ and therefore, among other things, this behavior is observed again.

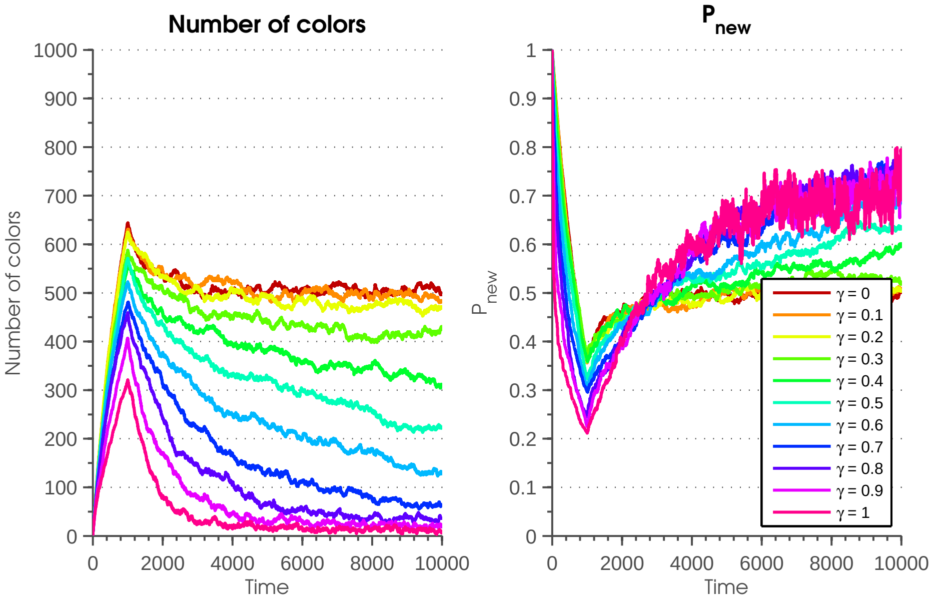


Figure 5.19: *The number of colors in the online structure (left) and the probability to draw a ball from the source that has a color that is not present in the online structure, P_{new}, (right) as a function of the time for different values of γ. $C = K = 1000$ and the probability to draw a color from the source corresponds to a power law distribution with parameter γ.*

When $C = K$ and γ grows, the number of colors present in the online structure decreases. This is due to the fact that in that case there are a few colors with a high probability to be drawn from the source and all the other colors have a small probability. Besides, the removal of the balls from the online structure happens uniformly at random over the number of colors present in the online structure. What happens when $C \neq K$ can be seen in the following figure.

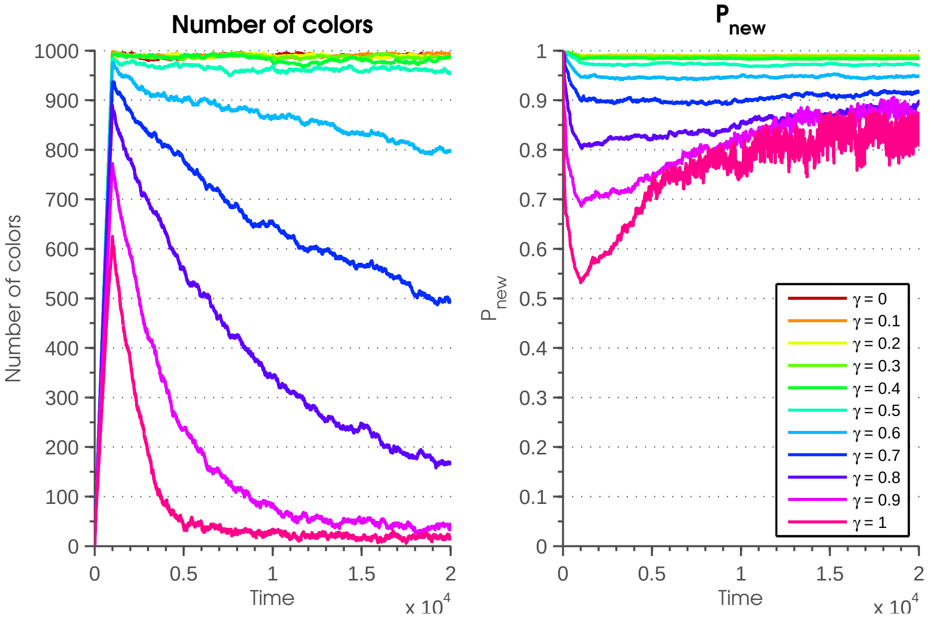


Figure 5.20: *The number of colors in the online structure (left) and the probability to draw a ball from the source that has a color that is not present in the online structure, P_{new}, (right) as a function of the time for different values of γ. $C = 100000$, $K = 1000$ and the probability to draw a color from the source corresponds to a power law distribution with parameter γ.*

This is a more extreme case than when $C = K$. Since there are more available colors, in the beginning there are more colors present in the online structure. Afterwards, the super colors come into play and so the number of colors in the online structure converges to a relatively small number. The smaller γ the smaller the biggest probability from the probability vector p, converging towards a uniform distribution of p. Therefore the decrease in the number of colors in the online structure is less fast when γ is smaller or there is no decrease at all. Increasing γ even more gives the expected behavior, as can be seen in Figure 5.21.

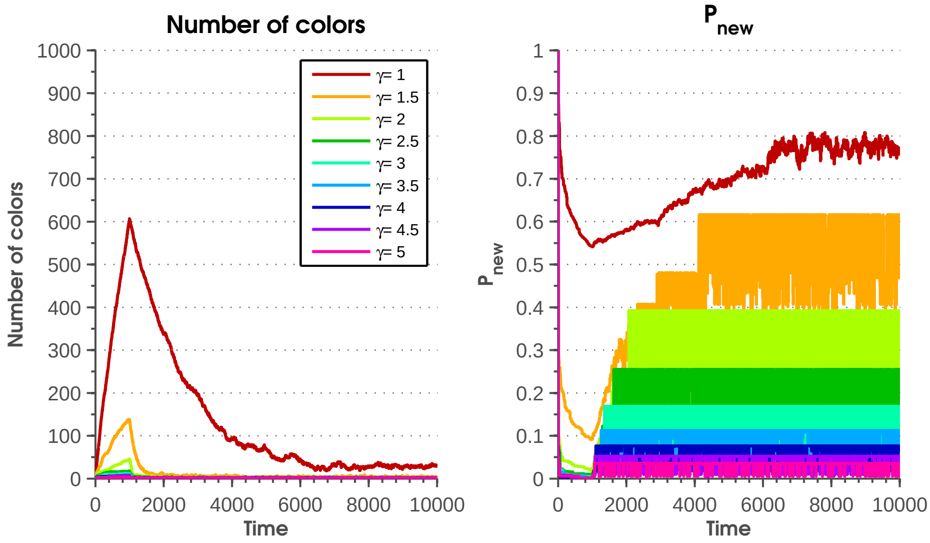


Figure 5.21: *The number of colors in the online structure (left) and the probability to draw a ball from the source that has a color that is not present in the online structure, P_{new}, (right) as a function of the time for different values of γ. $C = 100000$, $K = 1000$ and the probability to draw a color from the source corresponds to a power law distribution with parameter γ.*

Note that for $\gamma = 1$ the behavior looks different than in Figure 5.20, but this is due to the scaling of the x-axis. In Figure 5.21 it is observed that when γ increases, the super color(s) dominate the online structure even more. Especially in the first K steps the super color is dominating more. Taking into account that when $\gamma = 5$ the probability for the super color equals 0.9644, this behavior is expected. The probability to draw a color from the source that is not present in the online structure, P_{new}, is unstable. What happens in case $\gamma = 5$ is the following. The number of colors in the online structure very often equals 1. If another color is added to the online structure, for example the color with the second highest probability to be drawn from the source, the probability to draw a ball from the source that has a color that is not present in the online structure decreases by 0.0301. Since there are only two colors present in the online structure, the probability to remove this color from the online structure is big compared to the probability of adding it. This is why P_{new} seems unstable, but this is caused by just a few of colors that are added to the online structure. For other values of γ the reasoning is similar.

Chapter 6

Data Analysis

We obtained the data from performed crawls showing the number of hosts with the number of associated pages. Every color represents a host and every page is a number. The colors are ordered by the number of associated pages and therefore color number 1 represents the host with the most associated pages. A plot showing the probability of each color for the biggest crawl performed this far with the BUbiNG web crawler is shown in Figure 6.1.

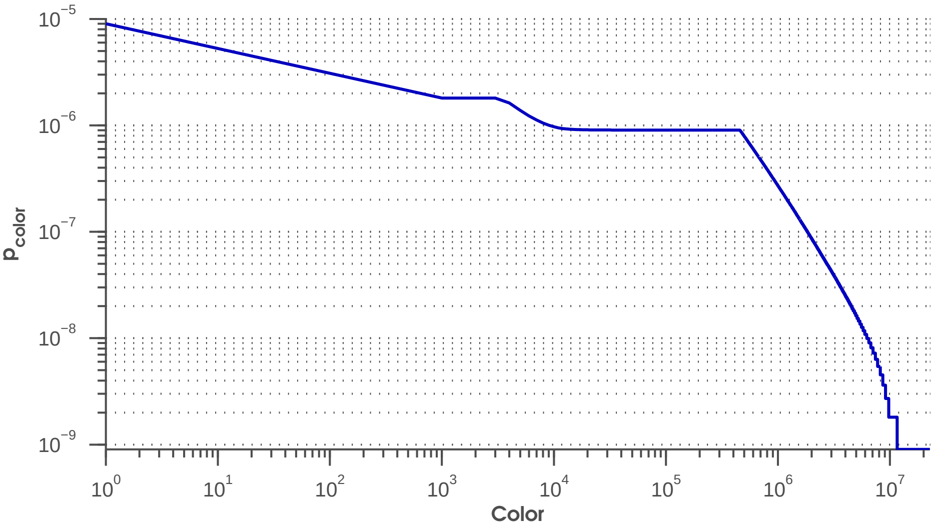


Figure 6.1: *The probability for each color to be drawn from the source from the biggest crawl performed thus far with the BUbiNG web crawler. Colors are sorted by decreasing probability.*

If the host sizes would be power law distributed, the log log plot would show a straight line. Still, there is a certain number of colors that is more likely to be drawn from the source than others. Therefore it is assumed that the colors follow a power law distribution. This data shows that about 10% of the colors has a high probability to be drawn from the source, while the other 90% of the colors have a small probability to be drawn from the source.

Chapter 7

Strategies for the use of the Offline Structure

In this chapter we introduce the offline structure into the model. Corollary 4.2.2 states that when p corresponds to a uniform distribution and C tends to infinity, no offline structure is needed in order to have a maximal throughput of the web crawler. Unfortunately, in practice, p does not correspond to a uniform distribution as has been seen in Chapter 6. Since the distribution of the colors follows most likely a power law distribution, this distribution is used. Note that the first K balls are added to the online structure in the same way as when there was no offline structure, no other strategy is used for this unless mentioned otherwise. Furthermore it is assumed that the offline structure is a first in first out (FIFO) queue.

For the use of the offline structure, several strategies are evaluated for maximizing the number of colors in the online structure. For every strategy for the use of the offline structure there is a strategy for the source, i.e., when to accept a ball from the source, and there is a strategy for the offline structure, i.e., when to accept a ball from the offline structure. Finally, priority is given to one of them, i.e., the player has to start with either the source or the offline structure.

The first strategies that are introduced are the neutral strategy and the greedy strategy. In the neutral strategy every ball is accepted, whereas in the greedy strategy a ball is only accepted when it has a color that is not present in the online structure. In Section 7.1 and Section 7.2 the neutral strategy and the greedy strategy are considered as a strategy for both the source and the offline structure.

7.1 Neutral Strategy for the Source

In the neutral strategy for the source, every ball is accepted from the source. For the offline structure there are two possibilities. Priority can be given to both the source and the offline structure.

Proposition 7.1.1. *The player should be selective with the source.*

Proof. Suppose that priority is given to the source, this means that no ball is added to the offline structure. Moreover, the offline structure is not even used and the results are the same as without the use of the offline structure.

Suppose that priority is given to the offline structure. Since the offline structure is empty at the beginning of the game and because no ball will be added to the offline structure, there is no difference to the case that priority is given to the source.
If the offline structure is filled with, for example, K balls on beforehand the offline structure can be used. Still, independent of the strategy for the offline structure, the offline structure does get empty at some point in time, because there are no balls added to the offline structure. Once the offline structure is

empty, only balls from the source can be accepted. Again the results are the same as without the use of the offline structure. The statement follows. □

7.2 Greedy Strategy for the Source

In the previous section it has been shown that the player should be selective with the source. Therefore in this section a ball is accepted from the source if it has a color that is not yet present in the online structure. Otherwise the ball is put in the offline structure. The strategies in this section are denoted by ”strategy source - strategy offline structure - priority”.

7.2.1 Neutral Strategy for Offline Structure

The first strategy that is explored is that a ball from the source is accepted greedily and a ball from the offline structure is accepted neutral. Priority can be given to both the source and the offline structure. The strategies can be found in Algorithm 1 and Algorithm 2.

Algorithm 1 Greedy - Neutral - Source Strategy

```
Draw a ball from the source.
if this ball has a color that is not present in the online structure then
    if the offline structure contains this color then
        put the ball in the offline structure.
        Accept a ball from the offline structure.
    else
        put the ball in the online structure.
    end if
else
    put the ball in the offline structure.
    Accept a ball from the offline structure.
end if
```

Algorithm 2 Greedy - Neutral - Offline Structure Strategy

```
if the offline structure is non-empty then
    accept a ball from the offline structure.
else
    use Algorithm 3.
end if
```

Note that Algorithm 3 is the algorithm to accept a ball from the source that has a color that is not present in the online structure. This algorithm is used in other strategies as well.

Algorithm 3 Accept a ball from the source that has a color that is not present in the online structure.

```
while no ball has been added to the online structure do
    Draw a ball from the source.
    if this ball has a color that is not present in the online structure then
        if the offline structure contains this color then
            put the ball in the offline structure.
        else
            put the ball in the online structure.
        end if
    else
        put the ball in the offline structure.
    end if
end while
```

Proposition 7.2.1. *The Greedy - Neutral - priority strategy gives the same results in stationarity as without the use of the offline structure for any priority-rule.*

Proof. Suppose that priority is given to the source. If a ball is not accepted from the source, this ball goes to the offline structure. Afterwards a ball is accepted from the offline structure, which is the ball that has just been put in the offline structure.

If priority is given the the offline structure, the following happens. As long as the offline structure is empty, only the source plays a role. However, there will be some balls drawn from the source that are not accepted to the online structure. Since the offline structure is not empty anymore, the offline structure has priority. Therefore all the balls that have been put in the offline structure are accepted to the online structure in the next steps. The statement follows. □

Since we know that the strategies mentioned in Algorithms 1 and 2 do not work, we try the following variant. Suppose now that priority is given to the source, but that the offline structure is filled with, e.g., K balls beforehand. Figure 7.1 presents the results.

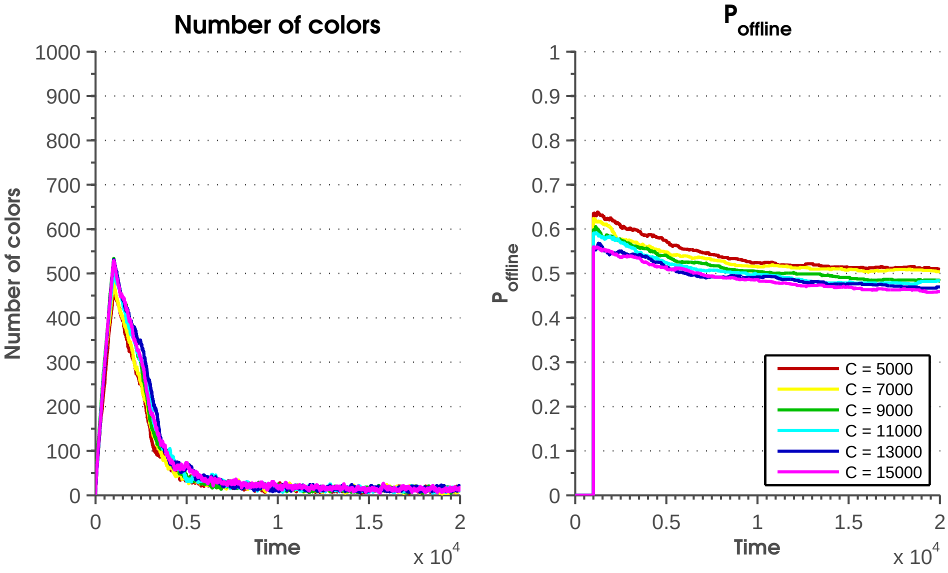


Figure 7.1: *The number of colors in the online structure (left) and the probability that a color is in the offline structure, $P_{offline}$, (right) as a function of the time for different values of the total number of colors C. The strategy can be found in Algorithm 2. The number of positions in the online structure $K = 1000$ and $\gamma = 1$*

Note that $P_{offline}$ is dependent on the initialization of the number of balls in the offline structure. In this case the offline structure has been filled with K balls.

Proposition 7.2.2. *The Greedy-Neutral-Source strategy when the offline structure is filled on beforehand with a certain number of balls gives the same results in stationarity as without the use of the offline structure.*

Proof. Suppose that a ball is not accepted from the source, then this ball goes to the offline structure and is replaced by a ball that is drawn from the same probability distribution. The statement follows. □

7.2.2 Greedy Strategy for Offline Structure

The other possibility is to accept a ball from the offline structure greedily as well. Priority is given to the offline structure, see Algorithm 4 for the Greedy - Greedy - Offline Structure strategy.

Algorithm 4 Greedy - Greedy - Offline Structure Strategy

```
if the first ball in the offline structure has a color that is not present in the online structure then
    accept a ball from the offline structure.
else
    use Algorithm 3.
end if
```

Only accepting a ball when it has a color that is not present in the online structure seems like a good strategy, however after a while all the colors are present in the offline structure and it is not possible anymore to continue. This leads to the following proposition.

Proposition 7.2.3. *The Greedy - Greedy - Offline Structure Strategy, see Algorithm 4, can not be continued after finite time.*

Proof. After finite time the offline structure contains (almost) all colors and therefore it is computationally hard, or even impossible, to find a color that is neither present in the online structure nor present in the offline structure.

Moreover it might happen that, after removing a ball from the online structure, the colors that are not present in the offline structure are present in the online structure. This means that, when the union of colors in the online structure and offline structure is equal to all possible colors, it is impossible to draw a color from the source that is neither present in the online structure nor present in the offline structure. □

Figure 7.2 gives an example of this behavior, although the simulation is stopped when the probability that a color is present in the offline structure is greater than 0.99. Beyond that, it is computationally hard to draw a color that is not present in the online structure from the source that can be accepted to the online structure.

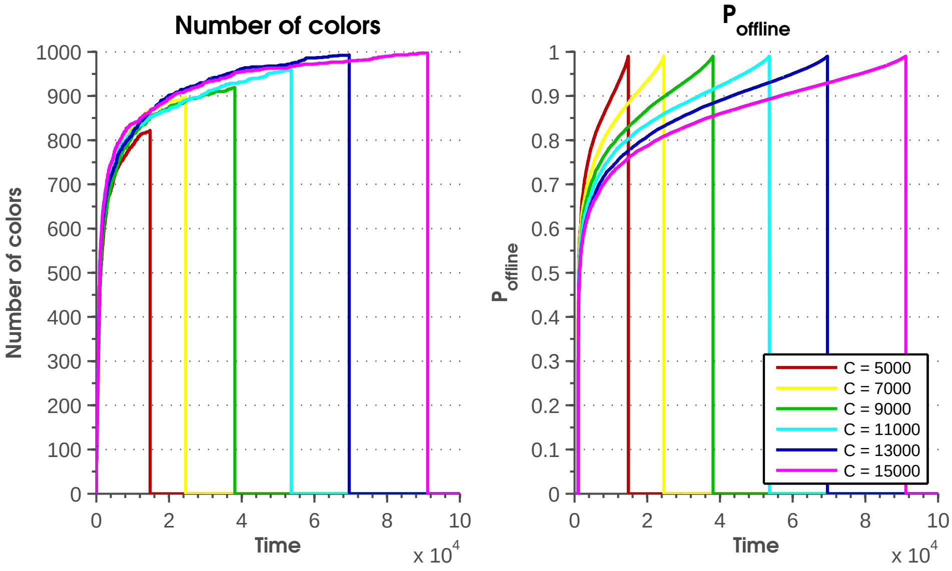


Figure 7.2: *The number of colors in the online structure (left) and the probability that a color is in the offline structure, $P_{offline}$, (right) as a function of the time for different values of the total number of colors C. The strategy can be found in Algorithm 4. The number of positions in the online structure $K = 1000$ and $\gamma = 1$. The drop occurs when the simulation stops.*

Adapting the strategy in such a way that when the probability that a color is in the offline structure is greater than 0.99, a ball from the offline structure is accepted gives the results presented in Figure 7.3. The strategy is can be found in Algorithm 5.

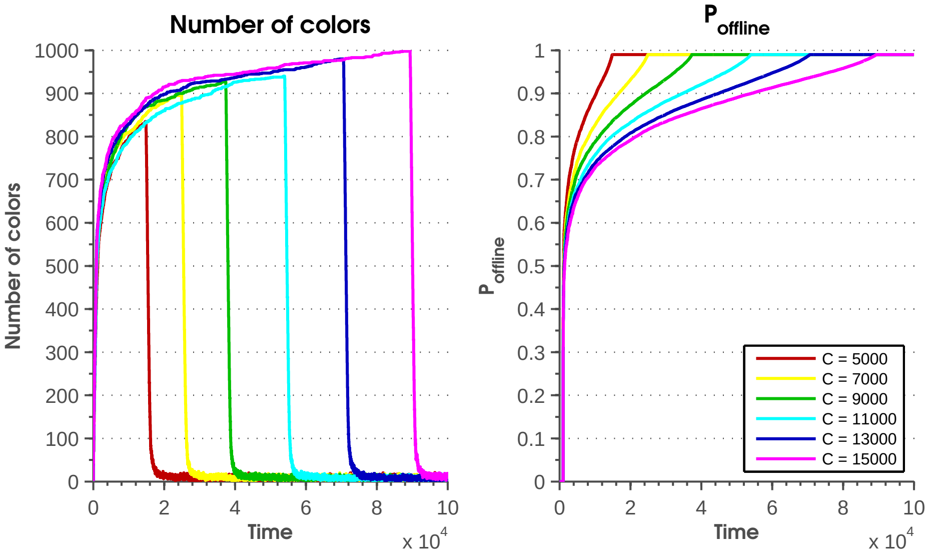


Figure 7.3: *The number of colors in the online structure (left) and the probability that a color is in the offline structure, $P_{offline}$, (right) as a function of the time for different values of the total number of colors C. The strategy can be found in Algorithm 5. The number of positions in the online structure $K = 1000$ and $\gamma = 1$.*

Algorithm 5 Greedy - Greedy - Offline Structure Strategy - 2

```
if the first ball in the offline structure has a color that is not present in the online structure then
    accept a ball from the offline structure.
else
    while no ball has been added to the online structure do
        if the probability that a color is in the offline structure is greater than $0.99$ then
            accept a ball from the offline structure.
        else
            Draw a ball from the source.
            if this ball has a color that is not present in the online structure then
                if the offline structure contains this color then
                    put the ball in the offline structure.
                else
                    put the ball in the online structure.
                end if
            else
                put the ball in the offline structure.
            end if
        end if
    end while
end if
```

Proposition 7.2.4. *The Greedy - Greedy - Offline Structure Strategy - 2, see Algorithm 5, does not improve the stationary behavior of number of colors in the online structure.*

Proof. When the probability that a color is in the offline structure is greater than 0.99, balls from the offline structure are accepted. The offline structure is filled with colors that have been rejected to the online structure earlier and therefore contains the super colors. These super colors are now accepted to the online structure and dominate the online structure. □

Since the previous adaptation does not give the desired result, the strategy in Algorithm 4 is adapted differently and when the second ball in the offline structure has a color that is not present in the online structure the ball before this one is also accepted. This strategy can be found in Algorithm 6. It is known that it in practice it takes a lot of time to do this search in the queue, although it is still checked whether this improves the result. Also in this case the simulation is stopped when the probability that a color is present in the offline structure is greater than 0.99, since accepting a ball from the offline structure when this is the case results in the super colors dominating the online structure.

Algorithm 6 Greedy - Greedy - Offline Structure Strategy - a

```
if the first ball in the offline structure has a color that is not present in the online structure then
    accept a ball from the offline structure.
else if the second ball in the offline structure has a color that is not present in the online structure then
    accept a ball from the offline structure.
else
    use Algorithm 3.
end if
```

Proposition 7.2.5. *The Greedy - Greedy - Offline Structure Strategy - a, see Algorithm 4, can not be continued after finite time.*

Proof. The proof follows the same reasoning as for Proposition 7.2.3. Noting that the second ball in the offline structure usually does not have a color that is not present in the online structure, because all the dominating colors have been put in the offline structure, concludes the proof. □

Numerical results of this strategy can be found in Figure 7.4.

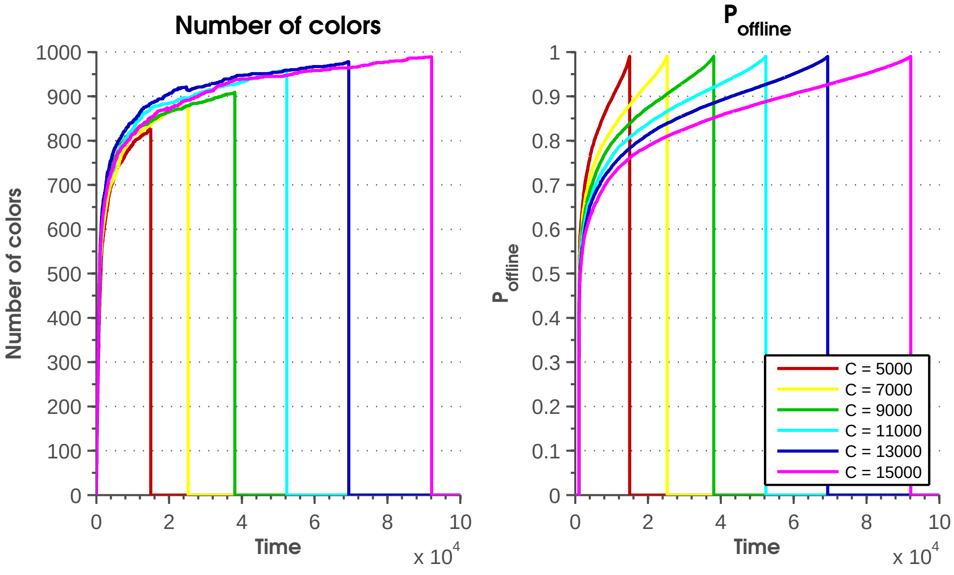


Figure 7.4: *The number of colors in the online structure (left) and the probability that a color is in the offline structure, $P_{offline}$, (right) as a function of the time for different values of the total number of colors C. The strategy can be found in Algorithm 6. The number of positions in the online structure $K = 1000$ and $\gamma = 1$. The drop occurs when the simulation stops.*

This strategy does not improve the previous strategy, see Figure 7.2, and it also does not increase the time until the probability that a color is present in the offline structure is greater than 0.99. Following similar reasoning as in the proof of proposition 7.2.5, checking whether the third ball in the offline structure has a color that is not present in the online structure does not result in an improvement of the number of colors in the online structure as well. This has been tested numerically and the numerical results support this conjecture. Since the search in the queue in the offline structure takes a lot of time in practice and is computationally demanding, a strategy in which all the balls before the first ball with a color that is not present in the online structure are accepted is not taken into account.

In this subsection priority has been given to the offline structure. Another possibility would be to give priority to the source. However, since balls from the offline structure are accepted greedily it is not necessary to check the offline structure more than once in every step. Remember that it could be computationally hard to draw a ball from the source that has a color that is not present in the online structure. Therefore it is better to check if the offline structure has a ball that has a color that is not present in the online structure before a ball is drawn from the source. The case to give priority to the source is therefore not taken into account. Considering all the results, the expectation is that the result is similar to the case where priority is given to the offline structure.

7.3 Randomized Strategy

The next strategy for the offline structure that is considered is randomization. This strategy can be found in Algorithm 7.

Algorithm 7 Randomized Strategy

```
With probability $r$ accept a ball from the source.
With probability $1 - r$ do
if the offline structure is non-empty then
    accept a ball from the offline structure.
else
    accept a ball from the source.
end if
```

Note that in the randomized strategy no priority is given the source nor to the offline structure.

Proposition 7.3.1. *In the randomized strategy, see Algorithm 7, no ball is added to the offline structure.*

Proof. As long as the offline structure is non-empty, a ball from the source is accepted to the online structure. □

Therefore the results are the same as without the offline structure. It also does not make sense to add, for example, K balls to the offline structure, because the offline structure does get empty at some point in time and then again no ball is added to the online structure. The following version is explored.

Algorithm 8 Randomized Strategy - a

```
while no ball has been added to the online structure do
    Draw a ball from the source.
    if the probability that a color is in the offline structure is greater than $0.99$ then
        put the ball in the offline structure.
        Accept the first ball from the offline structure.
    else
        With probability $r$
        if the color of the ball is already present in the offline structure then
            put the ball in the offline structure.
        else
            put the ball in the online structure.
        end if
        With probability $1 - r$
        put the ball in the offline structure.
        Accept the first ball from the offline structure.
    end if
end while
```

Proposition 7.3.2. *In the Randomized strategy - a, see Algorithm 8, the result in stationarity is the same as without the use of the offline structure.*

Proof. If a ball goes to the offline structure, a ball is accepted from the offline structure as well. This is the same ball, so the statement follows. □

Also when it is checked whether the offline structure contains 99% of the probability distribution before drawing a ball from the source, the offline structure remains empty and the reasoning is similar. The results are therefore the same as without offline structure.

Another possibility can be found in Algorithm 9. Numerical results are presented in Figure 7.5.

Algorithm 9 Randomized Strategy - b

```
With probability $r$ do
if the probability that a color is in the offline structure is greater than 0.99 then
    accept a ball from the offline structure.
else
    use Algorithm 3.
end if
With probability $1 - r$ do
if the offline structure is non-empty then
    accept a ball from the offline structure.
else
    accept a ball from the source.
end if
```

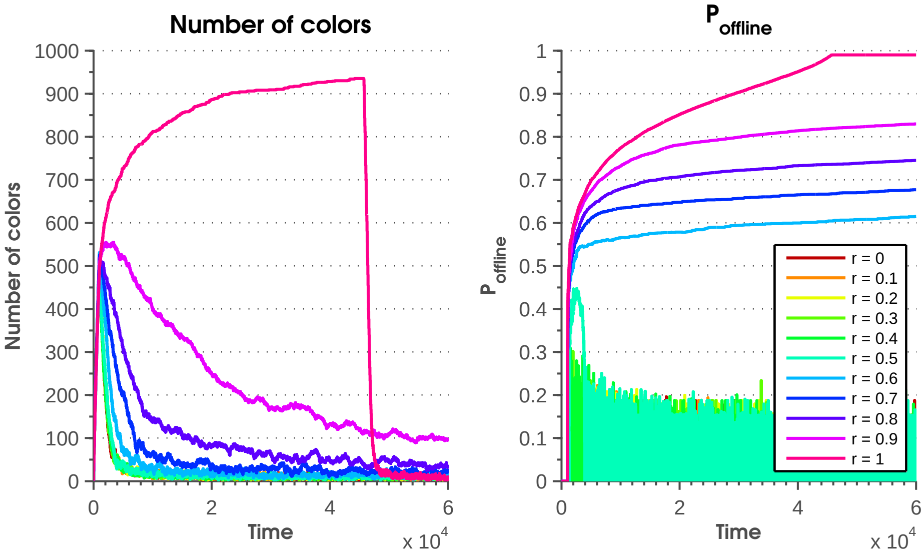


Figure 7.5: *The number of colors in the online structure (left) and the probability that a color is in the offline structure, $P_{offline}$, (right) as a function of the time for different values of r. The strategy can be found in Algorithm 9. The total number of colors $C = 10000$, the number of positions in the online structure $K = 1000$ and $\gamma = 1$.*

So it is actually hard to make a good use of the offline structure. In the first steps the behavior of the randomized strategy is better for certain values of r. However, stationarity is showing the same behavior as when there was no offline structure. Notice that when $r = 0$ there will not be any ball in the offline structure and therefore the result is the same as it was without the offline structure. When $r = 1$ the result is almost the same as when only accepting a ball that has a color that is not present in the online structure. However, now only from the source a ball with a color that is not present in the online structure is accepted. Since the strategy when a ball is accepted greedily from both the source and the offline structure works good until most of the colors are present in the offline structure, we combine the Randomized strategy - b and the Greedy-Greedy-Offline Structure Strategy as presented in Algorithm 10. Figure 7.6 shows what happens when changing r.

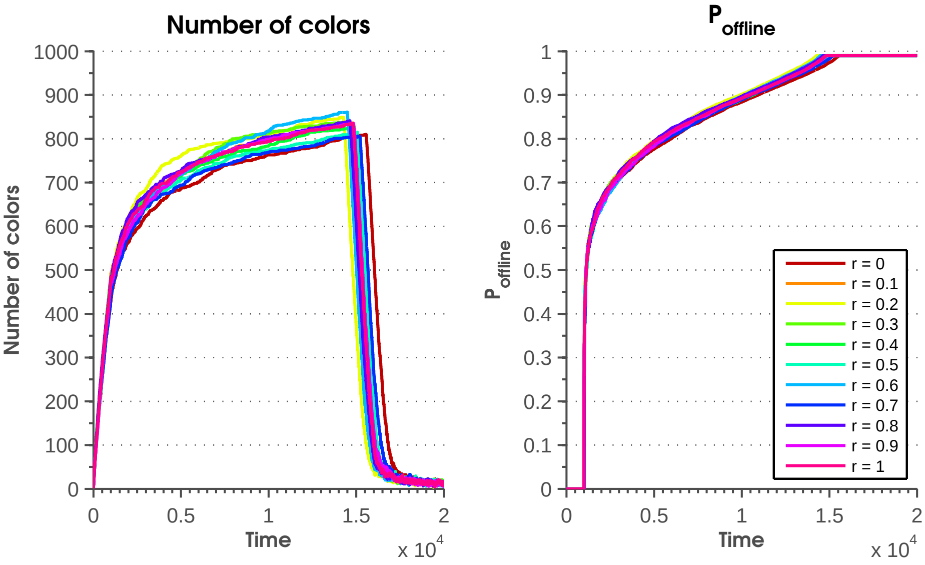


Figure 7.6: *The number of colors in the online structure (left) and the probability that a color is in the offline structure, $P_{offline}$, (right) as a function of the time for different values of r. The strategy can be found in Algorithm 10. The total number of colors $C = 5000$, the number of positions in the online structure $K = 1000$ and $\gamma = 1$.*

Algorithm 10 Randomized Strategy and Greedy-Greedy-Offline Structure Strategy combined

```
if the probability that a color is in the offline structure is greater than $0.99$ then
    Draw a ball from the source.
    if this ball has a color that is not present in the online structure then
        put the ball in the online structure.
    else
        With probability $r$
        if possible then accept the ball to the online structure.
        else
            put the ball in the offline structure.
            Accept a ball from the offline structure.
        end if
        With probability $1 - r$
        put the ball in the offline structure.
        Accept a ball from the offline structure.
    end if
else
    use Algorithm 3.
end if
```

Note that it can be impossible to accept a color from the source that is not present in the online structure. It might happen that all the colors that are not in the offline structure are in the online structure so it is impossible to find a color that both is not present in the online structure and can be accepted to the online structure. The randomization combined with the Greedy-Greedy-Offline Structure Strategy has no effect at all.

7.4 Mice and Elephant Strategy

Considering all the results, the above strategies result in a small number of colors in the online structure in stationarity. The upcoming strategy is inspired by M. Crovella et al. [4]. In this article the authors examine whether web servers might provide better service by using non-traditional service ordering for connections. Their goal is to improve the mean response time of the web server and it turns out that it is better to schedule the connections first with the shortest remaining processing time instead of using a processor sharing discipline.
Our goal is to maximize the number of colors present in the online structure. The removal of the balls from the online structure can be seen as a sort of shortest remaining processing time when there are K colors present, because when doing so priority is given to colors with smaller probability to be drawn from the source as much as possible.

The notion of ”mice and elephants” was considered in the literature, for example in telecommunications. By L. Guo and I. Matta [7] mice are defined as the internet connections that are short in terms of the amount of traffic they carry, while elephants are the small fraction of the connections that carry a large portion of the traffic. N. Brownlee and K. Claffy [3] define elephants as ”high volume transmission control protocol (TCP) streams” and mice are the ”small TCP streams”. As a last example, by M.A. Marsan et al. [10], mice are the short-lived TCP flows and elephants are the long-lived TCP flows.

Following similar analogy, in the mice and elephant strategy the colors are divided into two classes. The first class is the one that corresponds to the large hosts, the elephants. The colors corresponding to the large hosts have high probability to be drawn from the source. The second class is the one that corresponds to the small hosts, the mice. The colors corresponding to the mice have smaller probability to be drawn from the source.

The strategy can be found in Algorithm 11. From now on every elephant color has it’s own queue in the offline structure.

Algorithm 11 Mice and Elephant Strategy

```
if a color is present in the offline structure, while it is not present in the online structure then
    accept this ball to the online structure.
else
    while no ball has been added to the online structure do
        Draw a ball from the source.
        if this ball has a color that belongs to the mice then
            accept the ball to the online structure.
        else
            put the ball in the offline structure in the queue corresponding to the color.
        end if
    end while
end if
```

Note that priority is given to the offline structure. If it is possible to accept a ball from the offline structure the player of the game does not have to hope to draw a ball from the source with a color that belongs to the mice, but can immediately accept a ball that has a color that is not present in the online structure.

The next question is, when a color has to be classified as mice and when color has to be classified as elephant. Let c be the number of elephant colors and let the total probability of the elephants be $1 - q$.
A first guess would be that when the probability for a color to be drawn from the source is smaller than $\frac{1}{K}$ a color is considered to be a mice. As has been argued in Section 4.3.2, in this case the rate of a color into the online structure is smaller than the rate of a color out of the online structure. In order to test this, a new variable β is defined. Let a color be a mice when the probability for that color to be drawn from the source is smaller than $\frac{1}{\beta K}$. See Figure 7.8 for some results. For comparison it is shown in Figure 7.7 what the behavior is for the parameters used in case there is no offline structure.

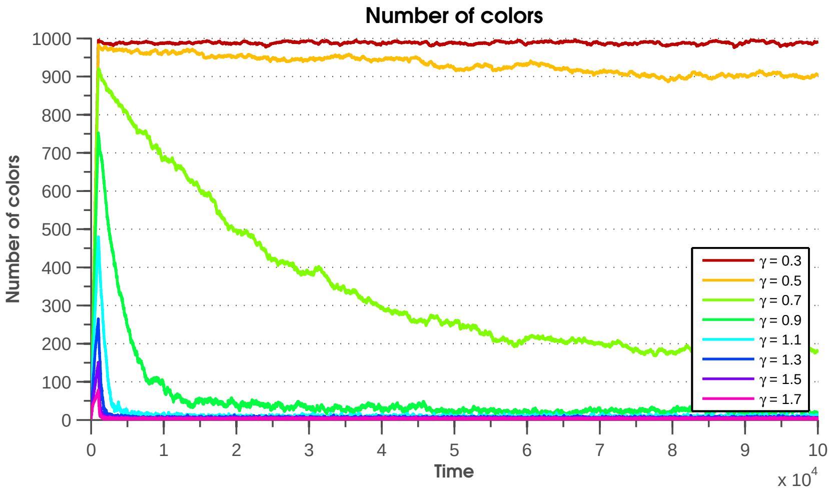
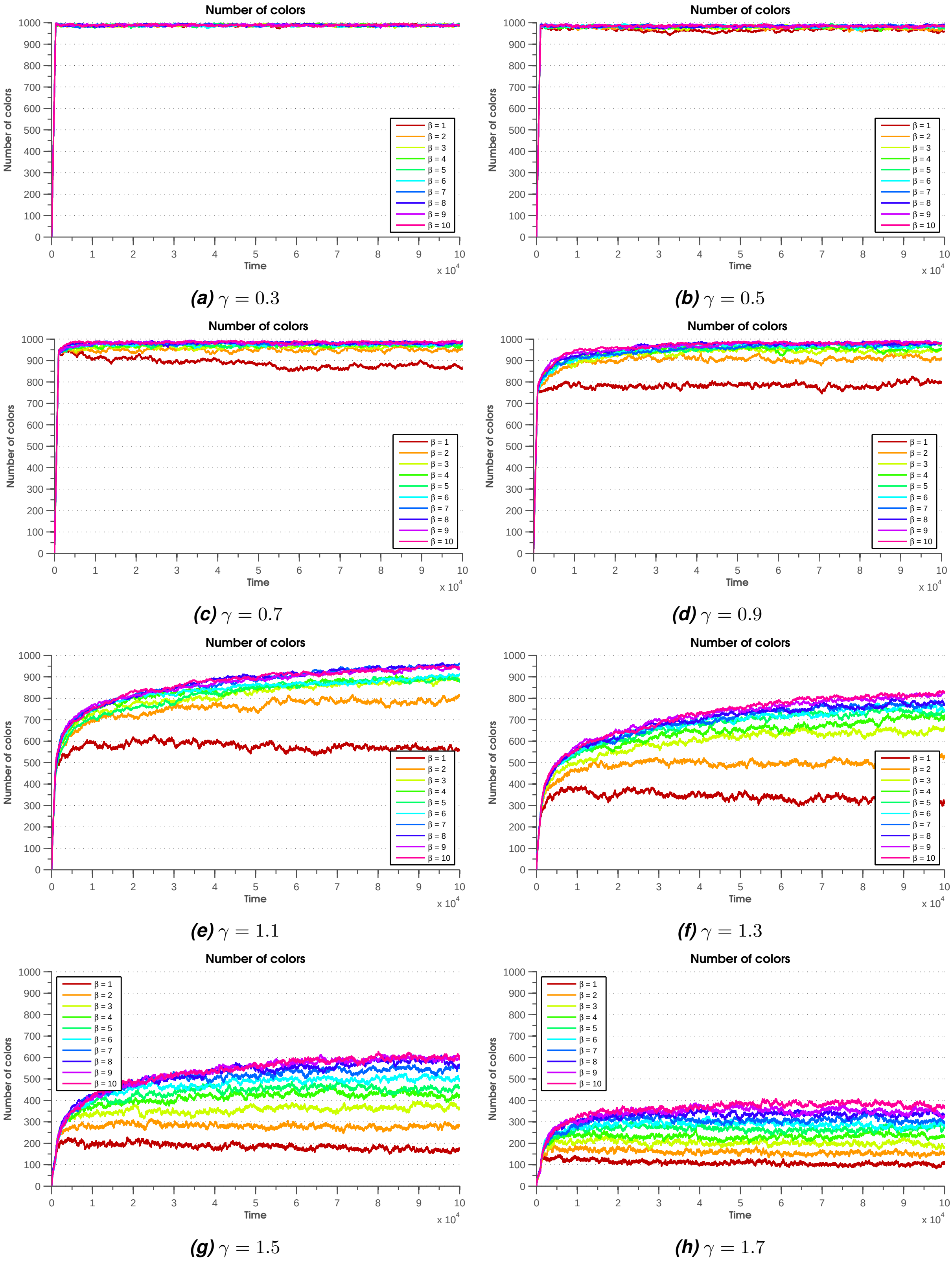


Figure 7.7: *The number of colors in the online structure without the use of the offline structure as a function of the time for different values of* γ*. The total number of colors* $C = 100000$ *and the number of positions in the online structure* $K = 1000$*. The probability to draw a color from the source corresponds to the power law with parameter* γ*.*

Note that the number of colors in the online structure is still decreasing for $\gamma = 0.5$. If $\gamma = 0.3$ the power law distribution is close to a uniform distribution and this explains the stationarity. Numerical results show that the number of colors in the online structure does not decrease until at least $t = 10^8$.



(a) $\gamma = 0.3$

(b) $\gamma = 0.5$

(c) $\gamma = 0.7$

(d) $\gamma = 0.9$

(e) $\gamma = 1.1$

(f) $\gamma = 1.3$

(g) $\gamma = 1.5$

(h) $\gamma = 1.7$

Figure 7.8: *The number of colors in the online structure for several combinations of γ and β. The total number of colors $C = 100000$ and the number of positions in the online structure $K = 1000$.*

The first guess, consider a color to be a mice if the probability for it to be drawn from the source is smaller than $\frac{1}{K}$, does not give a good performance. However, the results do show that with this strategy for the use of the offline structure the number of colors in the online structure is increased, see Figure 7.7 to compare without the use of the offline structure. Note that in some figures there is no convergence yet. In some other figures, for example Figure 7.8g for $\beta = 1$, the number of colors in the online structure is decreasing. This is because among the mice there are still colors that dominate the online structure. Also, Figure 7.8h suggests that it is not possible to increase the number of colors in the online structure by increasing β further. However, this is not true. Increasing β to, for example 200, gives results with at stationarity about 950 different colors in the online structure. In Table 7.1 and 7.2 the number of elephant colors c and their corresponding probability $1 - q$ are shown for the same parameters as in Figure 7.8.

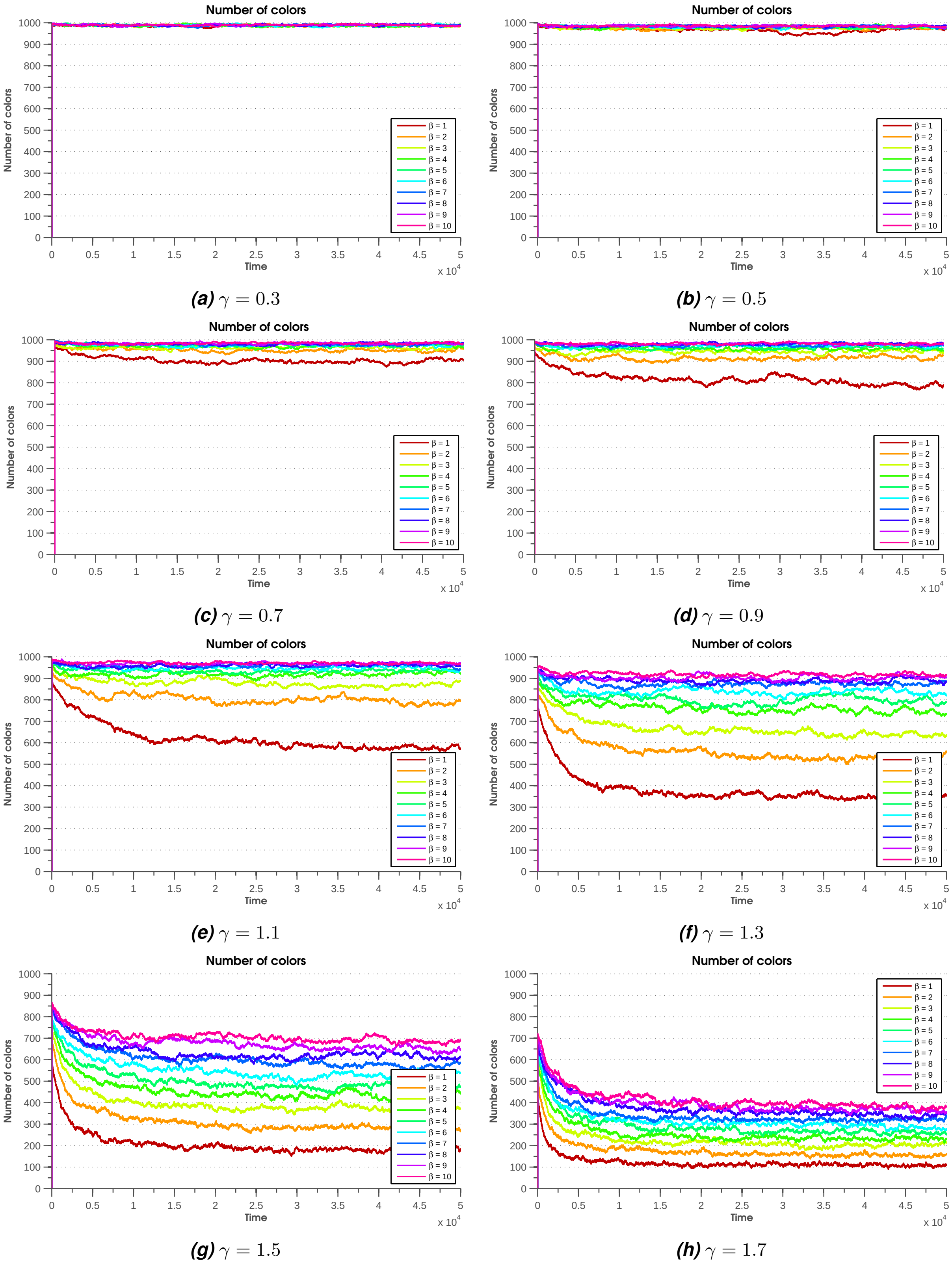
Table 7.1: *The number of elephant colors, c, for combinations of γ and β. The total number of colors $C = 100000$ and the number of positions in the online structure $K = 1000$.*

		γ							
		0.3	0.5	0.7	0.9	1.1	1.3	1.5	1.7
β	1	0	2	25	68	86	72	52	38
	2	0	10	69	148	162	123	83	57
	3	0	22	124	233	234	168	109	72
	4	0	40	187	320	304	210	133	86
	5	1	62	257	411	372	249	154	98
	6	2	90	334	503	439	287	174	109
	7	4	123	416	597	506	323	193	119
	8	6	160	504	693	571	358	211	129
	9	9	203	596	790	635	392	228	138
	10	14	251	693	888	699	425	245	147

Table 7.2: *The probability of an elephant color to be drawn from the source, $1 - q$, for combinations of γ and β. The total number of colors $C = 100000$ and the number of positions in the online structure $K = 1000$.*

		γ							
		0.3	0.5	0.7	0.9	1.1	1.3	1.5	1.7
β	1	0	0.0027	0.0587	0.2627	0.5635	0.7866	0.8965	0.9462
	2	0	0.0080	0.0889	0.3180	0.6162	0.8222	0.9184	0.9594
	3	0	0.0127	0.1110	0.3524	0.6454	0.8404	0.9291	0.9655
	4	0	0.0179	0.1291	0.3774	0.6655	0.8525	0.9360	0.9696
	5	2.2140e-04	0.0227	0.1447	0.3978	0.6807	0.8612	0.9407	0.9722
	6	4.0124e-04	0.0278	0.1587	0.4145	0.6929	0.8681	0.9443	0.9742
	7	7.0655e-04	0.0329	0.1713	0.4290	0.7033	0.8737	0.9473	0.9758
	8	9.7250e-04	0.0378	0.1830	0.4418	0.7119	0.8784	0.9497	0.9771
	9	0.0013	0.0429	0.1939	0.4533	0.7195	0.8824	0.9517	0.9782
	10	0.0019	0.0480	0.2041	0.4636	0.7262	0.8858	0.9534	0.9791

One would expect that the stationary behavior of the online structure is not influenced by the number of colors after K steps, the initialization. Another possibility for the initialization is to only accept one ball of the elephant colors and if another ball is drawn with the same elephant color it goes to the offline structure. With this other initialization the elephant colors also can not dominate the online structure during the initialization, while with the previous strategy they could. The next figure shows the behavior. Note that due to the parameters chosen the number of elephant colors and the probability of an elephant color to be drawn from the source can be found in Table 7.1 and Table 7.2 respectively.



(a) $\gamma = 0.3$

(b) $\gamma = 0.5$

(c) $\gamma = 0.7$

(d) $\gamma = 0.9$

(e) $\gamma = 1.1$

(f) $\gamma = 1.3$

(g) $\gamma = 1.5$

(h) $\gamma = 1.7$

Figure 7.9: *The number of colors in the online structure for several combinations of* γ *and* β*. The total number of colors* $C = 100000$ *and the number of positions in the online structure* $K = 1000$*.*

The expectation is indeed right that in stationarity the behavior is similar, see Figure 7.8 for comparison. However it takes less steps to reach stationarity and therefore on average the number of colors in the online structure is higher when using the new initialization. The reason for this is that the elephant colors do not get a chance to dominate the online structure, and hence stationarity is achieved in a smaller number of steps. An advantage of the decrease of the number of steps is that the computations to reach stationarity take less time. A disadvantage is the use in practice, because then it is not known what the probability is for a color to be drawn from the source and so it is not known on beforehand which color is considered to be an elephant color. The remedy for this is discussed in Chapter 8.

7.4.1 Analysis on the Mice and Elephant Strategy

The elephant colors, colors $1, ..., c$, have a total probability of $1 - q$ to be drawn from the source and the mice, colors $c + 1, .., C$, have a total probability of q to be drawn from the source. Let η_1 be the number of colors in the online structure within the elephants and let η_2 be the number of colors in the online structure from the mice. Then because of stationarity:

$$\eta_2 - \frac{\eta_2}{\eta_1 + \eta_2} \frac{\alpha \eta_2}{\eta_2} + q^* \quad = \quad \eta_2,$$

where α is the fraction of unique colors in the online structure among the mice and q^* is the probability to draw a ball that has a color, corresponding to a mice, that is not present in the online structure. Then it follows that

$$\begin{aligned} \frac{\alpha \eta_2}{\eta_1 + \eta_2} &= q^*, \\ \eta_2 &= \frac{\eta_1 q^*}{\alpha - q^*} \leq \frac{c q^*}{\alpha - q^*} \leq \frac{c q}{\alpha - q}, \end{aligned}$$

where the first inequality follows because it could happen that not all the elephants are present in the online structure. Unfortunately it is not possible to get expressions for α.

Note the analogy of $\frac{q^*}{\alpha - q^*}$ with the mean number of customers in the system of an $M|M|1$ queue, $\frac{\lambda}{\mu - \lambda}$. In the case studied here α is the departure rate of unique "mice" queues and q^* is the arrival rate of unique "mice" queues. To have a stable queue, in the latter case it should hold that $\lambda < \mu$. This explains why $q^* < \alpha$.

Chapter 8

Strategy in practice

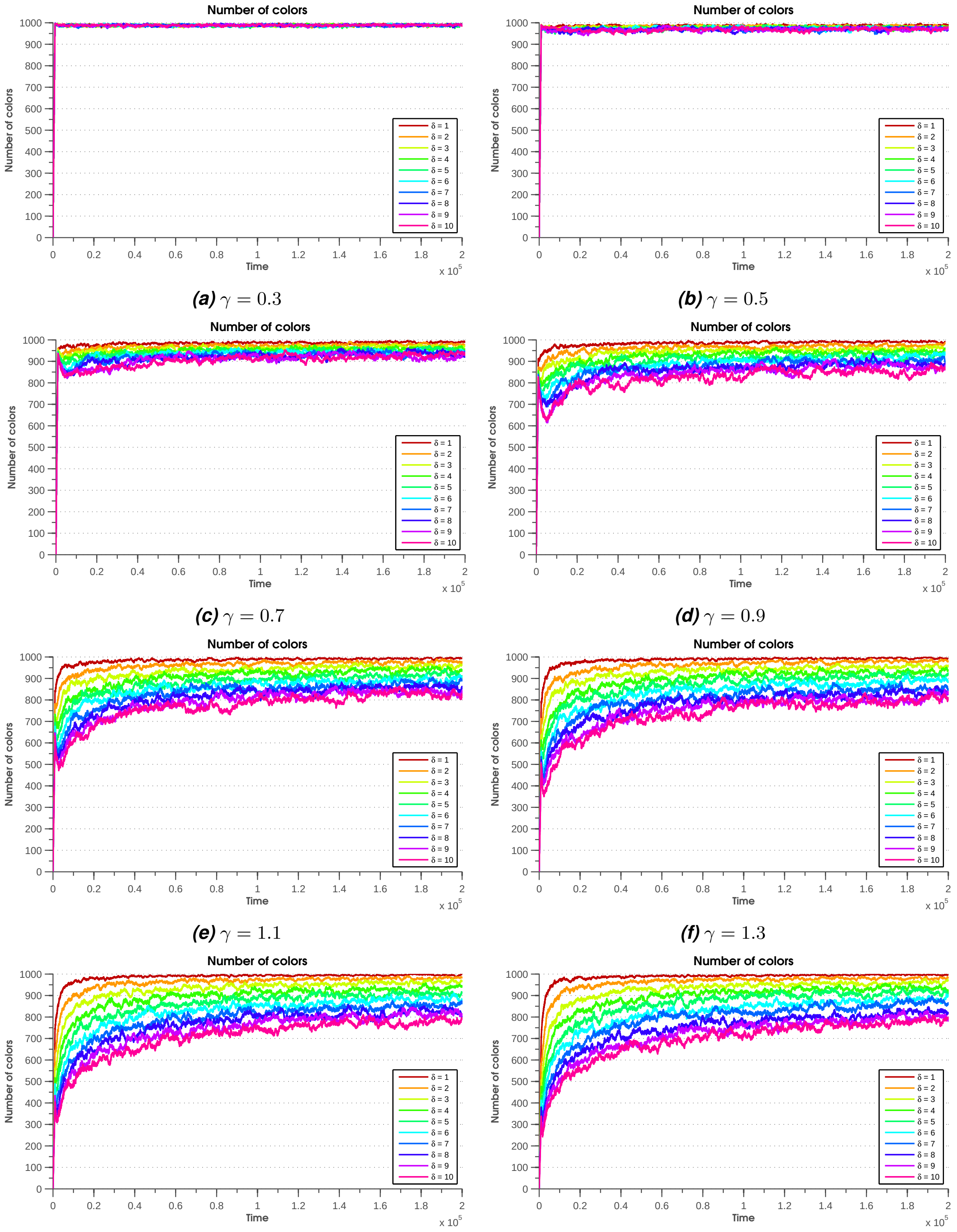
In Chapter 7 several strategies for the use of the offline structure have been tried. The mice and elephant strategy worked best and therefore in this chapter possibilities for the implementation in practice are discussed.

8.1 Parameters Used

Here we introduce realistic parameters. The number of positions in the online structure K largely depends on the machine that is used, but is in the range of $[10^7, 10^8]$. The total number of colors in the biggest crawl performed this far is of order 10^7, see Chapter 6. In this chapter also the distribution for a color to be drawn from the source has been discussed, the power law distribution. Although a distribution is known, in practice it is not known for example which color has which probability to be drawn from the source and it is also not known on beforehand how many colors there are.

To start with, the assumption still holds that there are C colors. To decrease the computation time of the simulations, the total number of colors and the number of positions in the online structure do not have realistic values for the practical use and in this chapter they are chosen to be 100000 and 1000 respectively.

Our goal now is to implement the mice and elephant strategy when the probability distribution of colors is not known on beforehand. In other words, the implementation that works good in theory has to be adapted for the use in practice. This leads to the following version of the mice and elephant strategy. Suppose now that a color is considered to be an elephant color when the number of balls of this color in the online structurer exceeds δ. This strategy is also used in the first K steps, the initialization, to decrease the number of steps towards stationarity. The strategy remains the same as in Algorithm 11, only the definition of mice and elephant colors has changed. The results are shown in Figure 8.1.



(a) $\gamma = 0.3$

(b) $\gamma = 0.5$

(c) $\gamma = 0.7$

(d) $\gamma = 0.9$

(e) $\gamma = 1.1$

(f) $\gamma = 1.3$

(g) $\gamma = 1.5$

(h) $\gamma = 1.7$

Figure 8.1: *The number of colors in the online structure for several combinations of* γ *and* δ*. The total number of colors* $C = 100000$ *and the number of positions in the online structure* $K = 1000$*.*

The number of elephant colors that corresponds to the figures above can be found in Table 8.1. The strategy is working robust. Of course the number of elephant colors is decreasing when δ is increasing. Also, it does not come as a surprise that when $\delta = 1$ the number of colors in the online structure is almost equal to K. This is due to the number of elephant colors. The strategy when $\delta = 1$ is almost the same as the greedy strategy where balls were only accepted when they have a color that is not present in the online structure. A difference is that when a ball with an elephant color is removed from the online structure, most likely a ball of this color is available in the offline structure since every elephant color has it's own queue in the offline structure.

Table 8.1: *The number of elephant colors, c, for combinations of γ and δ. The total number of colors $C = 100000$ and the number of positions in the online structure $K = 1000$.*

		γ							
		0.3	0.5	0.7	0.9	1.1	1.3	1.5	1.7
δ	10	0	4	39	103	183	259	322	377
	9	0	3	37	107	189	273	341	397
	8	0	4	51	120	213	294	380	421
	7	0	5	55	141	244	326	401	472
	6	0	9	70	164	275	380	451	524
	5	0	14	72	202	326	433	534	588
	4	0	26	114	270	411	539	625	693
	3	2	40	199	421	586	712	804	876
	2	40	173	500	798	1027	1188	1231	1269
	1	2162	2711	3026	3133	3034	2832	2613	2487

Note that this adapted strategy does not give the same number of elephant colors every time a simulation is done. Therefore the probability of an elephant color, $1 - q$, also changes. Table 8.2 gives the probability for the elephant colors to be drawn from the source. In Table 8.3 an upper bound for the probability of an elephant color can be found i.e., it is assumed that the elephant colors are the c colors with highest probability to be drawn from the source.

Table 8.2: *The probability of an elephant color to be drawn from the source, $1 - q$, for combinations of γ and δ. The total number of colors $C = 100000$ and the number of positions in the online structure $K = 1000$.*

		γ							
		0.3	0.5	0.7	0.9	1.1	1.3	1.5	1.7
δ	10	0	0.0041	0.0702	0.2915	0.6246	0.8626	0.9594	0.9892
	9	0	0.0036	0.0684	0.2931	0.6273	0.8652	0.9607	0.9896
	8	0	0.0043	0.0779	0.3013	0.6368	0.8686	0.9628	0.9900
	7	0	0.0050	0.0805	0.3121	0.6470	0.8733	0.9637	0.9908
	6	0	0.0073	0.0874	0.3240	0.6561	0.8800	0.9660	0.9914
	5	0	0.0089	0.0885	0.3381	0.6689	0.8854	0.9687	0.9921
	4	0	0.0130	0.1043	0.3602	0.6853	0.8939	0.9712	0.9929
	3	1.3966e-04	0.0159	0.1255	0.3928	0.7088	0.9039	0.9748	0.9940
	2	0.0017	0.0320	0.1698	0.4393	0.7448	0.9209	0.9797	0.9953
	1	0.0358	0.1116	0.2842	0.5498	0.8061	0.9422	0.9862	0.9970

Table 8.3: *An upper bound on the probability of an elephant color to be drawn from the source, $1-q$, for combinations of γ and δ. The total number of colors $C = 100000$ and the number of positions in the online structure $K = 1000$.*

		γ							
		0.3	0.5	0.7	0.9	1.1	1.3	1.5	1.7
δ	10	0	0.0044	0.0708	0.2917	0.6260	0.8632	0.9597	0.9893
	9	0	0.0036	0.0693	0.2944	0.6286	0.8657	0.9609	0.9897
	8	0	0.0044	0.0789	0.3027	0.6381	0.8693	0.9631	0.9901
	7	0	0.0051	0.0813	0.3145	0.6487	0.8741	0.9641	0.9909
	6	0	0.0075	0.0894	0.3257	0.6579	0.8810	0.9663	0.9915
	5	0	0.0098	0.0903	0.3414	0.6708	0.8866	0.9692	0.9922
	4	0	0.0140	0.1076	0.3640	0.6881	0.8956	0.9717	0.9931
	3	4.0124e-04	0.0179	0.1320	0.3997	0.7138	0.9061	0.9754	0.9942
	2	0.0040	0.0394	0.1825	0.4541	0.7526	0.9234	0.9806	0.9955
	1	0.0681	0.1627	0.3326	0.5829	0.8217	0.9473	0.9874	0.9973

The upper bound of the probability of an elephant color is getting better for larger values of δ and γ. But it can be seen from these data that it is likely the c colors with highest probability to be drawn from the source are the elephant colors.

Comparing the results of the new definition of an elephant color with the results of the previous definition in Figure 7.8 leads to the following conclusions. Note that it is only possible to compare the solutions when the number of elephant colors are about the same. If this is the case, both methods give similar behavior in stationarity, see for example for the parameters $\gamma = 1.3$, $\delta = 5$ and $\beta = 10$. This result makes sense, since a similar amount of colors is put in the offline structure. Moreover, most of the elephant colors will be the same colors.

One observation about this strategy is that the number of elephant colors changes when the total number of steps performed in a simulation, denoted by T, changes. In Table 8.4 the number of elephant colors is shown for different values of T. For every value of T two simulations are done, also to show that the number of elephant colors is not necessarily the same in every simulation. For γ the value 0.9 has been chosen. The reason for this is that with this value for γ there is a significant amount of elephant colors, while the computation time is much lower than for example with $\gamma = 1.7$. This makes sense because in the latter case the probability of drawing an elephant color from the source is close to 1 and therefore it is computationally demanding to search for a ball with a color that belongs to the mice.

Table 8.4: *The number of elephant colors, c, for combinations of δ and T. The total number of colors $C = 100000$, the number of positions in the online structure $K = 1000$ and $\gamma = 0.9$.*

		T					
		100000		200000		300000	
		Simulation 1	Simulation 2	Simulation 1	Simulation 2	Simulation 1	Simulation 2
δ	10	91	91	103	102	108	112
	9	97	93	111	116	121	124
	8	108	101	119	133	130	137
	7	121	119	145	135	162	151
	6	129	139	168	164	176	180
	5	169	158	196	195	216	222
	4	210	214	264	265	289	303
	3	315	317	409	382	450	477
	2	579	607	795	786	912	927
	1	2100	2082	3112	3093	4006	3993

The table shows that indeed the number of elephant colors increases when T increases. This makes sense, because every color that happens to have δ colors in the online structure is considered to be an elephant color. Therefore it would make sense to search for elephant colors for a fixed amount of steps and afterwards use these elephant colors. The fixed amount is chosen in such a way that afterwards all the dominating colors are part of the elephant colors. Before this is implemented in a final strategy, first it is checked whether it is realistic to give every elephant color it's own queue in the offline structure.

8.2 Use of Offline Structure

At this moment, every elephant color has it's own queue in the offline structure. To see whether it is possible to implement this in practice, we give a short explanation of how the offline structure is used in practice.

Every queue is a file and every time a queue is accessed it needs to be opened:

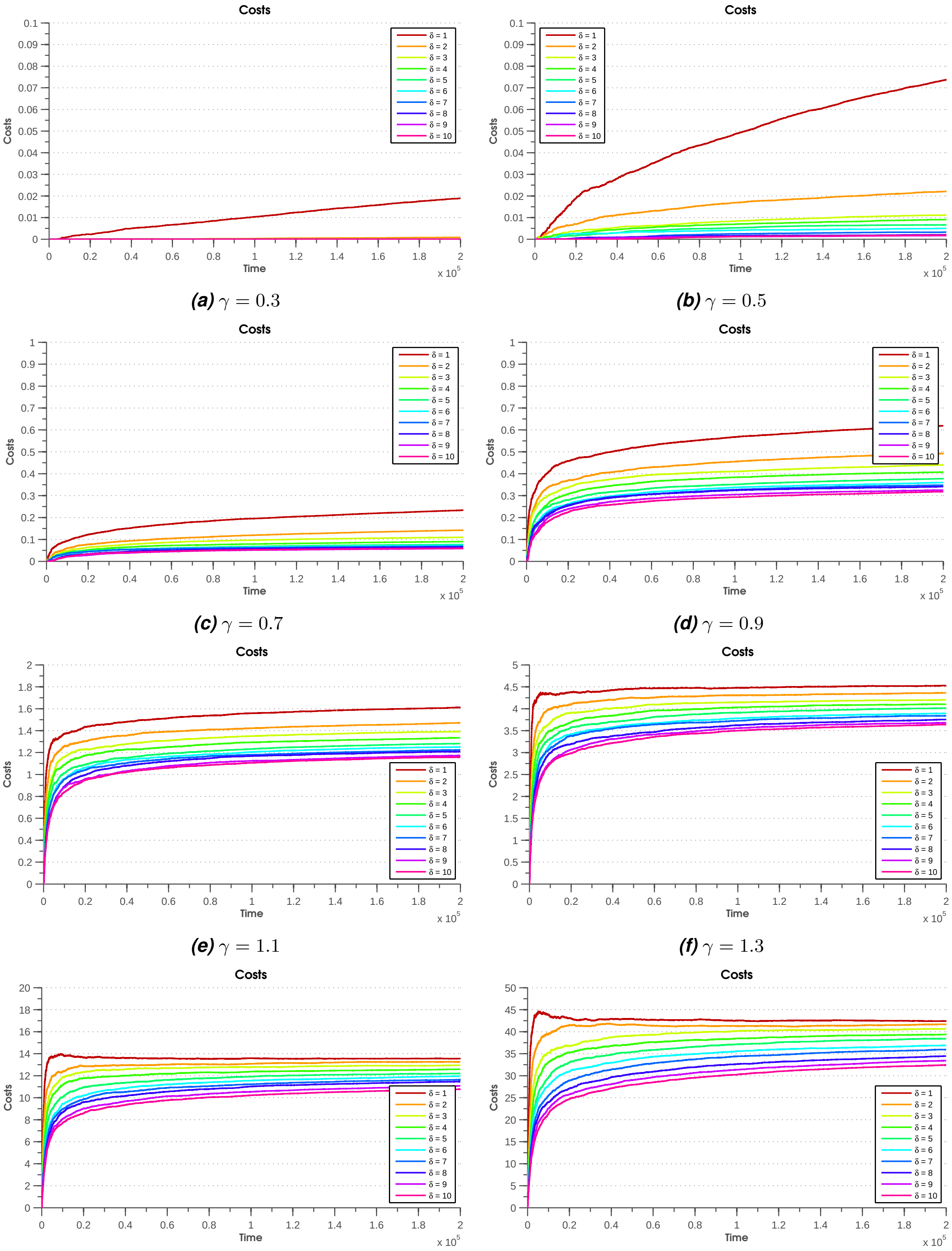
- either the queue is opened when it is needed and then it can be closed when it is not needed anymore. Both operations are very costly. Or,
- the queue remains open, but there is a limit on the number of open files. This limit is in the order of 10^2 or 10^3 at most.

Every time a ball goes to the end of the queue the ball needs to be written at the end of the queue. A disk head has to move to the end of the queue physically. If the balls are put at the end of the same queue, this is not a problem. As soon as the writing has to be done at the end of many queues in an interleaved way, the head starts to move back and forth which is extremely costly.

In order to measure how often the head has to move back and forth when every color has it's own queue in the offline structure, we make the following calculation. Suppose that a variable $m(t)$ indicates how many times the head has to move back and forth in the interval $[0, t]$. It is assumed that all files remain open, although in practice there is a limit. Then the costs per time unit at time t are calculated by $\frac{m(t)}{t}$. The results in Figure 8.2 are calculated during the same simulation as for the results in Figure 8.1.

When the costs are smaller than 1 this means that on average in every step the head has to move at most once. If the costs are greater than 1 during every step the head has to move at least once on average. This can happen because all the elephant colors that are drawn from the source need to be put at their places. Notice that when $\gamma < 1.5$ there is not yet a convergence in the expected number of costs, but most likely this happens when there are more steps performed in the simulation. When $\gamma \geq 1.5$ there is a convergence in the costs. Still, the costs are high and the head has to move about 14 times per step on average when $\gamma = 1.5$. This is very often and since it is a costly operation, it is tested what happens when there is just one queue in the offline structure.

Figure 8.3 shows the differences between giving every elephant color it's own queue and having just one queue in the offline structure. For the first case figures are shown from Figure 8.1, they are repeated for comparison.



(a) $\gamma = 0.3$

(b) $\gamma = 0.5$

(c) $\gamma = 0.7$

(d) $\gamma = 0.9$

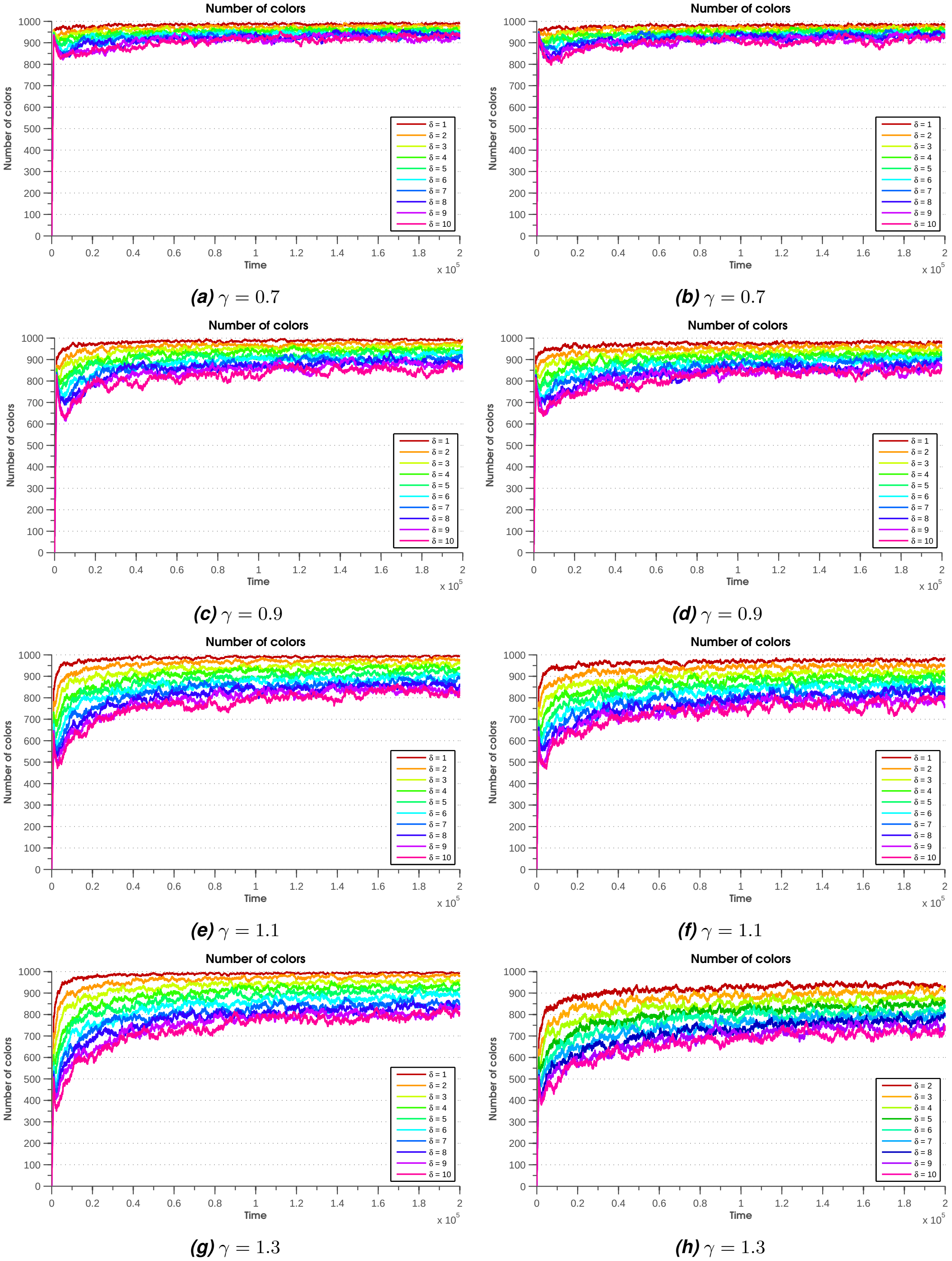
(e) $\gamma = 1.1$

(f) $\gamma = 1.3$

(g) $\gamma = 1.5$

(h) $\gamma = 1.7$

Figure 8.2: *The costs of moving the disk head back and forth for several combinations of* γ *and* δ. *The total number of colors* $C = 100000$ *and the number of positions in the online structure* $K = 1000$.



(a) $\gamma = 0.7$

(b) $\gamma = 0.7$

(c) $\gamma = 0.9$

(d) $\gamma = 0.9$

(e) $\gamma = 1.1$

(f) $\gamma = 1.1$

(g) $\gamma = 1.3$

(h) $\gamma = 1.3$

Figure 8.3: *Comparison of the number of colors in the online structure for several combinations of* γ *and* δ *when using a single queue for every elephant color in the offline structure (left) and when using one queue in the offline structure for all elephant colors (right). The total number of colors* $C = 100000$ *and the number of positions in the online structure* $K = 1000$.

Figure 8.3 shows that in case there is just one queue in the offline structure the number of colors in the online structure does not significantly decrease. The number of elephant colors in case there is one queue in the offline structure is shown in Table 8.5. Note that there is no simulation performed for the combination $\gamma = 1.3$ and $\delta = 1$ because of computation time. Computation for the combination $\gamma = 1.3$ and $\delta = 2$ took 13 hours and since the number of dominating colors is even higher for $\delta = 1$ this computation also requires a lot more time. Moreover, in case $\delta = 1$ every color is an elephant color once it has appeared in the online structure and when a color appears in the online structure once does not imply that the color is an elephant color. Therefore from this moment onwards $\delta = 1$ is omitted. Since the costs in case every elephant color has it's own queue are rather high, as has been shown in Figure 8.2, in practice preference is given to one queue in the offline structure.

Table 8.5: *The number of elephant colors, c, for combinations of γ and δ. The total number of colors $C = 100000$ and the number of positions in the online structure $K = 1000$. The offline structure consists of one queue.*

		γ			
		0.7	0.9	1.1	1.3
δ	10	40	114	222	331
	9	41	131	234	364
	8	48	138	264	411
	7	56	164	309	456
	6	71	196	349	535
	5	85	237	437	647
	4	141	329	584	863
	3	217	532	875	1270
	2	589	1128	1733	2429
	1	4335	5699	7084	-

Also in this strategy, when using one queue in the offline structure, when the number of steps that is performed increases also the number of elephant colors increases. In Table 8.6 the number of elephant colors is shown for different values of T. Again for every value of T two simulations are done, also to show that the number of elephant colors is not necessarily the same in every simulation.

Table 8.6: *The number of elephant colors, c, for combinations of δ and T. The total number of colors $C = 100000$, the number of positions in the online structure $K = 1000$ and $\gamma = 0.9$. The offline structure consists of one queue.*

		T					
		100000		200000		300000	
		Simulation 1	Simulation 2	Simulation 1	Simulation 2	Simulation 1	Simulation 2
δ	10	99	99	114	114	122	119
	9	101	104	131	126	143	125
	8	118	116	138	144	163	158
	7	129	137	164	162	175	192
	6	150	155	196	188	202	205
	5	179	187	237	238	276	263
	4	262	253	329	328	367	353
	3	376	371	532	508	607	612
	2	762	768	1128	1100	1389	1423

The tables show that indeed the number of elephant colors increases when T increases. This makes sense, because every color that happens to have δ colors in the online structure is considered to be an elephant color. This number increases when T increases. When taking a closer look at the table it is noted that for $\delta = 9$ it happens that for a simulation the number of elephant colors is smaller for $T = 300000$ then for $T = 200000$. Also, the number of elephant colors is higher when there is just one queue in the offline structure instead of one queue for every elephant color, see Table 8.4 for comparison. The reason for this is that it happens less often that a ball can be accepted from the offline structure, since now there is just one queue in the offline structure. The time span to search for elephant colors is discussed in the next section.

8.3 Time Span to search for Elephant Colors

It is expected that when choosing δ big enough that there will be convergence of the number of elephant colors. To this end, Table 8.7 shows the number of elephant colors for combinations of δ and T, for larger values of T than in Table 8.6.

Table 8.7: *Extension of Table 8.6. The number of elephant colors, c, for combinations of δ and T. The total number of colors $C = 100000$, the number of positions in the online structure $K = 1000$ and $\gamma = 0.9$. The offline structure consists of one queue.*

		T					
		400000		500000		600000	
δ	10	129	138	137	141	151	141
	9	137	139	156	156	158	155
	8	167	166	178	171	176	183
	7	192	190	198	195	201	215
	6	233	225	244	244	254	263
	5	283	302	301	314	321	334
	4	421	420	444	436	474	450
	3	683	676	753	743	767	799
	2	1618	1605	1756	1805	1982	1946

In Table 8.7 it can be seen that it happens more often that a longer simulation run does not necessarily yields more dominating colors and this supports the thought of convergence of the number of elephant colors. Since these are all separate simulations, two simulations are performed with total length of $T = 500000$ and every 50000 steps the number of elephant colors is shown. This can be found in Table 8.8 and Table 8.9.

Table 8.8: *The number of elephant colors, c, for combinations of δ and T during a simulation. The total number of colors $C = 100000$, the number of positions in the online structure $K = 1000$ and $\gamma = 0.9$. The offline structure consists of one queue.*

		$T \cdot 10^3$									
		50	100	150	200	250	300	350	400	450	500
δ	10	77	95	105	114	120	123	130	133	139	139
	9	87	106	121	130	136	140	145	153	156	159
	8	95	120	132	141	150	160	163	167	172	175
	7	105	128	146	154	170	181	190	195	200	206
	6	124	153	174	190	200	213	225	234	238	242
	5	144	187	213	236	254	268	278	295	304	317
	4	181	254	290	325	362	372	388	408	426	436
	3	281	377	447	513	562	620	666	707	734	766
	2	542	791	961	1116	1253	1385	1491	1589	1691	1770

Table 8.9: *The number of elephant colors, c, for combinations of δ and T during a simulation. The total number of colors $C = 100000$, the number of positions in the online structure $K = 1000$ and $\gamma = 0.9$. The offline structure consists of one queue.*

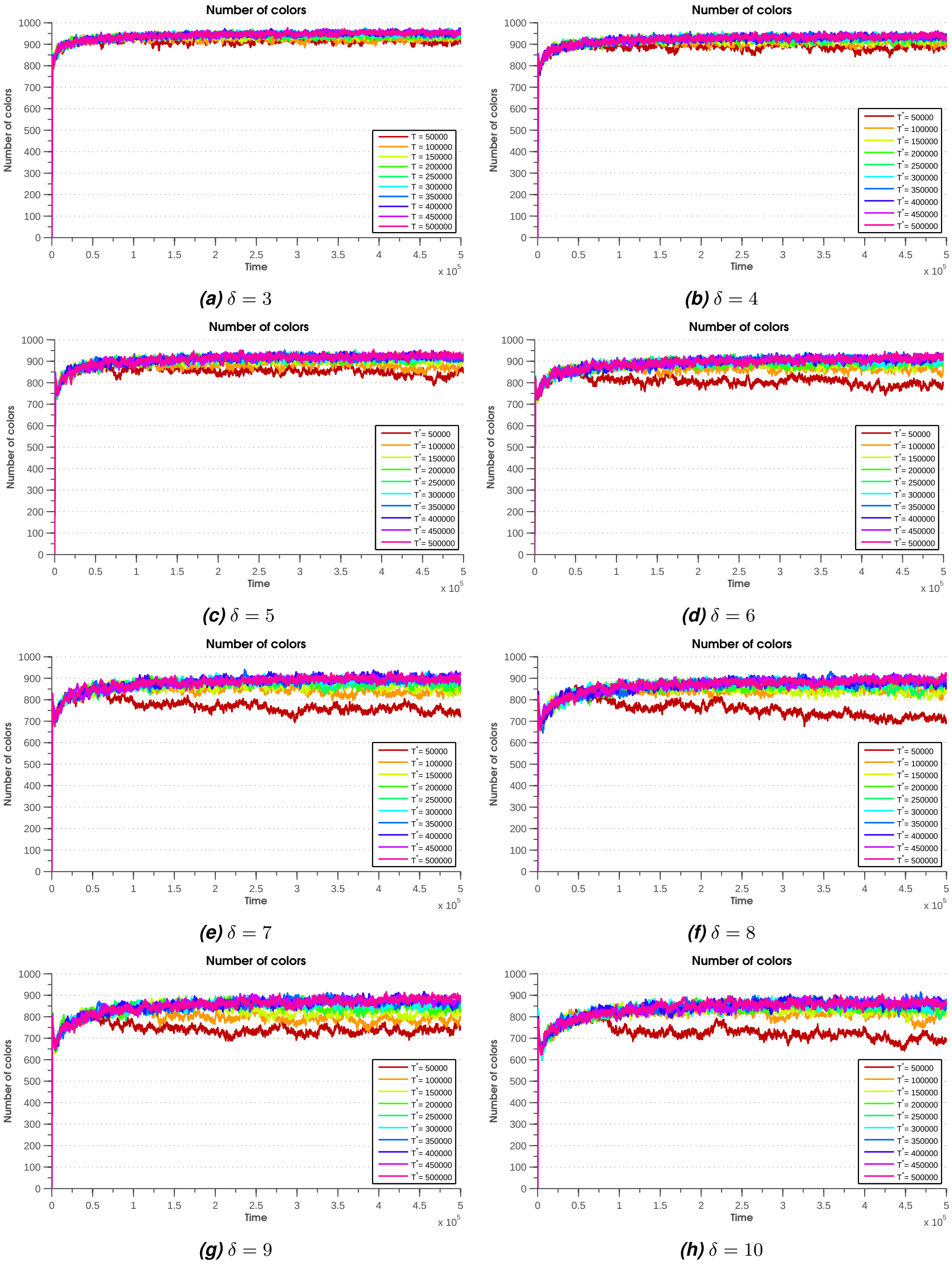
		$T \cdot 10^3$									
		50	100	150	200	250	300	350	400	450	500
δ	10	77	90	103	109	113	125	130	132	137	140
	9	86	107	117	132	139	145	152	154	158	160
	8	89	115	129	142	147	152	159	166	173	180
	7	104	133	153	173	181	188	192	197	205	209
	6	129	155	173	183	193	202	215	223	234	237
	5	162	199	232	251	266	279	289	296	304	310
	4	191	247	292	320	348	378	404	419	429	444
	3	292	390	460	507	552	587	630	662	688	715
	2	548	761	956	1124	1250	1368	1483	1582	1681	1776

The table shows that the number of elephant colors keeps increasing, although the amount in which it is increasing does decrease. Note that since the number of elephant colors keeps increasing it might happen that at some point in time (almost) all the colors are considered to be an elephant color and when having one queue in the offline structure it is not possible anymore to continue. There is one example in which there is no increase of the number of elephant colors, $\delta = 10$ from $T = 450000$ to $T = 500000$ in Table 8.8, but this does not necessarily imply that there is convergence. Therefore, suppose for a certain amount of time T^* a ball is considered to be an elephant color when the number of balls of this color exceeds δ. After this time T^* no colors are added to the list of elephant colors and it is assumed that from this point onwards the colors with a higher probability to de drawn from the source are already added to the list of elephant colors. The results are presented in Figure 8.4, the number of elephant colors can be found in Table 8.10.

Table 8.10: *The number of elephant colors, c, for combinations of δ and T^*. The total number of colors $C = 100000$, the number of positions in the online structure $K = 1000$ and $\gamma = 0.9$. The offline structure consists of one queue and the total number of steps performed $T = 500000$.*

		$T^* \cdot 10^3$									
		50	100	150	200	250	300	350	400	450	500
δ	10	72	100	108	119	116	125	132	134	140	144
	9	81	102	116	128	132	133	141	149	152	154
	8	92	115	132	145	144	154	166	154	177	176
	7	104	123	144	154	168	180	183	199	190	195
	6	126	152	170	180	201	204	238	227	237	245
	5	157	185	221	235	253	268	281	291	301	312
	4	201	249	291	329	343	370	378	413	429	428
	3	296	377	460	510	531	596	646	693	719	728

The results in Figure 8.4 show that T^* needs to be bigger when δ increases, but in general when T^* increases the number of colors in the online structure does not change. The number of elephant colors also show this behavior, because the number of elephant colors can be lower for higher values of T^* than for smaller values of T^*, see for example for $\delta = 10$.



(a) $\delta = 3$

(b) $\delta = 4$

(c) $\delta = 5$

(d) $\delta = 6$

(e) $\delta = 7$

(f) $\delta = 8$

(g) $\delta = 9$

(h) $\delta = 10$

Figure 8.4: *Comparison of the number of colors in the online structure for several combinations of* δ *and* T^**. The total number of colors* $C = 100000$ *and the number of positions in the online structure* $K = 1000$*.*

In order to give a rule of thumb for the time to search for elephant colors, T^* , more research has to be done. Still, it can be said that if after a reasonable amount of steps in the crawling process the number of colors in the online structure does not change, no search for more elephant colors has to be done. It it turns out that the number of colors in the online structure does decrease, some elephant colors can be added to the list of elephant colors. If not, the crawling process can be continued without more searching for elephant colors.

Before introducing the final strategy a value for δ is chosen. Based on all the simulations performed, $\delta = 5$ seems like a proper choice. The motivation for this choice is the following. Choosing $\delta = 1$ is no option, because then every color is an elephant color once it has been in the online structure only once. This is not a proper definition for an elephant color. The same reasoning holds for $\delta = 2$ and also with $\delta = 3$ there is not much fluctuation possible in the number of balls of a certain color in the online structure. For all the examples that are shown it is clear that the number of elephant colors is significantly lower with $\delta = 5$ instead of $\delta = 4$ and it also gives a bit more space for fluctuations for the number of colors in the online structure before a color is considered to be an elephant color. Still, for $\delta = 5$ the results are good and therefore this seems a proper choice as a rule of thumb for when a color is considered to be an elephant color. This choice for δ is implemented in the final strategy.

8.4 Recommended Strategy

In the previous sections the final strategy has been designed and adapted, and in this section this is summarized. The final strategy takes practical issues into account, such as a disk head moving back and forth when different queues in the offline structure are considered and the fact that no probability distribution is known for the total number of colors. The offline structure consists of one queue. The recommended strategy can be found in Algorithm 12.

Algorithm 12 Recommended Strategy

```
if the first ball of the offline structure has a color that is not present in the online structure then
    accept this ball to the online structure.
else
    while no ball has been added to the online structure do
        Draw a ball from the source.
        if this ball has a color that belongs to the mice then
            accept the ball to the online structure.
        else
            put the ball in the offline structure.
        end if
    end while
    while the number of colors in the online structure is not stable do
        if from a certain color there are $5$ balls in the online structure then
            this color belongs to the elephants.
        end if
    end while
end if
```

If the search for elephant colors has quit and the number of colors in the online structure decreases significantly, start adding color to the elephants again.

A possible adaptation to this strategy would be the following: if a ball has an elephant color and it is neither in the online structure nor in the offline structure, accept this ball to the online structure. This adaptation could slightly change the results presented in this thesis, but not considerably.

Chapter 9

Conclusions and Discussion

9.1 Conclusions

In this thesis, the decision process for the BUbiNG web crawler has been optimized. First the web crawler has been studied without the workbench virtualizer, this is taken as a reference frame. In this case it is shown analytically that the throughput of the web crawler can be maximized if the distribution of the hosts the URLs belong to corresponds to a uniform distribution. Note that this is only the case when the total number of hosts is significantly greater than the number of positions in the workbench. This result means that if the probability to draw a URL from a certain host from the sieve corresponds to a uniform distribution, the workbench virtualizer is not necessary at all in order to maximize the web crawler's throughput. In practice, the probability to draw a URL from a certain host from the sieve does not correspond to a uniform distribution but it behaves more like a power law distribution. For the remainder of the conclusions it is assumed that the probability to draw a URL from a certain host corresponds to a power law distribution.

It has been shown analytically that large hosts represent a problem. These hosts dominate the workbench and therefore the number of different hosts in the workbench remains low. This is what the designers of the web crawler also observed in practice.

After these analyses, several strategies have been tested numerically for the decision policies of the web crawler. Natural strategies did not work, because the workbench virtualizer is not always useful. For example in the greedy strategy, see Algorithm 4, after a finite time the workbench virtualizer contains so many hosts that it is computationally hard or even impossible to draw a host from the sieve that is not present in the workbench.
Another possible strategy that has been tested is randomization, i.e., with probability r a URL from the sieve is accepted and with probability $1 - r$ a URL from the workbench virtualizer is accepted. Using this strategy no URL goes to the workbench virtualizer and therefore the strategy does not make any difference compared to the case without the use of the workbench virtualizer.
Several variations on these strategies are tested as well, but none of them resulted in an improvement in throughput for the BUbiNG web crawler.

The mice and elephant principle that occurs in telecommunications leads to a strategy that improves the throughput of the web crawler. The main idea of this strategy is that hosts that are drawn frequently from the sieve are seen as the elephants, while the hosts that are drawn less frequently from the sieve are considered to be mice. Suppose that a URL is accepted to the workbench when the host belongs to the mice and that every elephant host has it's own queue in the offline structure. Thus, when a URL with an elephant host is drawn from the source this URL goes to the workbench virtualizer and therefore this host can not dominate the workbench.

Several variations of the mice and elephant strategy have been tested. The recommended variation of this strategy in order to optimize the decision process of the BUbiNG web crawler is presented in

Algorithm 12. Using this strategy, the throughput of the web crawler is about 90% of the maximum throughput. Note that this depends on the parameters used.

9.2 Discussion

In this thesis the assumption has been made that the total number of hosts is finite. This seems like a realistic assumption, because in practice the number of web pages is finite.

Analytical results are obtained in case there is no workbench virtualizer. The main assumption for the analytical results is that the hosts are homogeneous. For future research, expressions can be derived for stationary behavior when the distribution of hosts to be drawn from the sieve is heterogeneous. Furthermore it might be possible to prove that without the use of the workbench virtualizer the number of distinct hosts in the workbench is maximized when the probability for each host to be drawn from the sieve is homogeneous.
The strategies for the use of the workbench virtualizer have been implemented in this thesis numerically. For future research possibly analytical results can be obtained to support the numerical results.

It is assumed that in practice the probability to draw a host from the sieve is power law distributed. The data set available suggests that the distribution in practice behaves similar as the power law distribution: there are a couple of hosts with a high probability to be drawn from the source, while other hosts have a small probability to be drawn from the source. Still, this might need further research.

The recommended strategy uses the time span to search for elephant hosts, T^*. However, in practice at some point in time a host is completely crawled and a new large host can appear. Instead of T^* we may use a different mechanism.

In this thesis the implementation of the mice and elephant strategy is not addressed, but we developed an approach to identify the problems and foresee the performance of strategies in practice.

Chapter 10

References

[1] P. Boldi, B. Codenotti, M. Santini, and S. Vigna. UbiCrawler: A Scalable Fully Distributed Web Crawler. *Softw. Pract. Exper.*, 34(8):711–726, 2004. doi: 10.1002/spe.587.

[2] P. Boldi, A. Marino, M. Santini, and S. Vigna. BUbiNG: Massive Crawling for the Masses. *Submitted for publication*, 2013.

[3] N. Brownlee and K. Claffy. Understanding Internet Traffic Streams: Dragonflies and Tortoises. *IEEE Communications*, 40 No. 10, Oct 2002:110–117, 2002. doi: 10.

[4] M. Crovella, R. Frangioso, and M. Harchol-Balter. Connection scheduling in web servers. In *Proceedings of the USENIX Symposium on Internet Technologies and Systems (USITS'99)*, 1999.

[5] A. Gnedin, B. Hansen, and J. Pitman. Notes on the occupancy problem with infinitely many boxes: general asymptotics and power laws. *Probability Surveys*, 4:146–171, 2007. doi: 10.1214/07-PS092.

[6] S. Goldberg. A Direct Attack on a Birthday Problem. *Mathematics Magazine*, 49(3):130–131, 1976. doi: 10.2307/2690189.

[7] L. Guo and I. Matta. The war between mice and elephants. In *ICNP*, pages 180–188. IEEE Computer Society, 2001.

[8] W. Knight and D. Bloom. A Birthday Problem. *The American Mathematical Monthly*, 80(10):1141–1142, 1973. doi: 10.2307/2318556.

[9] S. Lagershausen. *Performing analysis of closed queueing networks*. Springer, 2013.

[10] M. A. Marsan, M. Garetto, P. Giaccone, E. Leonardi, E. Schiattarella, and A. Tarello. Using partial differential equations to model tcp mice and elephants in large ip networks. *IEEE/ACM Trans. Netw.*, 13(6):1289–1301, 2005. doi: 10.1109/TNET.2005.860102.

[11] F. H. Mathis. A Generalized Birthday Problem. *SIAM Review*, 33(2):265–270, 1991. doi: 10.1137/1033051.

[12] E. H. Mckinney. Generalized Birthday Problem. *The American Mathematical Monthly*, 73(4):385–387, 1966. doi: 10.2307/2315408.

[13] J. I. Naus. An Extension of the Birthday Problem. *The American Statistician*, 22(1):27–29, 1968. doi: 10.2307/2681879.

[14] K. Trivedi and R. Wagner. A Decision Model for Closed Queuing Networks. *IEEE Transactions on Software Engineering*, 5(4):328–332, 1979. doi: 10.1109/TSE.1979.234199.