

Beehaviour

‘A STUDY ON FORAGING STRATEGIES USED BY BEES’



Thesis submitted in partial fulfilment
of the requirements for the degree of
Master of Science of Operations Research
at the Faculty of Humanities and Sciences
of Maastricht University

Xavier Gubbels

August 2008

Supervisors:
dr. Frank Thuijsman
dr. ir. Jeroen Kuipers

Summary

Foraging strategies are widely studied and are of great help in finding solutions for all kinds of problems. These strategies are interpreted by for instance biologists, economists and mathematicians to explain certain behaviour. Why do animals act in a certain manner? What can explain their optimal foraging behaviour? How can these strategies be used to solve the traffic congestion problem?

Two strategies based on foraging behaviour are ε -Sampling Strategy and Failures strategy. These strategies are based on the foraging behaviour of bumblebees. Bees tend to do the right thing in an environment of many bees, but seem to do the wrong thing when they are alone. The right thing is adopting the Ideal Free Distribution and the wrong is stick to the Matching Law. Both of the strategies explain that in a multi-bee environment, they will distribute themselves over the flowers (nectar sources) according to the Ideal Free Distribution. But in a single-bee situation a bee will match, by its number of visits, the nectar supply of the available flowers (nectar sources), according to the Matching Law.

Since they are so simple, can these basic strategies be extended or merged into new strategies with possible better performance and results? For this question some extensions to the basic strategies and a new strategy were developed and these extensions and strategies were simulated. Furthermore, in the paper introducing the basic ε -Sampling strategy, this strategy was only explained by situations with 2 different sources / types of flowers. When raising this number of sources, will the same results be found and what has to be adjusted to the basic strategy to perform these situations?

In the case of the ε -Sampling strategy, the extensions include a dynamic ε and the addition of more types of flowers. For the Failures strategy an extension is proposed, that relates the number of failures to a time window. Additionally a new strategy, named ε -Failures strategy, which combines the key elements of the basic strategies is introduced.

After defining the extensions and new strategies, they are tested by simulations and the results are compared to the basic strategies. It can be concluded that all of the extensions and also the new combined strategy won't perform as good as the basic strategies and although some results may show that they perform as well, this is due to the fact that in these cases the effect of the extensions can be neglected and thus the extended strategies will perform as if they are the basic strategies.

It can be said that the strength of these basic strategies is just their robustness and simplicity, by disturbing these basic principles the results and performance will drop.

Table of Contents

| | |
|--|-----------|
| SUMMARY | 2 |
| TABLE OF CONTENTS | 3 |
| PREFACE..... | 4 |
| 1. PRELIMINARIES | 5 |
| INTRODUCTION | 5 |
| MOTIVATION..... | 5 |
| BACKGROUND | 6 |
| GOAL OF THE THESIS..... | 7 |
| OUTLINE OF THE THESIS | 7 |
| 2. STRATEGIES | 8 |
| INTRODUCTION | 8 |
| MOTIVATION..... | 8 |
| ϵ -SAMPLING STRATEGY | 9 |
| <i>Basics</i> | 9 |
| <i>From 2 to n colours</i> | 12 |
| <i>Dynamic ϵ-Sampling Strategy</i> | 17 |
| FAILURES STRATEGY | 19 |
| <i>Basics</i> | 19 |
| <i>Failures Time Window Strategy</i> | 21 |
| COMBINATION OF ϵ -SAMPLING STRATEGY AND FAILURES STRATEGY | 23 |
| <i>Introduction</i> | 23 |
| <i>The ϵ-Failures Strategy</i> | 23 |
| 3. SIMULATION | 24 |
| INTRODUCTION | 24 |
| SIMULATIONS..... | 24 |
| <i>Matching Test for ϵ-Sampling Strategy</i> | 25 |
| <i>Matching Test for Dynamic ϵ-Sampling Strategy</i> | 26 |
| <i>IFD Test for ϵ-Sampling Strategy</i> | 27 |
| <i>IFD Test for Dynamic ϵ-Sampling Strategy</i> | 28 |
| <i>Matching Test for Failures Strategy</i> | 28 |
| <i>Matching Test for Failures Time Windows Strategy</i> | 28 |
| <i>Testing the ϵ-Failures Strategy</i> | 29 |
| 4. RESULTS..... | 30 |
| INTRODUCTION | 30 |
| ϵ -SAMPLING STRATEGY | 30 |
| DYNAMIC ϵ -SAMPLING STRATEGY | 37 |
| FAILURES STRATEGY | 41 |
| ϵ -FAILURES STRATEGY..... | 45 |
| 5. CONCLUSIONS AND RECOMMENDATIONS | 47 |
| INTRODUCTION | 47 |
| CONCLUSIONS | 47 |
| RECOMMENDATIONS FOR FUTURE RESEARCH | 49 |
| 6. REFERENCES | 50 |
| 7. APPENDIX..... | 51 |

Preface

In this thesis I will present you the study entitled 'Behaviour', which I wrote it in the context of the Master's program 'Operations Research' of the study Knowledge Engineering. To finish this study you will be considered to make a Master thesis, besides achieving good results for all the exams and projects. This can be done in several ways, you can do some practical experience in the business sector or do scientific research. I have chosen the latter option, a scientific research. For me the most difficult part of the entire study, because I had to do this on my own. During projects there was always some cooperation with other students and also studying for exams was more than once done in groups. Now I had to do all the work by myself, which has costs a lot of blood, sweat and tears. It also took a (too) long time.

I had never thought that I would talk about flowers and bees, but now I am! Many have scratched their head when I started talking about the subject of this thesis. It is not an everyday topic and to a lot of people the subject doesn't sound scientific or mathematical, but after further explanation about this subject, most people were convinced of the goal that I had in mind, this thesis!

I especially would like to thank Frank Thuijsman. He was always there to help me when I didn't know how to continue. He was able to give me new inspiration and ideas to continue, even though this sometimes lasted longer than expected. I also would like to thank Jeroen Kuipers for his help and guidance, especially on the Matlab simulation. Finally I thank Manon, my parents and everyone else that I do not mention by name for all their support, patience and confidence over the past period. Without these people, I was not able to complete this thesis.

Maastricht, 26th August 2008

Xavier Gubbels

1. Preliminaries

Introduction

This chapter is intended to initiate the thesis written in the context of the graduation of the master program in Operations Research at the University of Maastricht. The research concerns the development of 'new' foraging strategies, which are based on the observed foraging behaviour of bumblebees. First, a study should be done to the background of the problem and existing foraging strategies used by bees, which the 'new' strategies are based on. In order to test the strategies and to view and analyze the results, a program is developed that simulates a bee-flower environment. From the results, obtained by the program and analysis on the basis of other existing strategies, conclusions and recommendations for further research will follow.

Motivation

Foraging theory studies the foraging behaviour of animals in response to the environment in which the animal lives. It considers the foraging behaviour of animals in reference to the payoff that an animal obtains from different foraging options. Foraging theory predicts that the foraging options that deliver the highest payoff should be favored by foraging animals, because it will have the highest fitness payoff (Stephens, D.W. & J.R. Krebs, 1986). One animal (or insect in this case) whose foraging behaviour has been studied is the bee. There are many articles published on the way that bees search for food in order to comply in their life requirements (e.g. Heinrich, 1976; Cartar & Dill, 1990). These ways can be characterized by strategies, and most of them are different in intent. We can distinguish between strategies that are deterministic or stochastic, discrete or continuous in nature, whether or not use fill-rate or fill-time, and for example look at the behaviour of bees at individual level (energy intake and consumption for its own life) or the effect within a group.

For all these strategies to be categorized and compared, it is important to have a (simulation) program which can be used to experiment. On the basis of the performances of different strategies, they can be compared to one another. These benefits are to be classified and categorized. It may take into account individual or group behaviour, maximizing nectar intake or minimization of flight time or energy.

The behaviour of bees can also be projected on, for example, the behaviour of people. Strategies used by bees, are also used by people in daily living in a certain way. Where do we obtain our groceries, which car we do have buy, on which party is to be voted? These choices are made by means of assumptions, expectations, feeling and more. To be able to understand these choices, models can be made of the reality. These models give (in a limited way) a view of reality, by the way of restricting possibilities and the surroundings. This simplification of reality can improve the ability to examine behaviour and making it easier to take conclusions.

Two strategies which have been examined and described already in literature, are ' ϵ -Sampling strategy' and 'Failures strategy'. These strategies will be further explained in the next chapter. These strategies assume behaviour of bees, but as said, can also be applied in our daily lives, for example in choosing a restaurant. The variables get another meaning, but the outcome is the same; an optimization of the reward given at a certain choice. The reward can be indicated in the case of the restaurant choice with the appreciation for price/quality proportion.

Some other problem: When commuting to work, the quickest route is pursued. When people have to travel from A to B and the roads are crowded, then they tend to distribute themselves over all possible tracks in such a way (without communicating to each other directly), that traffic will be equally distributed over all possible routes from A to B. As a result the average travel time of all commuters from A to B is minimized.

The subject of this thesis is the possible improvement of 2 existing strategies, 'ε-Sampling' and 'Failures'. The strategies are adapted and combined with each other to develop new strategies, with possibly better results, where better should be interpreted as more realistic. Results concern amongst other things, how the bees distribute themselves over flower types (Ideal Free Distribution), how rapidly this distribution is reached and what this means for the behaviour of individual bees.

Background

Bees were already studied by former philosophers such as Aristoteles and Virgil, where the emphasis lay especially on their history and product. A lot of others have studied bees, in all kinds of manners, by the centuries, however most of the knowledge has been gathered since the sixteenth century.

Even Albert Einstein is supposed to have said: "If the bee disappeared off the surface of the globe then man would only have four years of life left. No more bees, no more pollination, no more plants, no more animals, no more man."

Scientists from the twentieth century have discovered how the sense organs of bees work, how bees communicate, the effect of the poison which is released when a bees stings and they have determined and analyzed the behaviour of bees by means of more and more sophisticated measuring systems. We now know for example that bees see several colours and flickering patterns (which they observe when they fly on short distance above flowers) and that they use certain strategies to optimize their nectar consumption.

Meanwhile bees are subject to a lot of research areas from several fields of biology, such as evolutionary biology, to (among other things in this case) mathematics. In mathematics, research concerning bees is done often in association with other research areas, for example biologists. Many models, concerning the choices which bees make during foraging, have been made, to get more insight on how and why bees forage and approach flowers in a certain manner (Kadmon, R, et al, 1991).

More recently it has been shown, that bees develop a long-term expectation of the nectar income and by means of the size of a bee, we can make an estimation of the size of the foraging area is (Greenleaf et al, 2007).

An Ideal Free Distribution (IFD) is an ecological term, which describes the manner which animals distribute themselves over different areas with food. The theory explains that the individual animals will distribute themselves proportionally to the available quantities of resources in each area. For example, if area A contains twice as much food as area B, then there will be twice as many animals in area A than in area B. The IFD theory predicts the partitioning of animals which forage in certain areas. This term and theory have been introduced by Fretwell and Lucas in 1970 (Fretwell & Lucas, 1970).

Ecologists frequently use this theory to test for violations. If a population of animals isn't distributed according to the ideal free distribution, it is interesting to find out why.

The Matching Law was first formulated by R.J. Herrnstein in 1961 (Herrnstein, 1961) following an experiment with pigeons. Pigeons had to choose between two buttons, each of which would lead to varying rates of food reward. It was observed that pigeons would choose the button with the higher food reward more often than the button with the lower payoff. Remarkable was, they did so at a rate that was similar to the rate of reward. Stated simply, the Matching Law suggests that an animal's response rate to a scenario will be proportionate to the amount or duration of positive reinforcement delivered.

Goal of the thesis

The aim of this thesis is a research on foraging strategies used by bees. This means examining and analyzing existing strategies and developing new strategies, which are based on the observed foraging behaviour of bumblebees. Furthermore the implementation of a program/tool, by which the strategies can be compared on their performances. The main goal is to get an answer to the problem statement and research questions.

Problem statement:

- Do the results from the basic paper hold for situations with more than 2 types of flowers and if the basic strategies are extended to meet more complex situations, how will this perform compared to the basic strategies?

This leads to the following research questions:

- Will the results in the basic paper also hold in more complex situations, i.e. more than 2 types of flowers?
- In what way have the basic strategies be adjusted or extended to meet the more complex conditions?
- How do the extended, new strategies perform in comparison to the basic strategies?

Outline of the thesis

The outline of the remaining part of this thesis is as follows: In chapter 2, several existing strategies are discussed and some new strategies will be introduced. Then in chapter 3, the existing and new strategies will be simulated and compared to each other. Chapter 4 describes the results, which are obtained during the research. Finally in chapter 5, the conclusions will be described and recommendations for further research are given.

2. Strategies

Introduction

In this chapter the strategies examined are explained. To start it is motivated why these particular strategies are chosen. Next these basic strategies are discussed to give a good explanation of these basic strategies, ε -Sampling Strategy and Failures Strategy. Hereafter the extension of the number of colours is discussed, since the results of the basic strategies only take 2 different colours into account. At the end of the chapter we introduce some extensions to this basic strategies (dynamic ε -Sampling Strategy and Failures Windows Strategy) and formulate a new strategy, ε -Failures Strategy, which uses the basic principles of the ε -Sampling Strategy and Failures Strategy.

Motivation

The basis for my research is an article (Thuijsman et al, 1995), which attempts to explain observed behaviour of animals (bumblebees, in this particular research). These bees do the wrong thing when they are alone, namely stick to the Matching Law. But in an environment of many foragers they act in the right way, that is adopting the Ideal Free Distribution.

The strategies have to meet the bees' distinctive characteristics to give a good representation of the reality. Bumblebees do not exchange foraging information and they try to maximize its average nectar intake per time unit. Nectar will be considered as the only reward for the bee in the model. A given flower type has a typical colour and gives a certain reward. In the basic paper (Thuijsman et al, 1995) only 2 types of flowers are used.

Furthermore, the strategies are very simple in design and therefore easy to understand. They are described by finite automata (Ben-Porath & Beleg, 1987; Kalai, 1990), by which the bees only respond to their own payoffs and remember only the payoffs of the last few visits. The decision rules, used to decide whether to stay in or to leave a patch, take only the last one or two flowers into account. This is in line with the small brains and bounded memory of the bees.

In the article, 2 simple foraging strategies are introduced; (i) the ε -Sampling Strategy and (ii) the Failures Strategy.

To meet the observed behaviour, both strategies have the characteristics that (i) in a single-bee setting, a bee will match it's number of visits to the nectar supply from the available sources, according to the Matching Law (Hernstein, 1961) and (ii) in a multi-bee community, the bees will distribute themselves over the available nectar sources according to the Ideal Free Distribution (Fretwell & Lucas, 1970).

ϵ -Sampling Strategy

Basics

In common words; a bee visits flowers of a certain (initial) colour repeatedly, but sometimes the bee samples a flower of a different colour by mistake. If the reward at the accidentally visited flower is higher, then the bee will stay at this new type of flower. If not, it will return immediately to the patch with the original colour.

More formal:

Definition ϵ -Sampling strategy

Let $a, \epsilon \in (0,1)$, let $a(t) \in \{Y,B\}$ represent the action selected and let $r(t) \in \mathbb{R}$ be the payoff at time $t \in \{1,2,3,\dots\}$.

Define $cl(1)=0$ and $cl(t+1)=acl(t)+(1-a)r(t)$ for $t \geq 1$.

Then $cl(t)$ is called the critical level at time t . Let Y_ϵ denote the mixed action: choose Y with probability $1-\epsilon$ and B otherwise and let B_ϵ be defined similarly. The ϵ -sampling strategy is defined by playing:

at $t=1$ use $Y_{0.5}$,

at $t=2$ use $a(1)_\epsilon$,

at $t > 2$ use $a(t-1)_\epsilon$ in case $a(t-1) \neq a(t-2)$ and $r(t-1) \geq cl(t-1)$, use $a(t-2)$ otherwise.

In this definition a and ϵ are the individual bee's factors.

In the multi-bee situation, we need some assumptions for the population to stabilize in the Ideal Free Distribution, namely:

(i) The total quantity of nectar at Y is equally distributed over the visiting bees at Y at each stage; that is, if at some stage $t \in \mathbb{N}$ there are n_Y bees at Y , then we assume that at this stage each of these bees is receiving $r(t) = y/n_Y$ units of nectar. Likewise for B .

(ii) There is no accumulation of nectar at a patch. All nectar is taken by the bees at each stage and if at some stage there are no bees at a patch then at the next stage the total quantity is still the same.

(iii) The bees' sampling factors ϵ are sufficiently close to 0 to have a negligible probability of two or more bees moving at the same time. Thus the distribution of bees over flower types changes by one bee moving either from Y to B or from B to Y . (Here 'moving from Y to B ' means: previously the bee was at Y , now it has gone to B for a sample and since the payoff received at B is larger than the critical level it decides to stay at B .)

(iv) We assume that, when going out sampling, the bee has been in Y sufficiently long to have its critical level close to y/n_Y .

The population will stabilize in the Ideal Free Distribution under the above assumptions. Firstly, due to its ε , each bee will sample the other colour infinitely often. Hence, if the process stabilizes in some distribution of bees over Y and B, then it must be such that no single bee strictly improve its payoff by moving to the other colour. Secondly, whenever a bee moves from Y to B we must have that $y/n_Y < b/(n_B+1)$, where n_Y and n_B are the numbers of bees before the move at Y and B respectively. If a potential function (Monderer & Shapley, 1988) is defined as

$$P(n_Y, n_B) = y \sum_{m=1}^{n_Y} 1/m + b \sum_{m=1}^{n_B} 1/m$$

where $\sum_{m=1}^0 1/m$ is understood to be equal to 0. With each bee movement the potential strictly increases and since there are only finitely many distributions of the n bees and the bees will keep moving as long as possible, the potential function will eventually reach its maximum. This leads to the situation (at this maximum) that $y/n_Y \geq b/(n_B+1)$ and $b/n_B \geq y/(n_Y+1)$, hence $y/n_Y \approx b/n_B$. Furthermore, if $y/n_Y = b/n_B$, then $n_Y/n_B = y/b$ and thus the bees will be distributed over the colours according to the Ideal Free Distribution.

In a single-bee situation we use an artificial type of flowers, namely Bernoulli flowers Y (yellow) and B (blue). The yellow flowers give a reward of 1 unit of nectar with probability p and 0 units otherwise. For the blue flower we have probability q for a full flower. In a natural situation, this can be described as a constant fraction p of the patch of yellow flowers is full (and fraction $(1-p)$ is empty) and a constant fraction q of the blue flowers is full (and fraction $(1-q)$ is empty).

The strategy can be described by the following Markov chain.

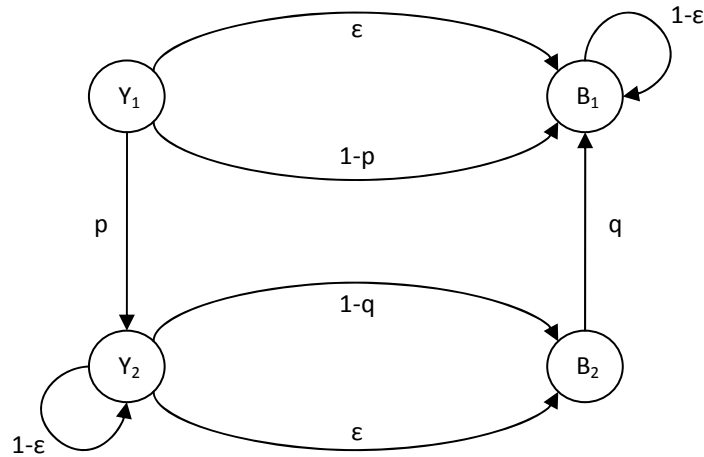


Figure 1: Markov chain for the ε -Sampling strategy

In this situation, we have 2 types of flowers (Y and B). Each time a bee visits a flower of type Y, the bee will get 1 unit of nectar with probability p and 0 units otherwise. This is the same for colour B, except in this case the bee will get a reward of 1 unit with probability q .

Y_1 and B_1 are the sample stages (the bee gets there with probability ε) and Y_2 and B_2 are the stages where a bee has chosen to stay in that patch (if the bee finds a full flower with probability p or q respectively).

As seen in the multi-bee situation, ε is the bee's sampling factor. Because of the rewards, the bee's critical level will always be between 0 and 1, and a bee will only stay in a sampled patch if it gets a full flower.

This Markov chain corresponds with the following transition matrix T given by:

$$T = \begin{array}{c} \\ \\ \\ \\ \end{array} \begin{array}{c} \\ \\ \\ \\ \end{array} \begin{array}{cccc} Y_1 & Y_2 & B_1 & B_2 \\ \left[\begin{array}{cccc} \mathbf{0} & \mathbf{p} & \mathbf{0} & \mathbf{1-p} \\ \mathbf{0} & \mathbf{1-\varepsilon} & \mathbf{\varepsilon} & \mathbf{0} \\ \mathbf{0} & \mathbf{1-q} & \mathbf{0} & \mathbf{q} \\ \mathbf{\varepsilon} & \mathbf{0} & \mathbf{0} & \mathbf{1-\varepsilon} \end{array} \right] \end{array}$$

The stationary distribution $\pi = (Y_1, Y_2, B_1, B_2)$ of T gives us the frequency of visits at Y ($Y_1 + Y_2$) and B ($B_1 + B_2$). The vector π is nonnegative, adding its components gives 1 and $\pi T = \pi$.

Computing this vector shows that the number of visits of colour Y is $p + q\varepsilon$ and the number of visits of flowers of colour B is $q + p\varepsilon$. Hence $n_Y : n_B \equiv p + q\varepsilon : q + p\varepsilon \approx p : q$.

This leads to the theorem that: *If ε is small, then a bee applying the sampling strategy in a single-bee experiment will exhibit matching the payoff probabilities by the frequency of its visits (the Matching Law).*

There has to be emphasized that the results presented in the paper do not really depend on ε . The only obligation which must be obeyed is that ε has to be close to 0, in order to have the bee staying long enough in a patch to get a good estimation of the patch's payoff.

After examining this strategy some questions can be asked. The paper only focuses on 2 different colours, it is said that the results will hold in situations with more than 2 different colours. This is something we want to examine and analyze, for both the single-bee and multi-bee setting.

In the paper it is also said that ε can be payoff dependent, thus getting larger if the bee gets low payoffs and will decrease if the bee receives high payoffs. But what if this dynamic ε becomes too large and close to 1? It is also stated that ε has to be sufficiently small for the bee to stay in a patch long enough and get a good estimate of the payoff in the patch. In the case of an ε close to 1, the bee has a great probability of leaving the patch and thus no opportunity to get an estimation of the patch's payoff. Another question is how ε has to be defined if it becomes dynamic.

These questions will be discussed in the following sections, which will be about the 'extensions' of the ε -Sampling strategy.

From 2 to n colours

The experiments in the paper only took 2 different colours into account. But will the results also stand for instances of the strategies with more than 2 different colours?

Probably they will. But when there are more than 2 colours, the model has to be extended with some rules. For instance, when a bee samples, it has to sample a different colour. For the choice of the sampling colour we assume that:

- A bee chooses a different colour at random.

It is possible that a bee doesn't know all the flowers and colours in the foraging area. Following to this, if a bee doesn't know all the flowers and colours in a patch, it surely doesn't know the highest estimated payoff of those flowers and colours and the last 2 assumption don't hold. Therefore a colour is chosen at random.

Furthermore, after (or more exact during) the sampling, what is the bee's action when it receives a payoff with a value below the critical level (or in the case of Bernoulli flowers gets a empty flower)? For this matter we made the following assumption:

- A bee will return to a flower of its previous visited colour if the bee doesn't get a positive payoff at the sampled flower (and colour).

In a multi-bee situation, the population reaches the Ideal free Distribution if

$$q_1/n_1 \approx q_2/n_2 \approx \dots \approx q_i/n_i$$

where $q_i > 0$ for all i

Where q_i is the total amount of nectar at patch i , n_i is the number of bees at colour i and i is the number of colours. With the help of the simulations we can computed these ratios and see if the population is distributed according to the Ideal Free Distribution.

For the case of a single-bee situation and $n > 2$, the Markov chain and transition matrix T have to be extended. For the Markov chain we introduce a new system of states and variables. The states and transitions can be characterized by

- p_n is the probability, for flowers with colour n , giving a reward of 1 unit of nectar and thus for the bee obtaining a success.
- c_{nm} , corresponds to a state of the bee, where the bee is sampling colour m while coming from colour n .
- ε , the sampling factor, with this probability the bee will sample another colour.

For an organized view the states have been grouped by colour, all states with c_{11} to c_{1m} have to be grouped, such as c_{22} to c_{2m} and c_{n1} to c_{nm} have to be grouped. With numbers, a state c_{14} corresponds to a bee which is currently visiting a flower of colour 1, but is sampling a flower of colour 4.

As an example a Markov chain for 4 colours is presented:

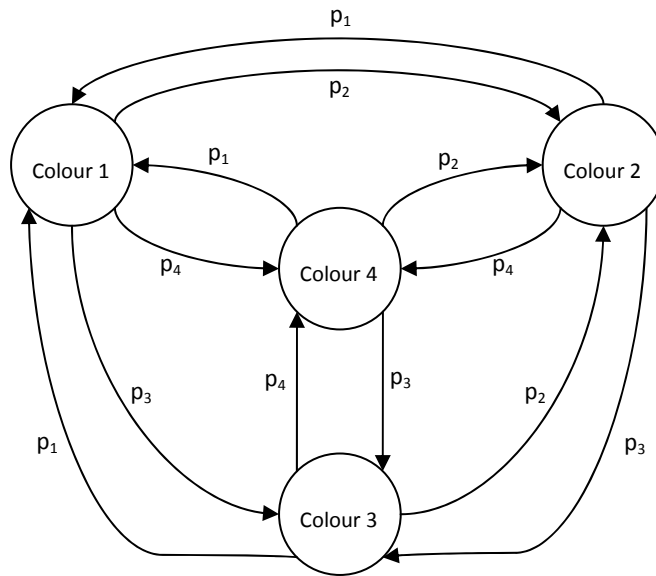


Figure 2: The Markov chain for 4 colours

A colour node in the Markov chain above can be subdivided by the inner colour states:

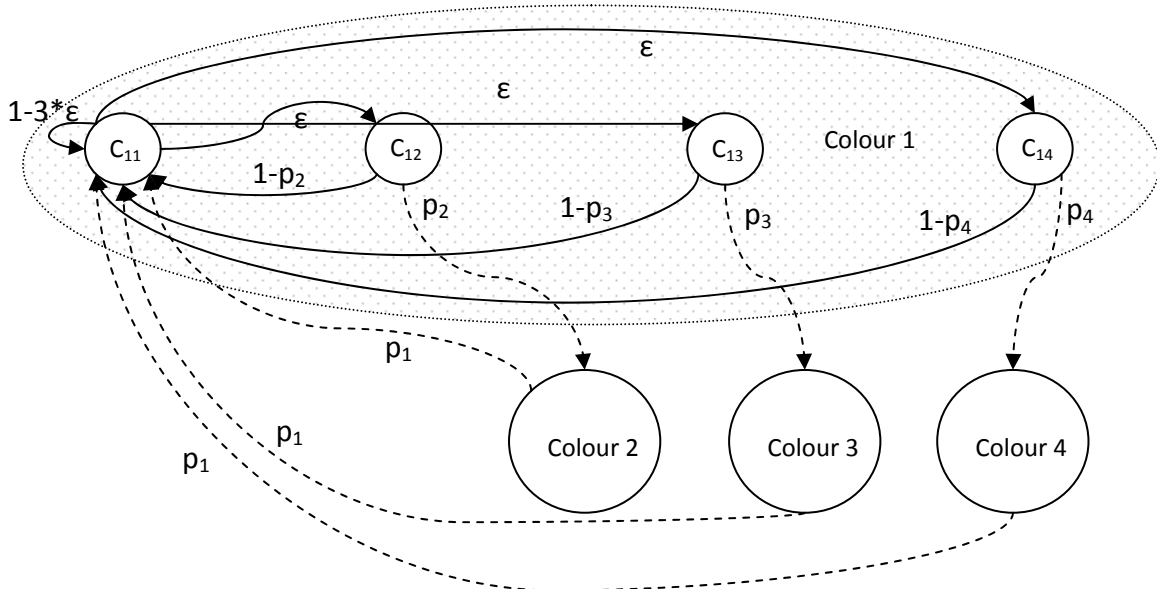


Figure 3: Inner colour states of the Markov chain

For the example the 'inner-colour' chain of colour 1 is chosen, the other colours can be represented similar.

The corresponding transition matrix T_n is given by:

$$T_n = \begin{bmatrix} A_{ii} & \cdots & A_{ij} \\ \vdots & \ddots & \vdots \\ A_{ji} & \cdots & A_{jj} \end{bmatrix}, \text{ for } i, j \in [1, n]$$

The elements of matrix T_n (each have n rows and n columns) have to be defined as:

$$A_{ij} = \begin{matrix} & \text{column } j \\ \text{row } j & \begin{bmatrix} 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & p_{jj} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \end{matrix}, \text{ if } i \neq j$$

$$A_{ij} = \begin{matrix} & \text{column } i \\ \text{row } i & \begin{bmatrix} 0 & \cdots & 0 & 1 - p_1 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 - p_{i-1} & 0 & \cdots & 0 \\ \varepsilon & \cdots & \varepsilon & 1 - (n-1) * \varepsilon & \varepsilon & \cdots & \varepsilon \\ 0 & \cdots & 0 & 1 - p_{i+1} & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 - p_n & 0 & \cdots & 0 \end{bmatrix} \end{matrix}, \text{ if } i = j$$

For a situation with 3 types of flowers, the following Transition matrix T_3 is defined:

$$T_3 = \begin{bmatrix} 1 - (n-1) * \varepsilon & \varepsilon & \varepsilon & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 - p_2 & 0 & 0 & 0 & p_2 & 0 & 0 & 0 & 0 \\ 1 - p_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & p_3 \\ p_1 & 0 & 0 & 0 & 1 - p_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \varepsilon & 1 - (n-1) * \varepsilon & \varepsilon & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 - p_3 & 0 & 0 & 0 & p_3 \\ p_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 - p_1 \\ 0 & 0 & 0 & 0 & p_2 & 0 & 0 & 0 & 1 - p_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & \varepsilon & \varepsilon & 1 - (n-1) * \varepsilon \end{bmatrix}$$

By finding the stationary distribution $\pi = (c_{11}, c_{12}, \dots, c_{ij}, c_{21}, c_{22}, \dots, c_{2j}, \dots, c_{i1}, c_{i2}, \dots, c_{ij})$ we can compute the frequency of visits of each colour. For colour 1 the frequency is $c_{11} + c_{12} + \dots + c_{1j}$ for colour 2 is $c_{21} + c_{22} + \dots + c_{2j}$ and so one.

Thus for the situation where $n=3$ (3 colours), $p_1=0.2$, $p_2=0.3$, $p_3=0.5$ and $\varepsilon = 0.05$, we'll get the following Markov chain and Transition matrix:

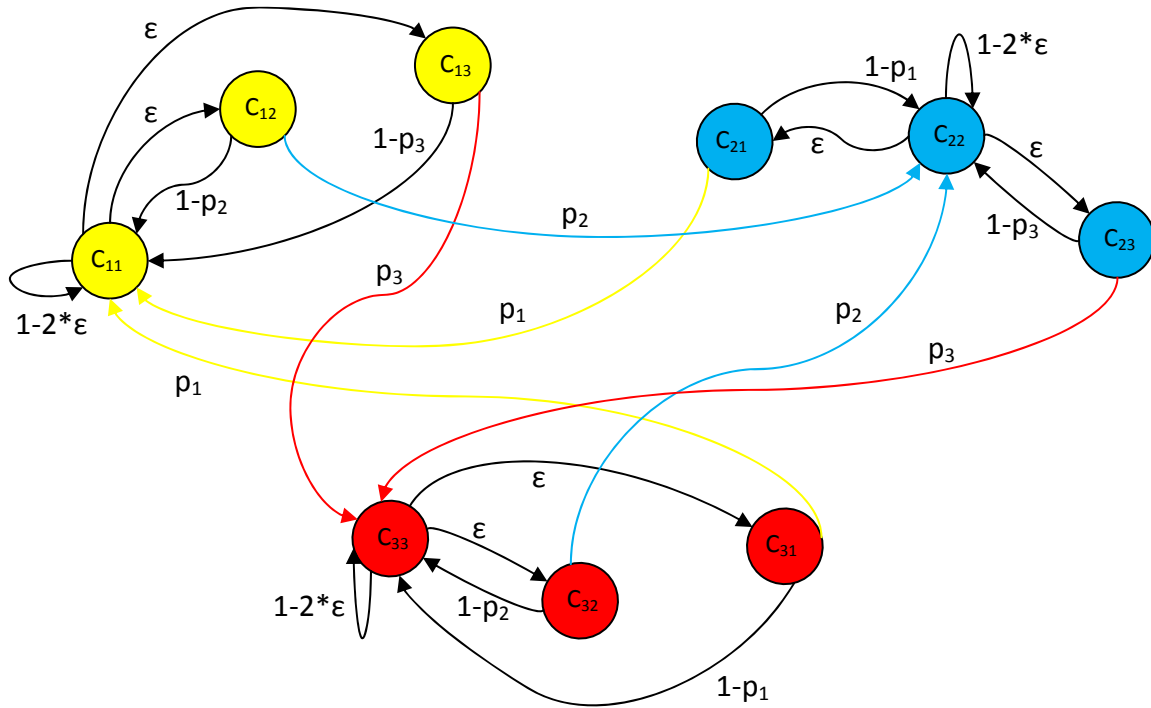


Figure 4: Markov chain for $n=3$

Replacing the variables with the corresponding probabilities gives:

$$T = \begin{bmatrix} 0.9 & 0.05 & 0.05 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.7 & 0 & 0 & 0 & 0.3 & 0 & 0 & 0 & 0 \\ 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.6 \\ 0.2 & 0 & 0 & 0 & 0.8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.05 & 0.9 & 0.05 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.4 & 0 & 0 & 0 & 0.6 \\ 0.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.8 \\ 0 & 0 & 0 & 0 & 0.3 & 0 & 0 & 0 & 0.7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.05 & 0.05 & 0.9 \end{bmatrix}$$

For computing the frequencies of visits to the different colours, we have to find stationary distribution $\pi = (\pi_{11}, \pi_{12}, \pi_{13}, \pi_{21}, \pi_{22}, \pi_{23}, \pi_{31}, \pi_{32}, \pi_{33})$ of T . This can be done by solving $\pi T = \pi$ and $\sum_{i=1}^9 \pi_i = 1$. We get $\pi = (0.1653, 0.0083, 0.0083, 0.0124, 0.2479, 0.0124, 0.0248, 0.0248, 0.4959)$.

Hence the frequency of visits per colour is;

- Colour 1 = $0.1653 + 0.0083 + 0.0083 = 0.1819$
- Colour 2 = $0.0124 + 0.2479 + 0.0124 = 0.2727$
- Colour 3 = $0.0248 + 0.0248 + 0.4959 = 0.5455$

The frequency may not be matching the probabilities of getting a positive reward (and even is 1.0001 due to rounding error), but this occurs since $p_1+p_2+p_3 = 0.2 + 0.3 + 0.6 = 1.1$. In this case we have to take a look at the normalized probabilities, so we get a clear view of the allocation of the frequency of the visits per colour.

Since $p_1+p_2+p_3 = 1.1$, the probabilities have to be divided by 1.1 to get the normalized probabilities. For the colours, the normalized probabilities for visiting a flower with reward 1 are:

- Colour 1 = $p_1 / (p_1 + p_2 + p_3) = 0.2 / 1.1 = 0.1818$
- Colour 2 = $p_2 / (p_1 + p_2 + p_3) = 0.3 / 1.1 = 0.2727$
- Colour 3 = $p_3 / (p_1 + p_2 + p_3) = 0.6 / 1.1 = 0.5455$

These values match with the computed values for the stationary distribution of T .

With the help of the simulations we can compute and analyze these rates for all of the desired situations. These will be discussed in de next chapters, where we start with how the simulations have been done and after that we discuss the results.

Dynamic ε -Sampling Strategy

The ε -Sampling Strategy is based on the fact that a bee accidentally visits the ‘wrong’ patch with flowers of a different colour. This mistake, or sample, is indicated by the probability ε , with $\varepsilon \in (0,1)$. This probability does not differ during the foraging of the bee and is considered to be ‘close to 0’ (it’s a mistake and happens very rare).

But what if the frequency of the sampling depends on the current payoff? We want to see what happens, when the bee will sample more often when the rewards at the current patch are below its critical level. On the other hand, the bee will feel less need to explore (sample) the patches with flowers of other colours when the bee only gets payoffs above its critical level.

The model has to be extended with a function, which describes this dynamic ε . We chose to let the dynamic ε function to be quadratic. It could also be linear, but the changes of epsilon would not be noticeable when ε is near 0 and thus would not have enough impact on the behaviour.

The dynamic ε function is defined as

$$\begin{aligned}\varepsilon(0) &= \varepsilon_0, \varepsilon_0 \in (0,1) \\ \varepsilon(1) &= \varepsilon(0) \\ \varepsilon(t+1) &= \varepsilon(0)^{1/(k(t)+1)}, \text{ for } t > 1\end{aligned}$$

For this function a variable k is introduced. This variable is a perspective of the number of failures and successes. This number can be defined in 2 ways, (i) from the beginning of the simulation and (ii) from the last success. The initial value of k is 0.

The first manner can be defined as just counting the difference between total number of failures and total number of successes witnessed until then. The definition of the ε -Sampling Strategy has to be extended with the following lines to satisfy this condition.

Let $k(t), s(t), f(t) \in \mathbb{N}$, $s(t)$ is the total number of successes and let $f(t)$ be the number of failures. Furthermore $r(t)$ is the current reward and $cl(t)$ is the current critical level.

Define $k(0)=0, s(0)=0$ and $f(0)=0$,

$$\begin{aligned}\text{if } r(t) \geq cl(t), s(t) &= s(t-1) + 1 \text{ and } f(t) = f(t-1), \\ \text{otherwise } s(t) &= s(t-1) \text{ and } f(t) = f(t-1) + 1 \\ \text{then } k(t) &= \max\{f(t) - s(t), 0\}\end{aligned}$$

As can be stated from the above, this function is colour independent. We assume that a bee’s memory does not keep track of the ratio in successes and failures per colour. A bee only remember the ratio of successes and failures of the current visiting colour. When a bee moves to another colour, the memory will be cleared and both numbers of successes and failures are set to 0.

The second method is the most straightforward of both. The bee will start counting after attaining an empty flower, thus failure. With every consecutive failure, k will be added with 1. Whenever the bee visits a full flower, a success, the value of k will be reset to 0. By this means, the definition of the ε -*Sampling Strategy* has to be extended with the following.

Let $k(t) \in \mathbb{N}$ and define $k(0)=0$, $r(t)$ is the current reward and $cl(t)$ is the current critical level, then

$$\text{if } r(t) \geq cl(t), k(t) = 0, \text{ otherwise } k(t) = k(t - 1) + 1$$

Notice that, when in both methods artificial Bernoulli flowers are used, $cl(t)$ doesn't have to be used. Because $r(t) \in \{0,1\}$, we can speak of noticing a failure ($r(t)=0$) or a success ($r(t)=1$), and since $cl(t) \in (0,1)$, a success has always $r(t) > cl(t)$ and for a failures the reverse $r(t) < cl(t)$.

Thus the latter definition extension can be rewritten as:

$$\text{if } r(t) = 1, k(t) = 0, \text{ otherwise } k(t) = k(t - 1) + 1$$

Another way of taking failures into account when deciding to stay or leave a patch is considered, but this fits more in an adaptation or extension of the Failures strategy and will be discussed in the next section. This method puts focus on a certain window of the last visits and the number of failures in that window. For instance, raise ε if 3 of the last 5 visited flowers are empty. This method could also be colour dependent, a bee could leave colour Y after 2 (consecutive) failures and leave colour B after 3 (consecutive) failures.

Both methods will be simulated and analyzed after the simulations and discussed in the following chapters.

Failures Strategy

Basics

In common words; A bee will move to another foraging patch after visiting a certain number of consecutive empty flowers, empty should be interpreted as ‘below critical level’.

In a formal way;

Definition Failures strategy

Let y, b be positive integers. The finite automaton $A(y,b)$ is given by:

- (i) Leave Y after y consecutive failures and move to B ,
- (ii) Leave B after b consecutive failures and move to Y .

In this experiment, there are two artificial flowers Y and B , that give 1 unit of nectar with probability p and q respectively (and 0 units otherwise). The event of receiving 1 unit shall be called a success, and receiving 0 is a failure.

As an example, the automaton $A(3,2)$ and the corresponding Markov chain can be displayed as seen in figure 5.

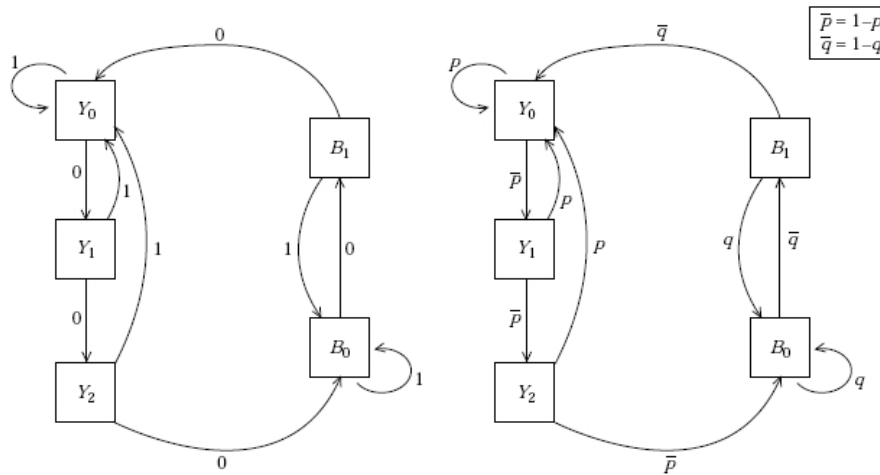


Figure 5: Automaton $A(3,2)$ and corresponding Markov chain (right)

Y_k indicates the state of being at Y , while the last k visits were failures (the B_k states should be interpreted similarly). This automaton leads to a transition matrix T , which is given by:

$$T = \begin{matrix} & \begin{matrix} Y_0 & Y_1 & Y_2 & B_0 & B_1 \end{matrix} \\ \begin{matrix} Y_0 \\ Y_1 \\ Y_2 \\ B_0 \\ B_1 \end{matrix} & \begin{pmatrix} p & \bar{p} & 0 & 0 & 0 \\ p & 0 & \bar{p} & 0 & 0 \\ p & 0 & 0 & \bar{p} & 0 \\ 0 & 0 & 0 & q & \bar{q} \\ \bar{q} & 0 & 0 & q & 0 \end{pmatrix} \end{matrix}$$

The stationary distribution $\pi = (y_0, y_1, y_2, b_0, b_1)$ of T gives us the frequency of visits at Y ($y_0 + y_1 + y_2$) and B ($b_0 + b_1$). The vector π is nonnegative, adding its components gives 1 and $\pi T = \pi$.

Using $A(3,2)$ the bee would confirm the Matching Law if and only if

$$\begin{cases} y_0 + y_1 + y_2 = p/(p+q) \\ b_0 + b_1 = q/(p+q) \end{cases}$$

For generality we use $A(y,b)$ instead of $A(3,2)$, one can show the following theorem (Thuijsman et al, 1995).

Theorem: *The finite automaton $A(y,b)$ matches the payoff probabilities p and q respectively if and only if*

$$\frac{p^2 p^{-y}}{1 - p^{-y}} = \frac{q^2 q^{-b}}{1 - q^{-b}}$$

For any p,q -situation of yellow and blue flowers with $p, q \in [\delta, 1 - \delta]$, with $\delta \in (0, 0.5)$ this theorem provides the existence of real number $y, b \in [1, M]$, where $M \in \mathbb{N}$. When we take the bounded recall of the bee into account, we have to notice that with $M=2$, δ cannot be smaller than 0.22, while with $M=3$, δ can be handled as small as 0.18. This leads to the fact that all p, q $[0.22, 0.78]$, matching can be obtained with an automata $A(y,b)$, with $y, b \in [1, 2]$.

In natural situations, it is frequently observed that p and q are small if there are many visits (Schreiber, 1993). In this case ($p \leq q \leq 1-p$ or $q \leq p \leq 1-q$) the bee can use a special automaton $A(x,x)$ (Thuijsman et al, 1995).

If $p \leq q \leq 1-p$ or $q \leq p \leq 1-q$, then there exists $x \geq 1$ such that

$$\frac{p^2 p^{-x}}{1 - p^{-x}} = \frac{q^2 q^{-x}}{1 - q^{-x}}$$

With a population of finitely many bees using $A(x,x)$, the Ideal Free Distribution can be observed. By the strong law of large numbers it can be said that the fraction of number of bees visiting patch Y converges to $p/(p+q)$, and the other part $q/(p+q)$ will be at B for large numbers bees and a large time t .

Consider a population of infinitely many bees, using $A(r,s)$ to forage patches Y and B , with respectively total nectar supply of y and b . Then p and q are determined by the proportions of bees currently present in each of the patches and this population can be distributed according to the Ideal Free Distribution.

Then with certain p and q a static result for the existence of the Ideal Free Distribution can be calculated, but there is no dynamic process for reaching the Ideal Free Distribution. We have to assume an infinite number of bees in order to have p and q not affected by a single bee moving from one state of the automaton to another state. With a finite number of bees these probabilities would depend on the precise number of bees in those states.

Failures Time Window Strategy

In the strategy only consecutive failures are taken into account, but what happens if the failures window is extended? With extending the failures time window, we mean that not only consecutive failures are measured, but the number m of failures in the last visits n is kept track of, where $m \leq n$. By acting this way, a bee will stay longer at a patch, but we expect that at the end the distribution of the population is the same as for the original strategy.

For instance, instead of changing colour after 2 consecutive failures, a bee will move to another colour if 2 of the last 5 visits at the current colour are failures.

We extend the Failures strategy model $A(y,b)$ to $A_n(y,b)$, where n stands for the size of the window of last visited flowers, then this strategy can be defined as:

The finite automaton $A_n(y,b)$ is given by

- (i) Leave Y if y of the last n visits where failures and move to B
- (ii) Leave B if b of the last n visits where failures and move to Y

Let $n, y, b \in \mathbb{N}, n \geq y$ and $n \geq b$

Note that in the case of $A_x(x,x)$ the model can be thought as if it's equal to $A(x,x)$. Because it takes the last x failures of the last x visits into account and this can be seen as x consecutive failures. This will only hold for an automaton $A_x(x,x)$, because with automaton $A_x(x,z)$ where $x > z$, the bee doesn't have to approach x consecutive failures on colour Z to move to colour X .

We consider that there has to be a model $A_n(2,2)$ which gives the same results as $A(2,2)$. By simulating and testing we try to find a suiting n .

Two of these automata in the single-bee situation are displayed in figure 6 and figure 7:

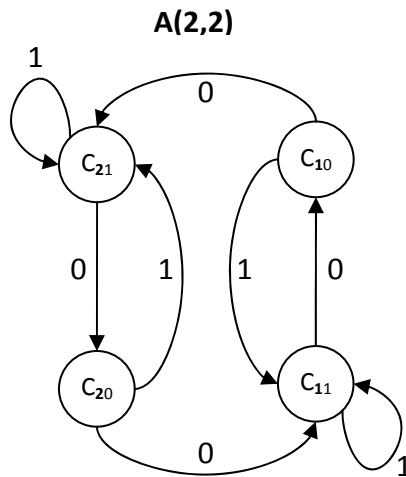


Figure 6: Automaton for A(2,2)

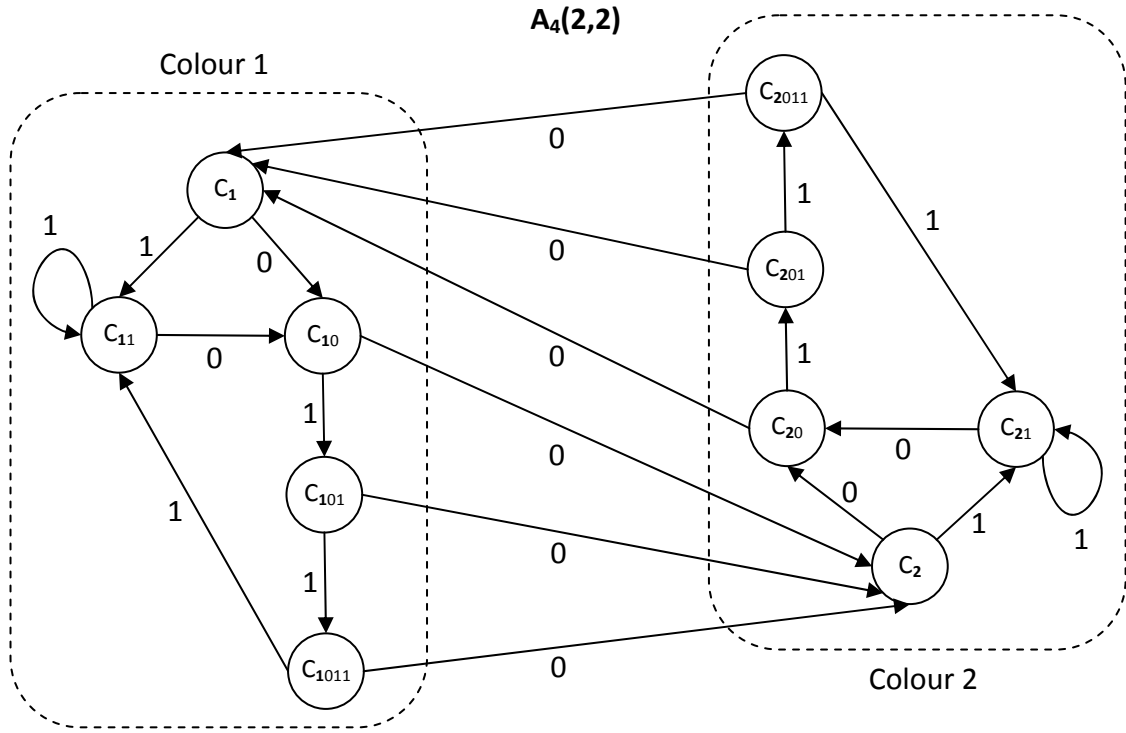


Figure 7: Automaton for $A_4(2,2)$

A state c_{ij} is characterized by 2 elements, i represents the colour and j represents the ‘failure sequence’. A failure is indicated by a 0-arrow and a success by a 1-arrow. These can be replaced with the probabilities p_i in case of a success and $(1-p_i)$ in case of a failure.

Again, by finding the stationary distributions of the transition matrices of the Markov chains above the frequencies of visits per colour can be computed.

For a single-bee situation we expect to be able obtain matching, but the variables and settings have to be chosen carefully. This is also a belief for a multi-bee situation, where an Ideal Free Distribution can be obtained if the variables are chosen carefully, but just in a couple of cases.

By simulating we try to resolve the matching and an Ideal Free Distribution by testing with a number of cases and variables. These simulations will be discussed in chapter 4 and the results accompanying these simulations will be discussed in chapter 5.

Combination of ε -Sampling Strategy and Failures Strategy

Introduction

After reading and discussing the paper, we thought it may be interesting to combine both of the strategies and see what the results of this combined strategy are. Take some of the characteristics of both strategies and merge them into a new strategy. These features are:

- The sampling factor ε of the ε -Sampling Strategy
- Change colour after a certain number of consecutive failures

The ε -Failures Strategy

This strategy can be described in common words as: A bee will use the ε -Sampling Strategy, but after a certain number of failures (this can be consecutive of a number n failures in the last m visits), the bee will move to another colour.

This can be viewed as a special case of the dynamic ε -Sampling Strategy, where ε becomes 1 after a number of failures, otherwise it will remain the same. The dynamic ε function can be defined as a step function, for instance:

$$\begin{aligned}\varepsilon(0) &= \varepsilon_0, \varepsilon_0 \in (0,1) \\ \varepsilon(1) &= \varepsilon(0) \\ \varepsilon(t + 1) &= \begin{cases} \varepsilon(0), & \text{if } k(t) \leq K \\ 1, & \text{otherwise} \end{cases}, \text{ for } t > 1\end{aligned}$$

where K stands for the maximum number of failures which may be perceived before leaving the patch and visit another colour.

The variable k is a perspective of the number of failures and successes and some of them already have been discussed in a section '*Dynamic ε -Sampling Strategy*'. This number can be defined in 3 ways, (i) from the beginning of the simulation, (ii) from the last success and (iii) using a window of last visited flowers. The last method is already introduced in the section '*Extending the Failures time window*'.

After simulating the dynamic ε -Sampling Strategy (and before creating this strategy) we concluded that using the first method for $k(t)$, ε became too much of a disruptive factor in the results. The frequency of visits was matching the probability of a successful visit (with reward of 1 unit of nectar) and matching would occur, but the way it worked was not according to the ε -Sampling Strategy. We chose to use the second manner, because this is more of a combination of both basic strategies. The last method is already an extension of the Failures Strategy and we would like to combine only the basics of both strategies and see and analyze if this 'new' strategy will produce good and interesting results.

3. Simulation

Introduction

The previous chapter discussed the strategies and their adjustments and extensions. In this chapter the simulations, carried out in order to obtain the results, are described.

The need for the simulations is quite clear, because the strategies (and their results) can be compared to each other. We chose for experimenting and simulating instead of a theoretical comparison, because in this way results can be compared in an orderly overview with simulated data.

From a given baseline, a simulation shows how the situation is changing and evolving in the course of time. With the results of the simulations, the strategies can be compared. These results are given in different ways, as numbers and graphs. With the help of these numbers and graphs, we can try to explain the behaviour of bees according to the different characteristics of the strategy.

With a simulation every single action of a bee can be watched and examined. In this way the behaviour of bees, using a certain strategy, can be explained and understood in a better way. This holds also for understanding the effect of a strategy.

Simulations

All of the strategies are modeled and implemented using Matlab. The choice for Matlab is simple, because it's a very extended mathematical program and easy to work with.

At first we started by implementing the ϵ -Sampling Strategy and Failures strategy and creating a user interface to represent the results. The initial idea was to create a tool, which could simulate and compare both of the basic strategies and their extensions. By the hand of this tool also individual flowers and bees could be viewed separately. But instead of focusing on particular data, we wanted to visualize and compare almost every little detail. This was not working very well, also because of the lack in knowledge in creating a usable user interface in Matlab. Creating different graphs and charts, which were quite nice looking and gave a lot of information, took an enormous amount of time and slowed the total simulation down. Furthermore the program became too complex, because there were a lot of variables which had to be set before starting a simulation. Also all of the different kinds of representations of the data were little confusing, because almost everything could be viewed apart.

After experiencing the slow speed of the simulations, mostly because of creating the plots, we decided to skip a nice looking interface and to focus on the data of the results of the simulations (and thus strategies). The raw data could always be transformed and represented by graphs and charts afterwards. This worked out right and we got the results quicker and were able to do more simulations in period of time.

Matching Test for ε -Sampling Strategy

Thus for ε -Sampling Strategy we started simulating with 1 bee, 2 flowers of different colours with probabilities of giving a reward of 1 unit of nectar of respectively $p_1 = 0.3$ and $p_2 = 0.7$, $\varepsilon = 0.05$ and $t = 1.000.000$.

We introduce t as the number of steps we simulate. Each step can be interpreted as a moment in time. In the case of the ε -Sampling Strategy a bee does some actions per step t :

- First the makes a decision to stay (with probability $1 - \varepsilon$) in the current patch or sample a patch of a different colour (with probability ε).
- After the decision to stay or sample, the bee will visit a flower of the current or sampling colour and remember the pay-off.
- In case of sampling, the bee will decide to stay in the new patch or return to the original colour, according to the payoff it received.
- In both cases the critical level is updated (which is not necessary, because we used Bernoulli flowers with reward 0 or 1, thus always above or below every critical level $cl \in (0,1)$).

We started simulating the ε -Sampling Strategy for a single-bee situation (test for Matching Law). At the end of the simulation, the results showed the total number of visits per colour. By dividing the number of visits by the total number of steps the average number of visits per colour are computed and this can be compared to the probabilities of giving a reward of 1 unit of nectar of the colours.

We noticed that the sampling visits cause some disturbance in the expected results. This disturbance is greater on the side of the colour with the smallest 'success' probability, which can be explained by the 'error'-part of the quotation $p_1 + p_2 * \varepsilon$. The error-part $p_2 * \varepsilon$ has more influence on the total $p_1 + p_2 * \varepsilon$, because $p_1 < p_2$.

Furthermore we noticed that ε and the number of steps in the simulation had to be chosen wisely. Since we couldn't set the number of steps to ∞ , ε had to be not too close to 0, so (enough) sampling visits were noticed.

Therefore we have to exclude the sampling visits from counting the number of visits per colour, to get a pure proportion of the frequencies of visits per colour. When a bee samples, this will not affect the number of visits of the sampled colour nor the number of visits of the original colour. Not just this, but also the whole step has to be neglected, because otherwise the values of the number of visits divided by the total number of steps will never reach to the flower's probabilities of giving a successful reward (which have to be found, according to the Matching Law). For instance:

After simulating with the following settings ($p_1 = 0.3$, $p_2 = 0.7$, $\varepsilon = 0.05$ and $t = 1.000.000$) we got the following results:

- 286.218 steps on colour 1
- 663.697 steps on colour 2
- 50.113 steps sampled

Then $286.218/1.000.000 \approx 0.286$ and $663.697/1.000.000 \approx 0.664$. These ratios are almost according to the success probabilities, but since we make an error by counting the steps a bee samples, these cannot reach the probabilities.

But after correcting the number of steps, we get $286.218/(1.000.000-50.113) \approx 0.301$ and $663.697/(1.000.000-50.113) \approx 0.699$. These values are almost exact the probabilities of the flowers giving a successfully reward and we can conclude that the bee will match the ‘success’ probabilities of the colours by the frequency of visits per colour.

After we got this right we started simulating with more colours and the results were as expected. We expected that the ε -Sampling Strategy would give the same results for more than 2 colours, thus that the frequency of visits of a colour matches the payoff probability of that colour. The results of the simulations showed that for 3 or more different colours matching could be observed as well.

Matching Test for Dynamic ε -Sampling Strategy

For simulating with the dynamic ε , we had to look at the 2 different ways of computing k .

We started with the first method (counting from the beginning) and noticed that the results were almost equal to the results of the basic strategy. The frequency of the visits per colour matched the probability of getting a positive payoff per colour, but after taking a closer look of the results, we noticed that the final value of ε (at the end of the simulation) is very dependant of the payoff's probability. If the sum P of the chosen probabilities ($p_1 + p_2 + \dots + p_n$) is smaller than 1, then ε becomes close to 1. Otherwise ε will stay 0.05 until the end of the simulation. We can explain this by the fact that if $P < 1$, a bee will encounter more empty flowers than full flowers during the simulation, thus k is expected to become larger instead of staying 0 (according to the function for k). When $P \geq 1$, the bee is expected to visit more full flowers than empty flowers and this leads to the value for k staying 0.

When simulating long enough ($t > 25.000$) matching will occur, but this is due to the fact that:

- In the case of $P < 1$, since k will, ε will increase close to 1. In the case of ε close to 1, the bee will sample almost every step. The sampling steps have no effect on the matching, because these steps are disregarded. From the time until ε becomes too great, the bee will behave according to the basic ε -Sampling Strategy (since ε is small enough). So only the ‘good’ steps are counted and this will match.
- In the case of $P \geq 1$, the dynamic epsilon ε isn't really dynamic. Since k will be 0, ε will stay $\varepsilon(0)$ and the bee will behave as if it uses the basic ε -Sampling Strategy and this is proven to match.

Because of this effect, the first method isn't very usable. We want to have a robust function, which will perform well under all kinds of situations and not only under well formed situations. Although it may look that this strategy show a good matching behaviour, it doesn't meet our requirements.

After this we decided to try method 2 (set the number of failures back to 0 after getting a positive payoff) for the simulations. After simulating and analyzing the results, we could conclude that this method was more appropriate. Since k is set back to 0 after visiting a full flowers, k won't be growing (in the case of $P < 1$) steadily or stay 0 (otherwise). Hence ε will not remain the same ($\varepsilon(0)$) or be near to 1.

There are some comments on the second method. First it doesn't perform as well as the basic strategy. Matching can be perceived, but not as exact as the matching in the basic ε -Sampling strategy. If the successful payoff probabilities of the different colours are (almost) the same the extent of matching will be better and more according to the basic strategy. When the successful payoff probabilities differ more, the error noticed in the extent of matching will be greater at the colour with the smallest probability.

This could be explained by the 'error'-part in the number of visits per colour. For 2 colours the number of visits on colour 1 is $p_1 + p_2 * \varepsilon$, the 'error'-part is $p_2 * \varepsilon$. When $p_1 < p_2$ and ε close to 0, then $p_1 + p_2 * \varepsilon < p_2 + p_1 * \varepsilon$ and $p_2 * \varepsilon$ has more influence on $p_1 + p_2 * \varepsilon$ then $p_1 * \varepsilon$ influences $p_2 + p_1 * \varepsilon$. If $p_1 = p_2$, then the error will be the same on both sides.

But since we exclude these sample steps from the computation of the frequency of the visits, there has to be some reason. This happens to be an effect of the dynamic ε function. A bee will sample more often when it visits a colour with small payoff, because the sample factor ε increases faster. Furthermore, the probability of staying at the sampled colour is bigger, because the probability of receiving a better payoff at the sampled colour is bigger, and hence the matching will be disturbed.

IFD Test for ε -Sampling Strategy

This simulation was more difficult because of the assumptions, for instance all of the nectar in a patch should be distributed over the bees visiting that patch. Not all assumptions were taken into account for the implementation. We implemented this strategy very simple. There are F_i flowers in patch i and they all are full. A bee will choose to stay in the current patch or sample (with probability ε) another. When a bee visits a patch it will go to a flower. If no flowers are free (thus all the flowers are taken by another bee), it will remain in the patch, but accidentally do nothing, there is no payoff for the bee. At the end of each stage the number of bees per patch B_i is counted and then the ratio F_i / B_i is computed.

Originally we showed all these ratios of the different colours in one graph, but this turned out to be confusing. The lines in the graph looked very shaky and it looked like the strategy wasn't behaving as expected. But this was because of the fact that we computed the wrong results. After adjusting the simulation and the scale of the graph, the results were better. But our expectations, the population would need some time to come to a stable Ideal Free Distribution and stay distributed in that way, weren't met. Instead we saw the distribution of the population vary a lot and after getting an Ideal Free Distribution the population wouldn't stay distributed in that way. This can be justified by the numbers we used for the variables. Because the speed of the simulation we chose not as much steps, bees and flowers as desired, since the simulation would crash or take too much time. If the number of steps would be much more, the influence of ε would be smaller and a particular distribution would hold longer.

For comparing the results we used the number of bees (n_i) per colour and the number of flowers per colour (f_i). Bees, which are sampling during a specific time step aren't taken into account for this numbers, because they haven't decided to stay at or leave a patch. For each colour a line is plotted with the number of bees at that colour, divided by the number of flowers of that colour per time step (n_i / f_i). In that way we can how the bees distribute themselves over the available flowers. If these fractions are equal, an Ideal Free Distribution is observed. In our simulations this wasn't observed very often, but the fraction were close to each other, so we can say that the population of the bees is almost distributed according to the Ideal Free Distribution. Furthermore we computed the average number of these fractions, to see if, over all the steps simulated, the 'average' distribution will be according to the Ideal Free Distribution. During the

Matching test the sampling stages of the single bee weren't counted to neglect the sampling stages of the bee and get no disturbed results. This wasn't possible for the IFD test, because this test consists of more bees. If we would neglect the stages where one (or more) bee(s) would sample, too much steps would have to be neglected. When the number of flowers per colour is (almost) equal to each other, the disturbing effect hasn't much influence on the results, but if they differ (a lot), the disturbing effect is noticeable, especially on the side of the smallest patch.

IFD Test for Dynamic ϵ -Sampling Strategy

This simulation is the same as the simulation for the basic ϵ -Sampling Strategy. The only part we had to adjust was the value for ϵ . Both of the methods for adjusting ϵ were implemented during the test for matching. Because we already simulated the IFD test for the basic strategy, we could use the observations and settings for this simulations.

Matching Test for Failures Strategy

This strategy was implemented on the basics of the ϵ -Sampling Strategy. But instead of sampling a colour with probability ϵ , the bee counts the number of failures (empty flowers) and moves to another colour if this number becomes high enough.

To take a look of the actions of a single bee during one stage:

- The bee visits a flower of certain colour and examines the payoff.
 - If the payoff is 0, the bee will add 1 to the number of failures.
- The bee checks if the total number of failures has become high enough.
 - If the total number of failures has reached a certain number, the bee will move to another colour at the next stage and the number of failures will be set back to 0.
 - Otherwise the bee will stay at the current patch

For the matching test we'll use a single bee and one single flower per colour with a certain probability of giving a reward of 1 unit of nectar (a success). During the testing of the simulations, we saw that when the total number of failures before moving to another patch are equal for each colour, the bee will match its frequency of visits per colour to the probability of getting a success on that colour.

This was also noticed if more than 2 colours were used in the simulation. We didn't do a lot of research on the Failures strategy with more than 2 colours, ie. $A(c_1, c_2, \dots, c_i)$, because the computing of the values for matching and an Ideal Free Distribution is more difficult and finding the correct values for all the variables is hard. But the implementation was able to use more than 2 colours, therefore we tested a little with this simulations to see how the strategy would behave.

But when the probabilities of getting a success on a colours were chosen different, it became difficult to find the correct value for the number of failures per colour, after which a bee will move to another colour and obtain matching. This was, as expected, very difficult when the number of colours would be more than 2.

The results of the simulations will be discussed in the next chapter.

Matching Test for Failures Time Windows Strategy

The implementation of this strategy is almost the same as the basic strategy. The only thing a bee would have to check is the number of failures in the last couple of steps, different to only counting consecutive failures.

Therefore we added a certain memory for the bee, so the bee remembers the last m visits. Then if the number n (number of failures) becomes too high in the last m visits, a bee will leave and move to another colour.

From the results of the simulations with equal probabilities of receiving a successful payoff at the colours, we perceived the same results as seen by the basic Failures strategy. But a mistake was found in the implementation. The ‘empty’ history of a bee wasn’t updated in the right way after a change of colour. By this mistake it was possible that failures, a bee encountered in the previous colour, would be counted by the failures at the current colour.

After adjusting implementation of the simulation, the simulation was a good representation of the extended Failures strategy. These results will be explained in the next chapter.

Testing the ε -Failures Strategy

We implemented this strategy by taking the implementation of the ε -Sampling Strategy and the Failures strategy and combine these into a new simulation. The basic of this strategy consists more out of the ε -Sampling Strategy and a counter for the number of failures. After some number of consecutive failures, the bee is forced to move to another colour, instead of staying at the patch or sampling.

As said the main part for this simulation is copied from the implementation of the ε -Sampling Strategy and added a part for switching colours after some number of failures.

Because the tests for Matching and IFD are already explained before for both basic strategies, these won’t be discussed here and we’ll move on the next chapter with the results.

4. Results

Introduction

In this chapter the results, derived from the experiments, are described. First we give an outline of the configuration of the experiment, thereafter the obtained results will be explained. The experiments and results are categorized by strategy and particular instance of a strategy. The first simulations will be explained more extended, since they are the first to be described.

ϵ -Sampling Strategy

At first we start with the results of the simulations of the basic ϵ -Sampling Strategy, as described in the paper which serves as the basic assumption for this thesis. The simulation will be divided in 2 instances, (i) a single-bee setting and (ii) a multi-bee setting. The single-bee setting will be used to check if a solitary bee, using the ϵ -Sampling Strategy, behaves according to the Matching Law. The multi-bee setting is simulated to check if the population will stabilize in the Ideal Free Distribution.

Basic single-bee setting [test for Matching Law]

For this simulation we used the following setup [the probability of a flower giving a reward of 1 unit nectar is mentioned by the term *full chance*]:

- 1 bee
- 1 flower of colour Y, full chance $y = 0.3$
- 1 flower of colour B, full chance $b = 0.7$
- 10.000.000 time steps
- $\epsilon = 0.05$

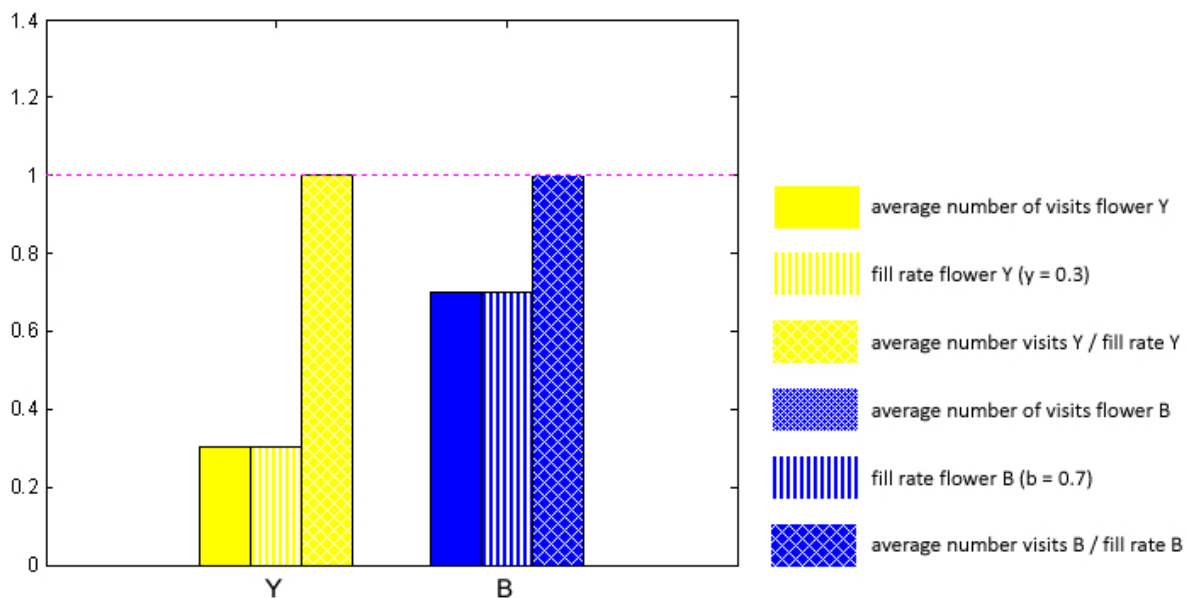


Figure 8

As we can see from the figure above, the bee will actually behave according to the Matching Law. The bars on the right represent the average number of visits divided by the full chance per colour (we call this the *Matching Factor*), and when this number is 1 (see the dotted line in the figure), the bee matches exactly its visits to the full chances.

The average number of visits per colour is almost equal to the full chance of the particular colour, as can be noticed from the following table.

| Colour | Full chance | Average number of visits | Matching Factor |
|--------|-------------|--------------------------|-----------------|
| Y | 0.3 | 0.3009 | 1.0029 |
| B | 0.7 | 0.6991 | 0.9988 |

Table 1

From 2 to n flowers in a single-bee setting [test for Matching Law]

We wanted to test if the assumptions from the basic paper, can be also applied to more than 2 colours. So we extended the number types of flowers in the simulation to 3 and ran the simulation again, and the setup is as following.

- 1 bee
- 1 flower of colour Y, full chance $y = 0.1$
- 1 flower of colour B, full chance $b = 0.3$
- 1 flower of colour R, full chance $r = 0.6$
- 10.000.000 time steps
- $\varepsilon = 0.05$

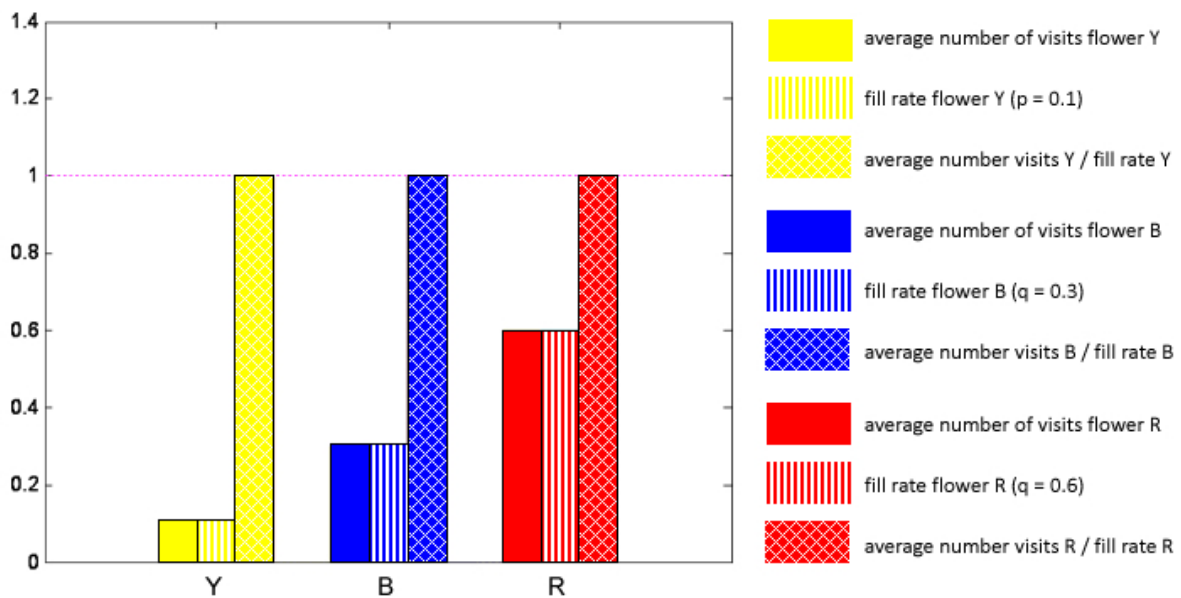


Figure 9

| Colour | Full chance | Average number of visits | Matching Factor |
|--------|-------------|--------------------------|-----------------|
| Y | 0.1 | 0.0995 | 0.9992 |
| B | 0.3 | 0.3003 | 1.0009 |
| R | 0.6 | 0.6002 | 1.0004 |

Table 2

The outcome of the simulation is as expected. The bee will visit the flowers in those proportions given by the full chances. Because the sum of the full chances is equal to 1 ($0.1 + 0.3 + 0.6 = 1$), the average number of visits is (almost) equal to the full chance.

If the sum of the full chances is greater than 1, a *normalized full chance* is used to see if the bee behaves in terms with the Matching Law. The normalized full chance can be interpreted as a weighted full chance and is described as:

$$\text{normalized full chance } nfc_i = \frac{c_i}{\sum_{i=1}^{nc} c_i}$$

where c_i = full chance of colour i ,
 nc = number of colours

This situation is demonstrated by the simulation with the following setup:

- 1 bee
- 1 flower of colour C_1 , full chance $c_1 = 0.1$, normalized full chance $nfc_1 = 0.06$
- 1 flower of colour C_2 , full chance $c_2 = 0.2$, normalized full chance $nfc_2 = 0.11$
- 1 flower of colour C_3 , full chance $c_3 = 0.2$, normalized full chance $nfc_3 = 0.11$
- 1 flower of colour C_4 , full chance $c_4 = 0.5$, normalized full chance $nfc_4 = 0.28$
- 1 flower of colour C_5 , full chance $c_5 = 0.8$, normalized full chance $nfc_5 = 0.44$
- 10.000.000 time steps
- $\varepsilon = 0.05$

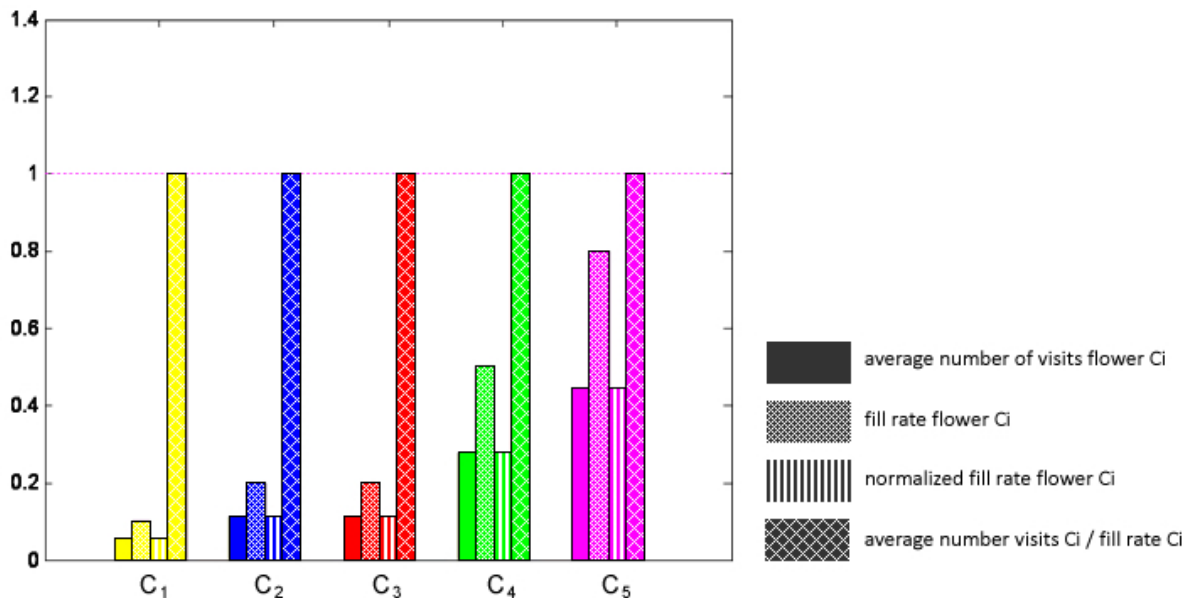


Figure 10

| Colour | Full chance | Normalized Full chance | Average number of visits | Matching Factor |
|----------------|-------------|------------------------|--------------------------|-----------------|
| C ₁ | 0.1 | 0.0556 | 0.0551 | 0.9926 |
| C ₂ | 0.2 | 0.1111 | 0.1113 | 1.0014 |
| C ₃ | 0.2 | 0.1111 | 0.1111 | 1.0000 |
| C ₄ | 0.5 | 0.2778 | 0.2780 | 1.0009 |
| C ₅ | 0.8 | 0.4444 | 0.4445 | 1.0000 |

Table 3

From the results above, it can be noticed that if the bee uses the ε -Sampling strategy, the frequency of the visits of the different colours matches the normalized full chance (the Matching Law) and this also holds for situations of more than 2 different colours.

Basic multi-bee setting [test for Ideal Free Distribution]

Because of the assumptions made in the basic paper, this simulation was more difficult to implement. The number of bees and the total number of flowers had to be chosen wisely to accomplish an Ideal Free Distribution.

For good and understandable results, the number of simulation steps has to be large and the number of bees has to be approximately the same as the total number of flowers. We'll explain this by the simulations. The results will be shown in tables, since the graphs are less informative.

If we show a graph of the ratios $nb_1/q_1 = nb_2/q_2 = \dots = nb_i/q_i$ per step, this will look like:

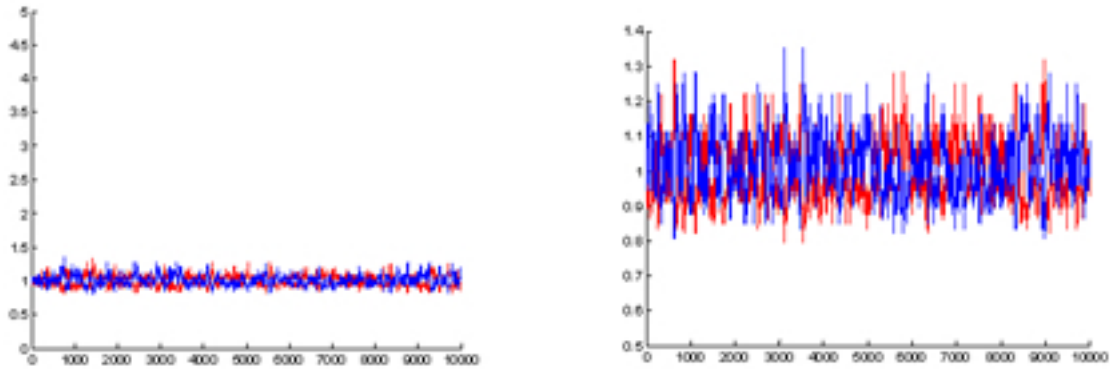


Figure 11: Example of 2 graphs with the ratios for colour 1 (red line) and colour 2 (blue line)

These graphs are not very useful. Sure we can see that the ratios are almost equal, but we cannot see the stages with an Ideal Free Distribution.

For comparing the results of the simulations we count the average fractions $nb_1/q_1, nb_2/q_2, \dots, nb_i/q_i$ for each colour, this value will be called average IFD rate A_i . This can be defined as the average fraction of bees per colour (the number of bees per colour divided by the number of flowers of that colour). The closer these numbers are to 1, the more the population will be distributed according to the Ideal Free Distribution.

We will start with the simulations of 2 different colours and 10 bees and increase the number of steps. For each number of A_i we took the average value of a 10 simulations.

| Steps | Bees | Colour 1 | Colour 2 | ϵ | A_1 | A_2 |
|-----------|------|----------|----------|------------|--------|--------|
| 1.000 | 10 | 5 | 5 | 0.05 | 0.9987 | 1.002 |
| 10.000 | 10 | 5 | 5 | 0.05 | 0.9998 | 0.9997 |
| 100.000 | 10 | 5 | 5 | 0.05 | 0.9997 | 1.0003 |
| 1.000.000 | 10 | 5 | 5 | 0.05 | 0.9998 | 1.0001 |

Table 4

We noticed that in the average fraction of bees per colour is (almost) equally distributed over the available colours and flowers, hence on average the bees will be distributed according to the Ideal Free Distribution. The fractions aren't equal to 1, because of the disturbing effect of the sampling bees.

When simulating, the simulation of 100.000 and 1.000.000 steps took quite a long time to perform and for the other simulations this big simulation will be skipped, because the results look the same for smaller simulations.

We wanted to see the influence of ϵ , the results of the simulations with different values for ϵ are given in the following table:

| Steps | Bees | Colour 1 | Colour 2 | ϵ | A_1 | A_2 |
|--------|------|----------|----------|------------|--------|--------|
| 10.000 | 10 | 5 | 5 | 0.01 | 0.9982 | 1.0125 |
| 10.000 | 10 | 5 | 5 | 0.05 | 0.9998 | 0.9997 |
| 10.000 | 10 | 5 | 5 | 0.10 | 0.9984 | 1.0024 |
| 10.000 | 10 | 10 | 10 | 0.01 | 0.5029 | 0.4972 |
| 10.000 | 10 | 10 | 10 | 0.05 | 0.5007 | 0.4988 |
| 10.000 | 10 | 10 | 10 | 0.10 | 0.5042 | 0.4978 |
| 10.000 | 10 | 8 | 12 | 0.01 | 0.6104 | 0.4276 |
| 10.000 | 10 | 8 | 12 | 0.05 | 0.6220 | 0.4265 |
| 10.000 | 10 | 8 | 12 | 0.10 | 0.6299 | 0.4258 |

Table 5

The results of the simulation show that the influence of ϵ on A_i is noticeable. But this is due to the fact that the number of steps is too little. When the number of steps will increase, the effect of the samples can be neglected and this will hold for any ϵ close to 0. On 10.000 steps, with $\epsilon = 0.01$, a bee will sample 10 times on average, but when $\epsilon = 0.1$, a bee will sample 100 times on the average. So, with such little steps, the influence of ϵ is noticeable.

We also see that if the total number of flowers is greater than the number of bees, the average number of bees per colour won't be 1. This can easily be explained by the fact that there are too much flowers and some flowers won't be visited during one step, since there are not enough bee to visit all the flowers in one step. But for an IFD, they have to be (almost) equal, as can be seen in the simulations with 10 bees and 10 flowers on both colours. The ratios A_1 and A_2 are almost equal. But when the total number of flowers is greater than the number of bees, but the number

of flowers at a patch will be less than the number of bees, the bees won't be distributed according to the Ideal Free Distribution. This can be described to the fact the bees will visit the smallest patch too often compared to the patch with the most flowers. We will try to explain this with the next simulations.

Next we look at the value of A_i when the number of flowers per colour is changed:

| Steps | Bees | Colour 1 | Colour 2 | ϵ | A_1 | A_2 |
|--------|------|----------|----------|------------|--------|--------|
| 10.000 | 10 | 5 | 5 | 0.05 | 0.9998 | 0.9997 |
| 10.000 | 10 | 2 | 4 | 0.05 | 1.8598 | 1.5707 |
| 10.000 | 10 | 3 | 7 | 0.05 | 1.1645 | 0.9286 |
| 10.000 | 10 | 2 | 2 | 0.05 | 2.5371 | 2.4607 |
| 10.000 | 10 | 1 | 7 | 0.05 | 1.7302 | 1.1825 |
| 10.000 | 10 | 3 | 6 | 0.05 | 1.2423 | 1.0446 |
| 10.000 | 10 | 10 | 10 | 0.05 | 0.5007 | 0.4988 |
| 10.000 | 10 | 5 | 15 | 0.05 | 0.9251 | 0.3580 |
| 10.000 | 10 | 8 | 12 | 0.05 | 0.6220 | 0.4265 |
| 10.000 | 10 | 7 | 13 | 0.05 | 0.7031 | 0.3908 |
| 10.000 | 10 | 10 | 20 | 0.05 | 0.4907 | 0.2550 |
| 10.000 | 10 | 50 | 50 | 0.05 | 0.1022 | 0.0987 |
| 10.000 | 10 | 100 | 100 | 0.05 | 0.0492 | 0.0507 |

Table 6

From table 6, we notice that if the total number of flowers is equal to the number of bees the average distribution of the bees will be (almost) as if it's in an Ideal Free Distribution. This is also observed if the number of flowers per colour is the same.

If the number of bees is greater than the total number of bees, the results show that the bees will be almost distributed as in an Ideal Free Distribution. If food sources (flowers) are scarce, the results show that bees using the ϵ -Sampling Strategy will divide themselves (on average) over the available sources, following the Ideal Free Distribution.

But when the total number of flowers greater than the number of bees and the number of flowers per colour differ, the average distribution won't be according to the Ideal Free Distribution. If we compute Colour 1 * A_1 and Colour 2 * A_2 , we note that these values will both be close to 5 ($7*0.7031 = 4.92$ and $13*0.3908 = 5.08$). So the bees will distribute themselves equally over the patches (as in 50% on patch 1 and 50% on patch 2), whatever the number flowers in the patches are, if the total number of flowers is greater than the number of bees.

It doesn't matter what flower a bee chooses, since there is always a full flower available. Hence the bees will be distributed equally over the patches.

If we simulate with more bees and more flowers, the results will be the same as the previous results as we can see in table 6. Only the numbers will be closer to the expected values since we use the average value of more bees and more flowers (this will also hold for more steps).

| Steps | Bees | Colour 1 | Colour 2 | ϵ | A_1 | A_2 |
|--------|------|----------|----------|------------|--------|--------|
| 10.000 | 100 | 50 | 50 | 0.05 | 1.0029 | 0.9973 |
| 10.000 | 100 | 100 | 100 | 0.05 | 0.5009 | 0.4983 |
| 10.000 | 100 | 20 | 50 | 0.05 | 1.5751 | 1.3688 |
| 10.000 | 1000 | 500 | 500 | 0.05 | 1.0009 | 1.0007 |
| 10.000 | 1000 | 1000 | 1500 | 0.05 | 0.4995 | 0.3341 |

Table 7

From 2 to n flowers in a multi-bee setting [test for IFD]

Next we will expand the number of colours and test if the results are the same for this simulations.. The results of the simulations with 3 colours are shown hereafter:

| Steps | Bees | Colour 1 | Colour 2 | Colour 3 | ϵ | A_1 | A_2 | A_3 |
|--------|------|----------|----------|----------|------------|--------|--------|--------|
| 10.000 | 15 | 2 | 4 | 6 | 0.05 | 1.4453 | 1.2753 | 1.1694 |
| 10.000 | 15 | 5 | 5 | 5 | 0.05 | 1.0045 | 0.9913 | 1.0062 |
| 10.000 | 15 | 2 | 5 | 8 | 0.05 | 1.3348 | 1.0801 | 0.8616 |
| 10.000 | 15 | 3 | 6 | 6 | 0.05 | 1.1906 | 0.9499 | 0.9529 |
| 10.000 | 15 | 5 | 10 | 15 | 0.05 | 0.9041 | 0.5377 | 0.3343 |
| 10.000 | 15 | 10 | 10 | 10 | 0.05 | 0.5021 | 0.4988 | 0.5014 |
| 10.000 | 15 | 10 | 20 | 30 | 0.05 | 0.5002 | 0.2502 | 0.1677 |
| 10.000 | 15 | 100 | 100 | 100 | 0.05 | 0.0497 | 0.0502 | 0.0499 |
| 10.000 | 150 | 35 | 50 | 65 | 0.05 | 1.1058 | 1.0283 | 0.9225 |

Table 8

It can be seen if the total number of flowers is equal to the number of bees or the number of flowers per patch is equal an average Ideal Free Distribution is observed, just as we saw in the situation with 2 types of flowers.

But the more the numbers of flowers per patch differ, the more disturbance in the ratios A_i . Since we don't simulate an infinite amount of time, there is a disturbance, caused by ϵ and this disturbance is the greatest at the patch with the smallest number of flowers. This disturbance will be less if the number of bees, number of flowers and number of steps will be greater.

In the case of food scarcity (more bees than flowers), the results match with the results found for the less complex situations with only 2 types of flowers.

If the number of bees is smaller than the total number of flowers, also the same distribution as in the less complex situation with 2 types of flowers is noticed. Thus the bees will distribute themselves equally over the patches, such that at each patch there are an equal number of bees. If we have 15 bees and 3 different flower types, then there will be an average of 5 bees per colour per step.

Dynamic ϵ -Sampling Strategy

For this strategy we will discuss both of the methods as stated in comparison to the basic strategy, firstly focusing on the matching test. The results for an Ideal Free Distribution will be discussed at the end of this section.

Basic single-bee setting [test for Matching Law]

For this simulation we used the following setup:

- 1 bee
- 1 flower of colour Y , full chance $y = 0.3$
- 1 flower of colour B , full chance $b = 0.7$
- 1.000.000 time steps
- $\epsilon = 0.05$

First we show the results for the first method (counting from the beginning of the simulation) illustrating it's behaviour.

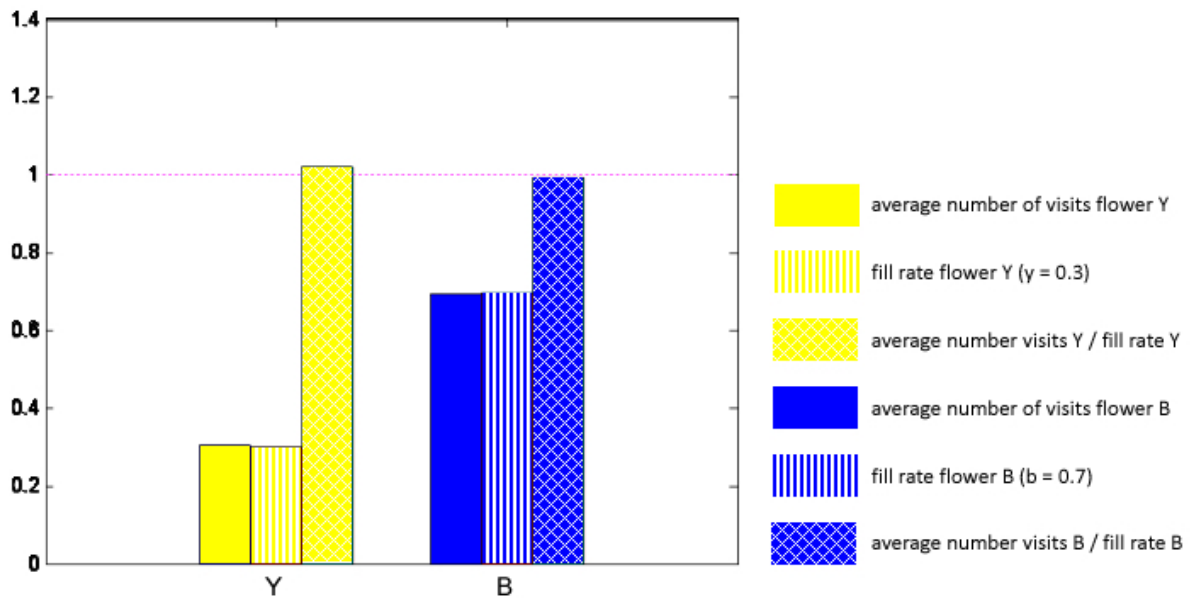


Figure 12

| Colour | Full chance | Average number of visits | Matching Factor |
|--------|-------------|--------------------------|-----------------|
| Y | 0.3 | 0.3064 | 1.0212 |
| B | 0.7 | 0.6936 | 0.9909 |

Table 9

The results of the second method (set number of failures back to 0 after a success) are shown below:

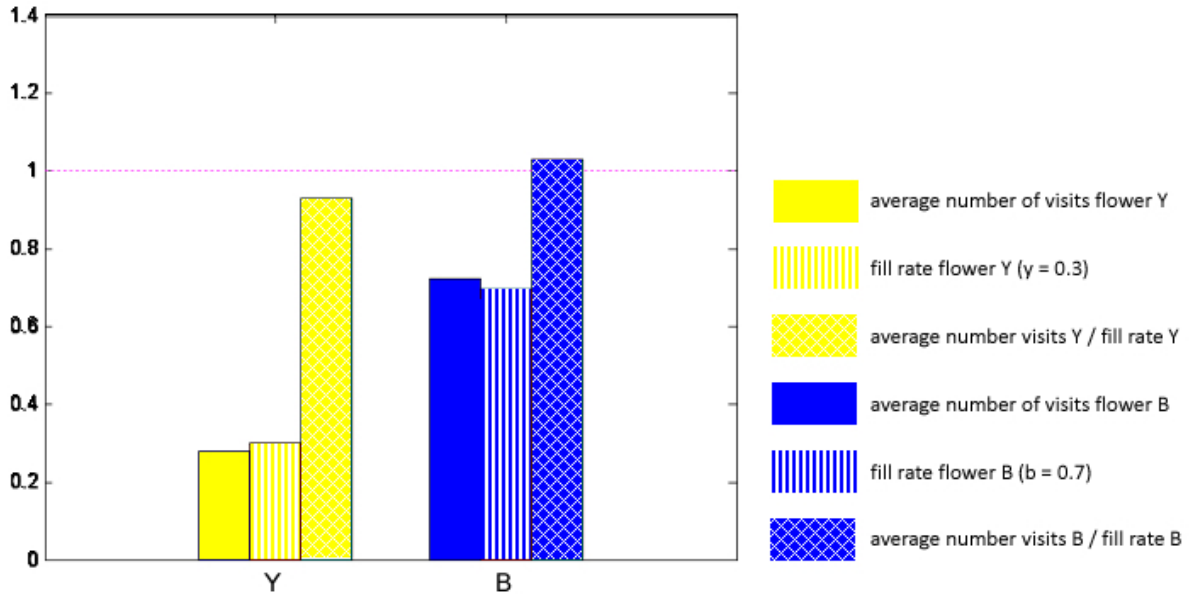


Figure 13

| Colour | Full chance | Average number of visits | Matching Factor |
|--------|-------------|--------------------------|-----------------|
| Y | 0.3 | 0.2795 | 0.9316 |
| B | 0.7 | 0.7205 | 1.0293 |

Table 10

As we can see from the figures and tables above, the bee will almost behave according to the Matching Law and the performance of the first method looks better. But as stated in chapter 3, this method acts a little more complicated as can be perceived from the results. Note that in this case the bee encountered 549934 successes and 399931 failures, thus before the value of ϵ is changing there have to be at least 150003 more consecutive failures. For the rest of this section we will use method 2.

From 2 to n flowers in a single-bee setting [test for Matching Law]

We extend the number types of flowers in the simulation to 5 and ran the simulation again, and the setup is as following.

- 1 bee
- 1 flower of colour C_1 , full chance $c_1 = 0.1$, normalized full chance $\eta c_1 = 0.06$
- 1 flower of colour C_2 , full chance $c_2 = 0.2$, normalized full chance $\eta c_2 = 0.11$
- 1 flower of colour C_3 , full chance $c_3 = 0.2$, normalized full chance $\eta c_3 = 0.11$
- 1 flower of colour C_4 , full chance $c_4 = 0.5$, normalized full chance $\eta c_4 = 0.28$
- 1 flower of colour C_5 , full chance $c_5 = 0.8$, normalized full chance $\eta c_5 = 0.44$
- 10.000.000 time steps
- $\epsilon = 0.05$

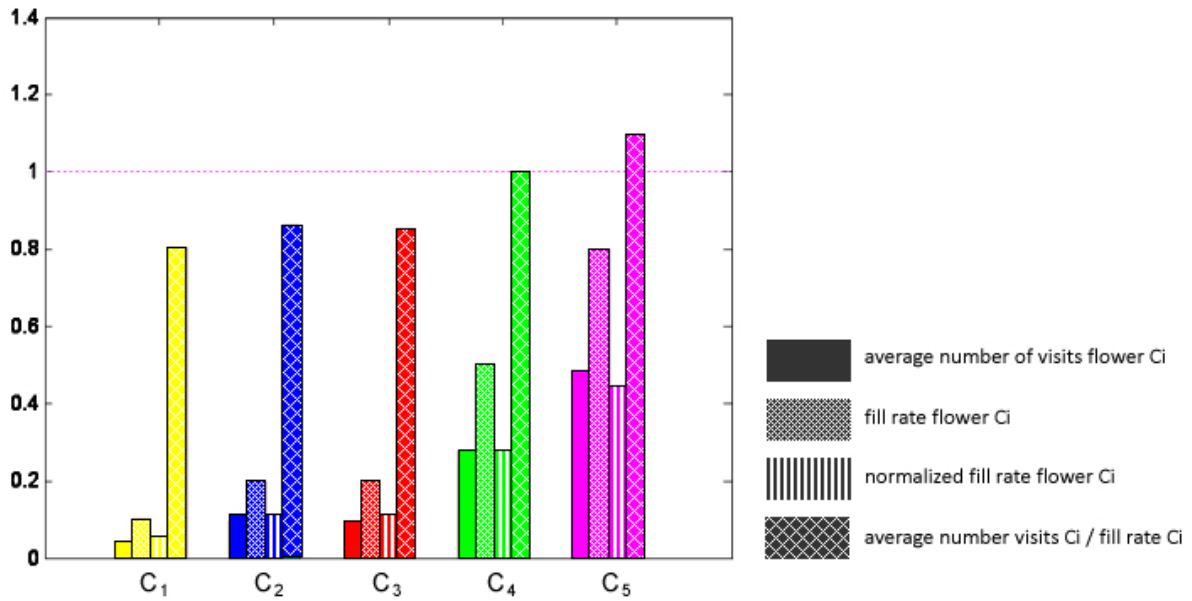


Figure 14

| Colour | Full chance | Normalized Full chance | Average number of visits | Matching Factor |
|----------------|-------------|------------------------|--------------------------|-----------------|
| C ₁ | 0.1 | 0.0556 | 0.0446 | 0.8032 |
| C ₂ | 0.2 | 0.1111 | 0.0958 | 0.8618 |
| C ₃ | 0.2 | 0.1111 | 0.0945 | 0.8509 |
| C ₄ | 0.5 | 0.2778 | 0.2780 | 1.0010 |
| C ₅ | 0.8 | 0.4444 | 0.4870 | 1.0958 |

Table 1

The simulations show that there will be less matching when using the dynamic ε -Sampling strategy. This can be explained by the fact that ε has to be close to 0 for acquiring the Matching Law. If ε will increase and will become close to 1, the basic principle of the strategy is undermined. This will cause the bee to sample more often when it visits flowers with low full chances then when the bee visits flowers with high full chance, because the probability of encountering a failure will be less and ε won't increase.

Dynamic ε -Sampling Strategy in a multi-bee setting [test for IFD]

We want to compare the results of this extension to the basic strategy, therefore we have to choose the same settings of the simulations of the basic strategy and apply these settings to the extended strategy.

| Steps | Bees | Colour 1 | Colour 2 | ε | A_1 | A_1 dynamic | A_2 | A_2 dynamic |
|--------|------|----------|----------|---------------|---------------|---------------|---------------|---------------|
| 10.000 | 10 | 5 | 5 | 0.05 | <i>0.9982</i> | 1.0350 | <i>1.0125</i> | 1.0703 |
| 10.000 | 10 | 3 | 7 | 0.05 | <i>1.1645</i> | 1.2894 | <i>0.9286</i> | 0.9512 |
| 10.000 | 10 | 8 | 12 | 0.05 | <i>0.6220</i> | 0.6648 | <i>0.4265</i> | 0.4340 |
| 10.000 | 10 | 10 | 10 | 0.05 | <i>0.5007</i> | 0.5161 | <i>0.4988</i> | 0.5365 |
| 10.000 | 10 | 50 | 50 | 0.05 | <i>0.1022</i> | 0.1069 | <i>0.0987</i> | 0.1037 |
| 10.000 | 100 | 50 | 50 | 0.05 | <i>1.0029</i> | 1.0511 | <i>0.9973</i> | 1.0541 |
| 10.000 | 100 | 100 | 100 | 0.05 | <i>0.5009</i> | 0.5261 | <i>0.4983</i> | 0.5265 |
| 10.000 | 100 | 100 | 200 | 0.05 | <i>0.4925</i> | 0.5286 | <i>0.2518</i> | 0.2620 |
| 10.000 | 100 | 20 | 50 | 0.05 | <i>1.5751</i> | 1.6780 | <i>1.3688</i> | 1.4341 |

Table 12

From the results in table 12 can be observed that the extension the basic strategy with a dynamic ε gives the same results as the basic ε -Sampling strategy. This can be explained by the fact that the results not really depend on ε . We only have to keep in mind that ε has to be close to 0, so a bee stays long enough in a patch.

In table 13 we show the results for the simulations with more than 2 colours (10.000 steps and $\varepsilon = 0.05$). These results are also in line with the results we observed during the simulations of the basic ε -Sampling Strategy for more than 2 colours. The disturbance by ε is a little bit more, but this happens because the ε is dynamic and will increase (and thus the influence and disturbance increases).

| Colour1 | Colour2 | Colour3 | A_1 | A_{1dyna} | A_2 | A_{2dyna} | A_3 | A_{3dyna} |
|---------|---------|---------|---------------|-------------|---------------|-------------|---------------|-------------|
| 5 | 5 | 5 | <i>1.0045</i> | 1.0711 | <i>0.9913</i> | 1.0470 | <i>1.0062</i> | 1.0397 |
| 2 | 5 | 8 | <i>1.3348</i> | 1.4842 | <i>1.0801</i> | 1.1644 | <i>0.8616</i> | 0.8749 |
| 3 | 6 | 6 | <i>1.1906</i> | 1.2845 | <i>0.9499</i> | 1.0154 | <i>0.9529</i> | 1.0067 |
| 5 | 10 | 15 | <i>0.9041</i> | 0.9422 | <i>0.5377</i> | 0.5448 | <i>0.3343</i> | 0.3754 |
| 2 | 4 | 6 | <i>1.4453</i> | 1.5877 | <i>1.2753</i> | 1.3536 | <i>1.1694</i> | 1.1807 |

Table 7

Failures Strategy

For this strategy we used an approach of trial and error, not only for implementing the strategy, but also for getting the results. It is stated in the paper introducing the basic strategies, that for all full chances $p, q \in [0.22, 0.78]$ one can obtain matching with automata $A(y, b)$ with $y, b \in \{1, 2\}$ and y and b denote the number of (consecutive) failures after which a bee will move to another colour.

Thus by the trial and error approach, we try to find sets of p , q and $A(y, b)$ for which matching occurs. Hereafter we simulate automata $A_n(x, y)$ and try to find values for n which give the same result as $A(x, y)$.

For this simulations we only consider a single-bee situation and try to find those instances where matching occurs, since it is stated only a static result for the existence of an Ideal Free Distribution and no dynamic process on how to reach an Ideal Free Distribution.

Basic single-bee setting $A(x, y)$ [test for Matching Law]

The simulations will be divided in 2 main parts, for automata $A(x, y)$ with $x = y$ and $x \neq y$. In both cases, there were situations which could occur when adding the full chances p and q . But the first case was more understandable and easy to find the right settings.

During the simulations we found out that these situations had their own characteristic results. We will explain this results without the graphs, because they all look like the ones that could be seen before.

$x = y$

- $p + q = 1$ or $p = q$

In this situations matching was seen in every simulation. The fact that instances of $p = q$ show matching is quite understandable, because every colour has the same full chance and is visited the same by the bee. In the other situations $p + q = 1$ the matching can be described to the fact that (since $x = y$) we can forget of the number of failures. It doesn't matter if $x = y = 2$ or $x = y = 20$. Thus we can state that these cases can be seen as $x = y = 1$. Since a bee will leave after encountering 1 failure with probability $1 - p$ (or $1 - q$ otherwise) and we have the requirement of $p + q = 1$, from where we get $1 - p = q$ (and $q = 1 - p$ otherwise). Thus a bee will leave a patch with a probability which is equal to the full chance of the other colour and therefore matching occurs.

- $p + q < 1$

No matching occurred in these instances, but it was noticeable that in every simulation the Matching Factor (the average number of visits divided by the full chance per colour) was above 1 for the colour with the smallest full chance. This can be seen as a bee visiting the flower with the lowest payoff too often. Because the sum of the full chances is smaller than 1, the bee will encounter more failures and will move to another colour more often, but this effect is less on the colour with the lowest full chance. If p and q are close to each other and $p + q$ is close to 1, this effect will be of less influence and the Matching Factors of both colours will be closer to 1.

- $p + q > 1$

For this instances also no matching could be noticed and the opposite effect as noticed before could be seen. In every simulation the Matching Factor (the average number of visits divided by the full chance per colour) was above 1 for the colour with the highest

full chance. Thus a bee will visit the flower with the highest payoff to often. The explanation follows from above, because the sum of the full chances is greater than 1, the bee will receive more successes and stay in the current patch. This effect is of more influence on the colour with the highest full chance. If p and q are close to each other and $p + q$ is close to 1, this effect will be of less influence and the Matching Factors of both colours will be closer to 1.

$x \neq y$

This situations are more difficult to classify, because the influence of the combinations x, y and p, q were very hard to understand. In one kind of simulation ($p = q$) the results were easy to understand.

- $p = q$

Since p and q are equal only the number of failures is has influence on how long a bee will stay in a patch. If x is twice as much as y , a bee will stay twice as long in the patch described by x , as can be seen below.

| Average number of visits | Leave after failures | Full chance | Normalized Full chance | Matching Factor |
|--------------------------|----------------------|-------------|------------------------|-----------------|
| 0.2500 | 3 (x) | 0.4 (p) | 0.5000 | 0.5000 |
| 0.7500 | 6 (y) | 0.4 (q) | 0.5000 | 1.5000 |

Table 8

This holds for the instances where $p + q \neq 1$ and $p + q = 1$.

- $p \neq q$

The simulations showed out that in matching occurs, some setting where matching can been seen are:

| x (patch 1) | y (patch 2) | p | q | MF_1 | MF_2 |
|---------------|---------------|-----|-----|--------|--------|
| 3 | 2 | 0.4 | 0.8 | 1.0053 | 0.9974 |
| 6 | 5 | 0.5 | 0.7 | 1.0006 | 0.9996 |
| 15 | 20 | 0.2 | 0.5 | 1.0030 | 0.9991 |
| 260 | 261 | 0.4 | 0.6 | 0.9939 | 1.0041 |

Table 15

After setting p and q , we adjusted x and y and examined the effect of this adjustment to the Matching Factors. We noticed that if $MF_1 < 1$ (and thus $MF_2 > 1$) we had to increase x . By this adjustment a bee will stay longer in patch 1 (with x and MF_1) and therefore the value of MF_1 will increase. If the ratios are chosen wisely (by trial and error), it can be possible to receive matching in every instance.

Extended Failures Time Window Strategy single-bee setting $A_n(x,y)$ [test for Matching Law]

For this simulations we used the results and settings of the simulations performed in the previous section. In this way, we can see if the results of this extension are the same compared to the basic strategy.

$x = y$

- $p = q$
If p and q are equal, the simulations show that there is matching, n doesn't have any effect on the results. This can be noticed since in this situation the extension can be neglected and thus the explanation given in the previous section can be applied also in this situation.
- $p + q = 1$
If $n = x = y = 1$, we notice matching, since this instance of $A_n(x,y)$ can be interpreted as the basic Failures Strategy. In this case it can be stated that $A_1(x,y) = A(x,y)$. But if $n > 1$, the matching will be disturbed, because the bee will stay too long at a patch. This behaviour wasn't expected, we expected that for every n matching would occur, but since the bee will stay minimally as long as the failures window n , this situation cannot be interpreted as $A_n(x,x) = A(x,x)$.
- $p + q < 1$ and $p + q > 1$
In both of the cases, the extended Failures strategy will act as if the basic Failures strategy. Again, if $n = 1$, the behaviour will exactly match the basic strategy, but when $n > 1$, this effect will be stronger. Since a bee will stay too long in a patch because of 2 reasons, (i) the sum of the full chances isn't 1 and thus a bee will stay too long in a patch (see previous section for explanation) and (ii) a bee will stay at least as long as the failures window.

$x \neq y$

- $p + q = 1$
For this instance, it was far more difficult to see matching. We tried a lot of instances, but in none of them was matching noticed. This can be explained to the fact that the influence of the failures window is stronger on one side. In the case $x \neq y$ and $n = x$ or $n = y$, a bee will only leave the patch where $n = x$ (or $n = y$ otherwise) after n consecutive failures, but on the other side a bee can visit at least 1 failure less before it will move to another colour. Since this probability isn't equal for both colours, no matching will occur.

- $p \neq q$

| n | x (patch 1) | y (patch 2) | p | q | MF₁ | MF₂ |
|----------|--------------------|--------------------|----------|----------|-----------------------|-----------------------|
| 3 | 3 | 2 | 0.4 | 0.8 | 0.0669 | 1.4666 |
| 5 | 3 | 2 | 0.4 | 0.8 | 0.2241 | 1.3880 |
| 10 | 3 | 2 | 0.4 | 0.8 | 0.4353 | 1.2824 |
| 20 | 3 | 2 | 0.4 | 0.8 | 0.5247 | 1.2376 |
| 200 | 3 | 2 | 0.4 | 0.8 | 0.5307 | 1.2347 |
| 2000 | 3 | 2 | 0.4 | 0.8 | 0.4892 | 1.7662 |
| 6 | 6 | 5 | 0.5 | 0.7 | 1.0874 | 0.9375 |
| 10 | 6 | 5 | 0.5 | 0.7 | 0.7884 | 1.1511 |
| 20 | 6 | 5 | 0.5 | 0.7 | 0.9974 | 1.0018 |
| 100 | 6 | 5 | 0.5 | 0.7 | 1.0150 | 0.9893 |
| 1000 | 6 | 5 | 0.5 | 0.7 | 1.1304 | 0.9069 |

Table 16

As we can see in the table above there cannot be a simple answer to the results. In some cases almost no matching is to be found ($x=3$ and $y=2$), but with the other instance in some cases (almost) matching can be seen. It looked like with well chosen values for all the variables matching occurs and the extended Failures strategy will perform as well as the basic strategy.

For now we don't have a fitting answer or explanation to this results, mostly because these simulations and results are obtained by trial and error, instead of computing the results in advance and simulating afterwards to check the computed results.

ϵ -Failures Strategy

We will simulate this strategy with the same setting as the basic ϵ -Sampling Strategy, because in that way we can measure it's performance correctly.

Basic single-bee setting [test for Matching Law]

For this simulation we used the following setup:

- 1 bee
- 1 flower of colour Y , full chance $y = 0.3$
- 1 flower of colour B , full chance $b = 0.7$
- 1.000.000 time steps
- $\epsilon = 0.05$

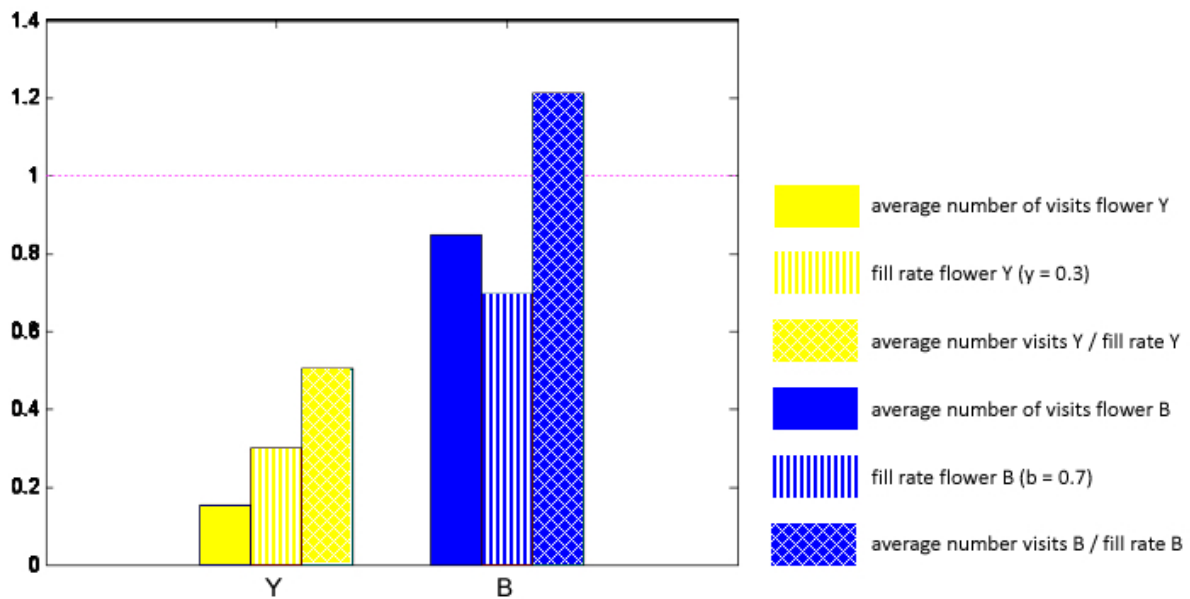


Figure 15

| Colour | Full chance | Average number of visits | Matching Factor |
|--------|-------------|--------------------------|-----------------|
| Y | 0.3 | 0.1516 | 0.5055 |
| B | 0.7 | 0.8484 | 1.2119 |

Table 9

As shown in the figure and table above, no matching is observed. We can explain this by something we noticed before. When simulating the dynamic ϵ -Sampling Strategy we noticed that when the full chances of the colours aren't equal the matching will be disturbed. Because a bee will encounter more failures on the colour with the lower full chance, it will change more quickly to the other colour. A bee visiting the most rewarding colour will act more likely as if it uses the basic ϵ -Sampling Strategy because the probability of visiting an empty flower is smaller.

By saying this we expect that this strategy will perform better when the full chances of the colours are equal, thus we will simulate this strategy again, but now with equal full chances for both of the colours.

For this simulation we used the following setup:

- 1 bee
- 1 flower of colour Y , full chance $y = 0.5$
- 1 flower of colour B , full chance $b = 0.5$
- 1.000.000 time steps
- $\varepsilon = 0.05$

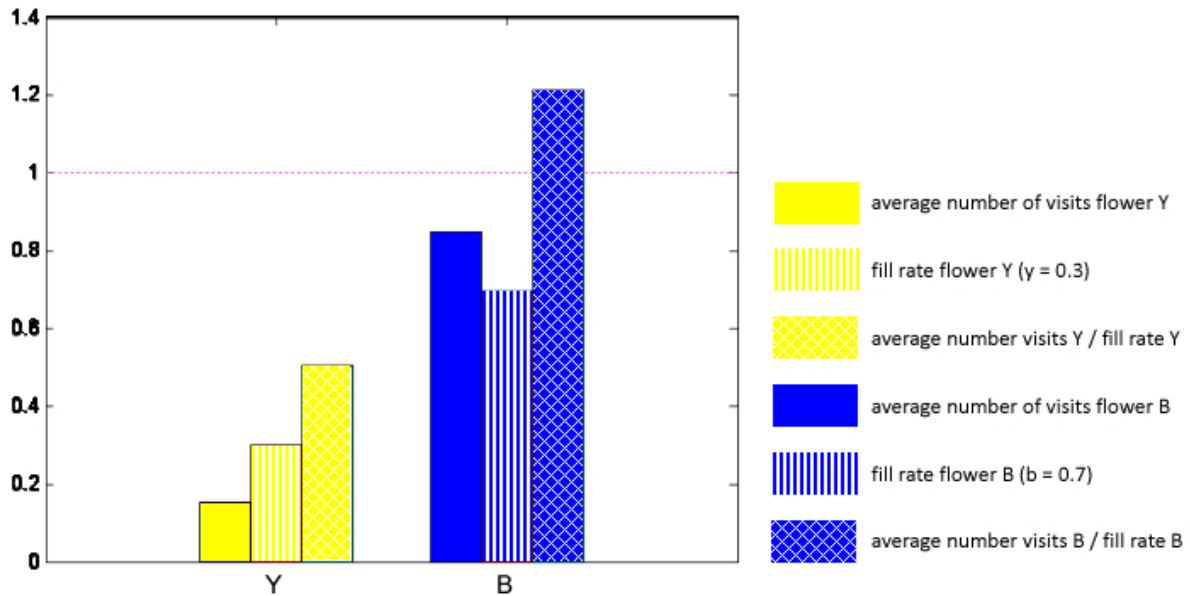


Figure 16

| Colour | Full chance | Average number of visits | Matching Factor |
|--------|-------------|--------------------------|-----------------|
| Y | 0.5 | 0.4995 | 0.9991 |
| B | 0.5 | 0.5005 | 1.0009 |

Table 108

This results show exactly what we expected. In case of equal full chances, matching will occur. This can be ascribed to the fact that the effect of the failures can be neglected. This happens with same probabilities per colour and won't have any influence on the average number of visits per colour.

Thinking of this, we don't have to simulate for testing if an Ideal Free Distribution will occur, this will be in line with the above. If the full chances are equal, at some stages an Ideal Free Distribution will exist. But in situations where the full chances (more exactly, the number of flowers per colour in case of the Ideal Free Distribution) will differ, an Ideal Free Distribution can occur, but not as often as with equal nectar supply per patch (we can say this because in the simulation every flower would produce 1 unit of nectar per step).

5. Conclusions and recommendations

Introduction

In the last chapter we give the conclusions of the research which has been done in order to write this thesis. These conclusions come from the development of the extensions of the basic strategies, the introduced ε -Failures strategy and the results, computed by the simulations. After that we will address some directions for further research.

Conclusions

With respect to the research questions and problem statement we may conclude the following:

Will the results in the basic paper also hold in more complex situations, i.e. more than 2 types of flowers?

As we observed in the results, the ε -Sampling Strategy will hold for situations with more than 2 types of flowers. When simulating, we had to choose the numbers of bees and flowers wisely to meet the assumptions made in the basic paper. If the total number of flowers is greater than the number of bees, an Ideal Free Distribution isn't likely to be seen, because it doesn't matter for the bees where to go, they will always receive a positive reward. When the number of bees is equal to the total number of flowers, an Ideal Free Distribution is observed, because the assumptions made in the paper are met. This can also be seen if the number of bees is greater than the number of flowers (scarcity of food), but the ratios of average number of bees per flower (per colour) would differ more.

In what way have the basic strategies be adjusted or extended to meet the more complex conditions?

After choosing and analyzing the basic strategies, a lot of extensions were taken into consideration. Some of them are part of this thesis, but most of them are not. The extensions had to be in line with the basics of these strategy. The strength of both of the strategies is their robustness and simplicity. We chose to make minor changes in some of the key features of a strategy. Furthermore we looked at the behaviour of the ε -Sampling Strategy in case of more colours. In the paper which served as the fundament for this thesis only situations with 2 colours were discussed. And since the paper only gave theoretical results for these strategies, we create a simulation to get a better view of the strategies.

For the ε -Sampling Strategy we introduced a dynamic ε -Sampling Strategy. This was considered to be more in line with the natural thought of leaving a patch faster if the rewards were bad. If a bee would observe more empty flowers in a patch of a certain colour, then the possibility of leaving that patch and move to another colour would increase.

For the Failures Strategy we introduced a 'Failure Time Window Strategy'. Instead of counting the consecutive failures and move to another colour if this number of failures reaches a certain number, this extension looks at the number of failures in the last couple of flower visits. By this manner, a bee will stay in a patch for a longer time and thus get the chance of gaining a better perception of the patch.

At last we introduced a 'new' strategy, the ' ε -Failures Strategy'. This strategy uses the basic principles of both ε -Sampling Strategy and Failures Strategy. A bee will forage a patch and with a small probability ε , the bee will sample a different patch. It will only stay at the other patch if there is a better (or higher) reward, otherwise it will return to the original patch. But if the bee visits a number of consecutive failures, it will also leave the patch and try to find a better patch.

How do the extended, new strategies perform in comparison to the basic strategies?

After implementing the strategies and simulating, the results show, that neither the extensions, like ‘failure window strategy’, nor the new dynamic ε -Sampling Strategy performed better than the basic strategies. In some cases (chosen wisely) the extensions and new strategy will perform almost as good as the basic strategies, but in no circumstances they lead to better results. This can be explained by the fact that the extensions of the strategies disturb the basic principles of the strategies and just these basic principles (especially a value for ε , which has to be close to 0) lead to optimal results.

Can the basic strategies be extended and will this improve their performance?

As we concluded, the basic strategies can be extended, in a lot of ways. We chose for 2 extensions (dynamic ε -Sampling Strategy and the Failure Time Window Strategy) and defining a new strategy (ε -Failures Strategy). These extensions and adjustments didn’t lead to better results or a better performance, in most cases performance dropped and the results declined. However, during the research a lot was learned about the strategies and it was very interesting to examine the extension of the number of colours. In the cases where the results of the extensions are (almost) as good as the results of the basic strategies, this happens because the influence of the extensions is reduced and the strategy will behave as the basic one.

When simulating, we noticed that the assumptions for finding an Ideal Free Distribution stated in the paper for the basic ε -Sampling Strategy weren’t met in the implementation of the strategy. But the results showed us that an Ideal Free Distribution could be noticed. By this we can say that the strategy is even more robust than is stated, because it will hold under less strict assumptions.

As some people might say: “Never change a winning team...”. However one can only try to improve it.

Recommendations for future research

If there was just more time... From the beginning of the research, one of the main tasks of my supervisor was keeping me focused and in the right direction. In one of our first meetings we discussed possible adjustments and extensions to the basic strategies. I optioned a couple of them and after one of those meetings I got the idea to create a tool where every strategy (and possible extensions) could be simulated, analyzed and compared. This was a little too much and I had to be hold back. Do one thing first and if that's done right, move forward. So I decided to focus on more colours, the dynamic ϵ and $A_n(x,x)$. I received a lot of answers to the questions I had starting this research and after finishing the research some questions still are open.

Some of the ideas and questions I had in mind for extending the basic strategies and aren't taken into consideration for this thesis were:

- Travel time: What happens if the travel time between flowers is taken into account? Will it be better to visit a flower which is nearer but has less nectar reward or to look for a richer flower more far away?
- Energy level: What happens if a bee has to visit the first flower it encounters, because otherwise it would starve?
- Addiction: What happens if bee develop an addiction for one particular colour? How does this affect the distribution of the population?
- Memory: By remembering more visits (and per colour) can the critical level be estimated in a better way? And in the case of more than 2 colours, when leaving a patch, which new colour has to be chosen? Is it better to visit a random patch or visit the patch with the highest estimated reward?
- Fly time: In the case of the basic strategies, a bee is always on a flower (unless no flower is free). But what if a bee decides to fly around instead of gaining nectar?
- Multiple strategies: How will a population of bees will be distributed if not all the bees use the same strategy?

Some of these questions and remarks have been examined in the past years, but it would be interesting to see how these robust and simple strategies would hold under these extensions.

The Matlab scripts took a long time to simulate, this was due to the fact of having not a supercomputer and a lack of great programming skills in Matlab. Maybe the simulations could be implemented in a different way to improve speed and performance.

Furthermore it would be nice to have a tool or program which can be used to simulate and compare the strategies at once. For now, we programmed and implemented the tool without a user interface. In fact for every simulation (strategy) a single script was created and adjustments of the variables would have to be done in the script. Therefore it would be nice to have all the different strategies and simulations together in one program. When this program could save all the performed simulations, they could be collected in a database and be available for further research. If the tool could be web-based, people all over the world (with an internet connection) could simulate the strategies and compare their results to the simulations from the database.

As stated in the first chapter, foraging strategies are used to solve congestion problems. The most well-known research is done by the School of Computer Science at the University of Dortmund (BeeJamA). It would be nice to see if the strategies stated in this thesis could also be used to solve this kind of problems.

6. References

- Stephens, D.W. & J.R. Krebs (1986). Foraging Theory. *Princeton Univ. Press*.
- Heinrich, B. (1976). The foraging specializations of individual bumblebees. *Ecol. Monogr.* **46**:105-128
- Cartar R., et al. (1990). Why are bumble bees risk-sensitive foragers? *Behavioural Ecology and Sociobiology.* **26**:121-127
- Kadmon, R., Shmida, A. and Selten, R. (1991). Within-plant foraging behaviour of bees and its relationship to nectar distribution in *anchusa strigosa*. *Israel Journal of Botany* **40**:283-294
- Greenleaf, S., et al (2007). Bee foraging ranges and their relationship to body size. *Oecologia* **153**:589-596
- Fretwell, S.D. and Lucas, H.L. (1972). On territorial behaviour and other factors influencing habitat distribution in birds. *Acta Biotheoretica* **19**:16-36.
- Hernstein, R.J. (1961). Relative and absolute strength of responses as a function of frequency of reinforcement. *Journal of the Experimental Analysis of Behaviour* **4**:267-272.
- Thuijsman, F., Peleg, B., Amita, M. and Shmida, A. (1995). Automata, Matching and Foraging Behaviour of Bees. *J. theor. Boil.* **175**:305-316.
- Ben-Porath, E & Peleg, B (1987). On the folk theorem and finite automata. Mimeo, Center for Research in Mathematical Economics and Game theory.
- Monderer, D. & Shapley, L.S. (1988). Potential Games. *Mimeo*.
- Kalai, E (1990). Bounded rationality and strategic complexity in repeated games. *Game Theory and Applications.* 131-157
- Schreiber, E. (1993). Foraging strategies in carpet inflorescences of *Alkanna strigosa* (Boraginaceae) by solitary bees. M.Sc. Thesis, Dept. of Ecology and Evolution, The Hebrew University of Jerusalem.
- Darwin, C. (1891). *The effects of cross and self fertilisation in the vegetable kingdom* (3rd ed.). John Murray, London.
- BeeJamA: <http://ls3-www.cs.uni-dortmund.de/en/projekte/bees/beejama/index.html>

7. Appendix

Matlab code for simulations

ϵ -Sampling strategy

Matching Test (basics)

```
% number of steps
nr_steps = 10000000;
samples = 0;

% epsilon
epsilon = 0.05;

% number of colours
nr_colours = 2;

% full chances
fillrate(1,1) = 0.3;
fillrate(1,2) = 0.7;

% set bee
bee = repmat(struct('colour', 0, 'epsilon', 0), 1, nr_steps);
r = randperm(nr_colours);
bee(1).colour = r(1,1);
bee(1).epsilon = epsilon;

% data to collect
on_colour(1,1) = 0;
on_colour(1,2) = 0;

% start simulation
t = 1;
for t = 1:nr_steps
    ep = rand;
    if(ep < bee(1).epsilon)
        % bee is sampling, so move to another colour
        samples = samples + 1;

        old_colour = bee(1).colour;
        co = 1;
        ct = 1;
        cos = zeros(1, nr_colours-1);
        for co = 1:nr_colours
            if(co ~= old_colour)
                cos(1,ct) = co;
                ct = ct + 1;
            end
        end
        cd = size(cos,2);
        cosr = randperm(cd);
        new_colour_id = cosr(1,1);
        new_colour = cos(1, new_colour_id);

        % check reward
        ep1 = rand;
        if(ep1 < fillrate(1, new_colour))
            % full, so stay in new patch
            bee(1).colour = new_colour;
        else
            % empty, return to previous colour
            bee(1).colour = old_colour;
        end
    else
        % no sampling, so stay in patch
        on_colour(1, bee(1).colour) = on_colour(1, bee(1).colour) + 1;
    end
end
```

Dynamic ε -Sampling strategy

Matching Test Dynamic Epsilon (basics)

```
% number of steps
nr_steps = 10000000;
samples = 0;

% epsilon
epsilon = 0.05;

% number of colours
nr_colours = 2;

% full chances
fillrate(1,1) = 0.3;
fillrate(1,2) = 0.7;

% set bee
bee = repmat(struct('colour', 0, 'epsilon', 0), 1, nr_steps);
r = randperm(nr_colours);
bee(1).colour = r(1,1);
bee(1).epsilon = epsilon;

% data to collect
on_colour(1,1) = 0;
on_colour(1,2) = 0;

% start simulation
t = 1;
for t = 1:nr_steps
    ep = rand;
    if(ep < bee(1).epsilon)
        % bee is sampling, so move to another colour
        samples = samples + 1;

        old_colour = bee(1).colour;
        co = 1;
        ct = 1;
        cos = zeros(1, nr_colours-1);
        for co = 1:nr_colours
            if(co ~= old_colour)
                cos(1,ct) = co;
                ct = ct + 1;
            end
        end
        cd = size(cos,2);
        cosr = randperm(cd);
        new_colour_id = cosr(1,1);
        new_colour = cos(1, new_colour_id);

        % check reward
        ep1 = rand;
        if(ep1 < fillrate(1,new_colour))
            % full, so stay in new patch
            bee(1).colour = new_colour;
        else
            % empty, return to previous colour
            bee(1).colour = old_colour;
        end
    else
        % no sampling, so stay in patch
        on_colour(1,bee(1).colour) = on_colour(1,bee(1).colour) + 1;

        % check reward and update epsilon
        ep2 = rand;
        if(ep2 < fillrate(1,bee(1).colour))
            % success
            bee(1).failures = 0;
        else
            % failure, so adjust number failures and epsilon
            bee(1).failures = bee(1).failures + 1;
            bee(1).epsilon = epsilon^(1/(bee(1).failures+1));
        end
    end
end
end
```

Failures strategy

```
% number of steps
nr_steps = 10000000;

% number of colours
nr_colours = 2;

% full chances
fillrate(1,1) = 0.2;
fillrate(1,2) = 0.2;

% change after number of consecutive per colour
change_after(1,1) = 2;
change_after(1,2) = 3;

% set bee
bee = repmat(struct('colour', 0, 'lastempty', 0), 1, nr_steps);
r = randperm(nr_colours);
bee(1).colour = r(1,1);
bee(1).lastempty = 0;

% data to collect
on_colour(1,1) = 0;
on_colour(1,2) = 0;

% start simulation
t = 1;
for t = 1:nr_steps
    on_colour(1,bee(1).colour) = on_colour(1,bee(1).colour) + 1;

    ep = rand;
    if(ep > fillrate(1,bee(1).colour))
        % failure
        bee(1).lastempty = bee(1).lastempty+1;
    end

    % Do we have to change colour or stay in the patch
    if(bee(1).lastempty == change_after(1,bee(1).colour))
        % change colour
        old_colour = bee(1).colour;
        co = 1;
        ct = 1;
        cos = zeros(1,nr_colours-1);
        for co = 1:nr_colours
            if(co ~= old_colour)
                cos(1,co) = co;
                ct = ct + 1;
            end
        end
        cd = size(cos,2);
        cosr = randperm(cd);
        new_colour_id = cosr(1,1);
        colour = cos(1, new_colour_id);
        bee(1).lastempty = 0;
    else
        % Stay in patch
        colour = bee(1).colour;
    end
    bee(1).colour = colour;
end
```

Failures (failures window)

```
% number of steps
nr_steps = 10000000;

% number of colours
nr_colours = 2;

% full chances
fillrate(1,1) = 0.2;
fillrate(1,2) = 0.2;

% change after number of consecutive per colour
change_after(1,1) = 2;
change_after(1,2) = 3;

% size of failures window
lastof = 2;

% set bee
bee = repmat(struct('colour', 0, 'emptyhistory', []), 1, nr_steps);
r = randperm(nr_colours);
bee(1).colour = r(1,1);
bee(1).emptyhistory = zeros(1,nr_steps);

% data to collect
on_colour(1,1) = 0;
on_colour(1,2) = 0;

% start simulation
t = 1;
for t = 1:nr_steps
    on_colour(1,bee(1).colour) = on_colour(1,bee(1).colour) + 1;
    ep = rand;
    if(ep > fillrate(1,bee(1).colour))
        % failure
        bee(1).emptyhistory(1,t) = 1;
    else
        % success
        bee(1).emptyhistory(1,t) = 0;
    end
    % Do we stay or move to another colour?
    if(t < lastof)
        colour = bee(1).colour;
    else
        B = bee(1).emptyhistory(end,(t - (lastof - 1)):end);
        lb = sum(B);
        if(lb >= change_after(1,bee(1).colour))
            % change of colour, too many failures
            old_colour = bee(1).colour;
            co = 1;
            ct = 1;
            cos = zeros(1,nr_colours-1);
            for co = 1:nr_colours
                if(co ~= old_colour)
                    cos(1,co) = co;
                    ct = ct + 1;
                end
            end
            cd = size(cos,2);
            cosr = randperm(cd);
            new_colour_id = cosr(1,1);
            colour = cos(1,new_colour_id);
            for ii=0:(lastof-1)
                bee(1).emptyhistory(1,t-ii) = 0;
            end
        else
            % Stay in patch
            colour = bee(1).colour;
        end
    end
    bee(1).colour = colour;
end
```

ϵ -Failures

```
% number of steps
nr_steps = 1000000;
samples = 0;

% epsilon
epsilon = 0.05;

% number of colours
nr_colours = 2;

% leave after this number of consecutive failures
nr_failures = 3;

% full chances
fillrate(1,1) = 0.3;
fillrate(1,2) = 0.7;

% set bee
bee = repmat(struct('colour', 0, 'epsilon', 0, 'failures', 0), 1, nr_steps);
r = randperm(nr_colours);
bee(1).colour = r(1,1);
bee(1).epsilon = epsilon;
bee(1).failures = 0;

% data to collect
on_colour(1,1) = 0;
on_colour(1,2) = 0;

% start simulation
t = 1;
for t = 1:nr_steps
    ep = rand;
    if(ep < bee(1).epsilon)
        % bee is sampling, so move to another colour
        samples = samples + 1;

        old_colour = bee(1).colour;
        co = 1;
        ct = 1;
        cos = zeros(1,nr_colours-1);
        for co = 1:nr_colours
            if(co ~= old_colour)
                cos(1,co) = co;
                ct = ct + 1;
            end
        end
        cd = size(cos,2);
        cosr = randperm(cd);
        new_colour_id = cosr(1,1);
        new_colour = cos(1, new_colour_id);

        % check reward
        ep1 = rand;
        if(ep1 < fillrate(1,new_colour))
            % full, so stay in new patch
            bee(1).colour = new_colour;
        else
            % empty, return to previous colour
            bee(1).colour = old_colour;
        end
    else
        % no sampling, so stay in patch
        on_colour(1,bee(1).colour) = on_colour(1,bee(1).colour) + 1;

        % check reward and update epsilon
        ep2 = rand;
        if(ep2 < fillrate(1,bee(1).colour))
            % success
            bee(1).failures = 0;
        else
            % failure, so adjust number failures
            bee(1).failures = bee(1).failures + 1;
            % check if we have to move to new colour
            if(bee(1).failures == nr_failures)
                % leave and choose new colour
            end
        end
    end
end
```



```
    old_colour = bee(1).colour;
    co = 1;
    ct = 1;
    cos = zeros(1,nr_colours-1);
    for co = 1:nr_colours
if(co ~= old_colour)
    cos(1,ct) = co;
    ct = ct + 1;
end
    end
    cd = size(cos,2);
    cosr = randperm(cd);
    new_colour_id = cosr(1,1);
    new_colour = cos(1, new_colour_id);
    bee(1).colour = new_colour;

    % set number of failures back to 0
    bee(1).failures = 0;

end
end
end
```