THE UNIVERSITY OF READING

Resource Allocation Analysis of the Stochastic Diffusion Search

PhD Thesis

Department of Cybernetics

Slawomir Nasuto

June 1999

ABSTRACT

The Stochastic Diffusion Search (SDS) was developed as a solution to the best-fit search problem. Thus, as a special case it is capable of solving the transform invariant pattern recognition problem. SDS is efficient and, although inherently probabilistic, produces very reliable solutions in widely ranging search conditions. However, to date a systematic formal investigation of its properties has not been carried out. This thesis addresses this problem.

The thesis reports results pertaining to the global convergence of SDS as well as characterising its time complexity. However, the main emphasis of the work, reports on the resource allocation aspect of the Stochastic Diffusion Search operations.

The thesis introduces a novel model of the algorithm, generalising an Ehrenfest Urn Model from statistical physics. This approach makes it possible to obtain a thorough characterisation of the response of the algorithm in terms of the parameters describing the search conditions in case of a unique best-fit pattern in the search space. This model is further generalised in order to account for different search conditions: two solutions in the search space and search for a unique solution in a noisy search space. Also an approximate solution in the case of two alternative solutions is proposed and compared with predictions of the extended Ehrenfest Urn model.

The analysis performed enabled a quantitative characterisation of the Stochastic Diffusion Search in terms of exploration and exploitation of the search space. It appeared that SDS is biased towards the latter mode of operation.

This novel perspective on the Stochastic Diffusion Search lead to an investigation of extensions of the standard SDS, which would strike a different balance between these two modes of search space processing. Thus, two novel algorithms were derived from the standard Stochastic Diffusion Search, ‘context-free’ and ‘context-sensitive’ SDS, and their properties were analysed with respect to resource allocation. It appeared that they shared some of the desired features of their predecessor but also possessed some properties not present in the classic SDS.

The theory developed in the thesis was illustrated throughout with carefully chosen

simulations of a best-fit search for a string pattern, a simple but representative domain, enabling careful control of search conditions.

Acknowledgements

I would like to express a deep gratitude to the Research Endowment Trust of the University of Reading for sponsoring a research project on Mathematical Analysis of Stochastic Diffusion Networks and Dr. J.M. Bishop for giving me the opportunity of working on this project. This thesis is a result of the research conducted on this project.

Further thanks go to the Department of Cybernetics, and especially to Dr. R. Mitchell and Dr J.M. Bishop for additional financial support in the last year of my studies.

Dr. J.M. Bishop was a supervisor of this thesis. Without his pioneering work on the Stochastic Diffusion Network this work would not be possible. I would like to thank him for encouraging independent approach towards scientific endeavour and lateral thinking. Our numerous discussions concerning the fundamentals of Artificial Intelligence and the philosophy of mind proved to be very stimulating and thought provoking. Mark’s friendly attitude and help went far beyond the call of supervisor’s duty and is greatly appreciated.

I would also like to thank several researchers who discussed with me various scientific issues and did not hesitate to spare their time to answer my numerous questions. The list (not exhaustive) includes Dr. Peter Cariani, Dr. Kerstin Dautenhahn, Prof. Karl Friston, Dr. Peter McOwan, Prof. Gregor Schoner, Dr. Steven Westland.

Dr. Kerstin Dautenhahn and Dr. Roger Bentley proof-read parts of this thesis. My warm thanks go to both of them. Their input contributed towards increasing the clarity of the exposition, remaining deficiencies being my sole responsibility.

Dr. Roger Bentley contributed greatly to the very friendly atmosphere in the department, being always cheerful and encouraging. I appreciate greatly our frequent nightly talks on various aspects of science and research in general and on the nature of PhD in particular.

I would like to thank also my fellow postgraduate colleagues and friends from the department, especially Matt and Martyn, who often offered me a helpful hand in solving several hardware and software problems, and to Stasha, for many hours spent on discussing scientific problems of common interest.

Several friends, now scattered throughout the world, influenced me and my way of thinking, thanks to all of you for being there.

I would like to express my deep gratitude to my parents Anna and Romuald Nasuto for constant support and encouragement. They always supported my interest in science, without them I would not be here.

Finally, but not lastly, I would like to thank my wife Marta and my daughter Gosia for coming to Reading during my studies - your presence here made easier many difficult moments. Marta also helped with taming Word to produce a desired output, showing more persistence than was left in myself. A big thank you for your support, patience, understanding and putting up with many inconveniences.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1. *Motivation for the work.*

One of the greatest problems facing philosophers and scientists is the nature of intelligence. The questions about human and animal intelligence acquired a new perspective with the invention of computing machines and modern computer science, and under the influence of the Turing-Church thesis they were quickly cast in computational terms. Thus, researchers started to build computational models, which they hoped would exhibit intelligent behaviour. The work related to modelling intelligent behaviour quickly split into two streams. Some models were directed at an explanation of human and animal behaviour and tried to follow closely experimental facts from psychology and neuroscience. They constituted a basis of the emerging disciplines of cognitive science, mathematical psychology, computational neuroscience.

Other models were more loosely related to detailed experimental evidence of specific behaviours. Instead, they were trying to explain general intelligence. The researchers pursuing this direction drew ideas from intuition, introspection, common sense knowledge, formal models, computational constraints etc., in the hope that general principles derived in this way could account for intelligent behaviour. The intuition behind this line of research was that, first of all, it should be possible to build intelligent systems without direct modelling of real life data and secondly that, once this goal was achieved, it would give an account of human intelligence as well. This line of thought was common in the Artificial Intelligence community. Initial work was firmly based on a belief that human reasoning can be explained completely in terms of formal systems.

In a construction of a typical deductive system the problem domain was first completely specified and a symbolic representation of constituent objects, quantities and relationships between them was constructed. This constituted a knowledge base of the system. An inference engine was a search method which specified how to traverse the knowledge base in order to perform efficiently inferences about the problem. Such systems could be analysed with the use of formal logic and their great strength followed from the fact that we could understand what they were doing, why they were undertaking particular inferential steps and, as a result, how they arrived at the proposed solutions. This transparency of the deductive systems was the basis of their reliability.

Unfortunately, the sources of their strength were at the same time contributing to their weaknesses. The understanding of this fact came from research in different domains.

Some of the criticisms originated from the work of K. Gödel, the relevant results are summarised for example in (Arbib, 1987), who showed that in any formal system that is consistent and sound there exist statements which are not provable within this system, i.e. the system has to be incomplete. This inherent limitation of formal systems was used as a main argument by Lucas in the 1960´s, recapitulated and extended in (Lucas, 1996), and by Penrose, (Penrose, 1990; Penrose, 1995), who advanced the argument that human intelligence cannot be based on computational principles. According to these authors, it would follow as a conclusion that it was not possible to realise a truly intelligent computer system. In other words, these authors claimed a fundamental gap between intelligence and program sophistication. They granted to the efforts of the Artificial Intelligence community at most the latter, and never the former.

Another criticism of computational approaches came initially from the work of philosophers, such as Wittgenstein and Heidegger, but soon became apparent to all researchers in the AI community. This problem was with the inability to cope with so called background or common sense knowledge.

A further problem faced by deductive systems was related to their inability to cope with insufficient or corrupted information.

These insights were at the basis of a paradigm shift in Artificial Intelligence from formal deductive systems to those based on a concept of pattern recognition. Pattern recognition based systems, augmented by a principle of limited resource computations, were believed to escape from the bottleneck caused by the use of formal logic because they could arrive at solutions that would not be supported by logical reasoning (Winograd, Flores, 1987).

In this paradigm fit classic systems based on a frame representation (Minsky, 1975) as well as work on analogical reasoning, e.g. (Hofstadter, 1998). The latter is based on the hypothesis that the strength of human intelligence lies in its ability to discover similarities in seemingly different, unrelated situations and solve problems by transfer of knowledge between different domains.

Thus, one can argue that one of the central tenets of the approach based on pattern recognition is invariant pattern matching, where one seeks a form of a systematic transform from one pattern, a problem whose solution is known, to another pattern representing the problem for which one seeks a solution. Finding such a relationship between the patterns representing these problems would enable the transfer of knowledge, subject to suitable modifications, necessary to solve the other problem.

It should be clear from the foregoing discussion that, in the context of the new approaches to the construction of intelligent systems, it is impossible to overestimate the importance of the type of representation one is using. In fact, the way the information is being processed is tightly coupled to the form of its representation. The standard approach of Artificial Intelligence was based on the Newell and Simon physical symbol system hypothesis, which proposed that intelligent behaviour could be achieved by manipulating *symbolic* representations, which in turn were built of atomic *symbolic* entities, (Newell, Simon, 1976). The typical interpretation of physical symbol systems holds that the symbolic representations are fixed, and set a priori by a programmer. This reflected an assumption, that cognition could be separated from the perception, which would proceed it and would therefore activate appropriate representations prior to any cognitive processing. Such an assumption was commonly used in cognitive science - e.g. Fodor and Pylyshyn argued that perception was independent of any cognitive or contextual influence, (Fodor, Pylyshyn, 1988). The use of fixed symbolic representations, following from this assumption, resulted in such desirable properties of the models as productivity and systematicity or structure sensitivity but also in brittleness and lack of ability to generalise. The related problem is also the lack of adequate learning mechanisms in the domain of the symbolic paradigm.

Another problem (related to the assumption about separability of cognition and perception) was termed blindness by Winograd and Flores, (Winograd, Flores, 1987). Blindness means effectively that systems based on these principles are in fact reflecting solely the intelligence of the programmer who encapsulated *his* understanding of the problem in an appropriate symbolic representation. In other words, the programs were a mere reflection of intentionality of human designers, and did not posses any capabilities of transcending the representations once given, and so they could not mimic creativity or understanding.

Some researchers working in Artificial Intelligence understood these limitations of physical symbol systems and tried to create systems that could avoid their pitfalls. For example, Hofstadter and his co-workers in their work on analogical reasoning departed from the classical fixed system of symbolic representations. In order to realise an analogical reasoning system, they approached the problem with an assumption that at the roots of creativity, and so of intelligence, is some form of nondeterminism, which enables exploration of different configurations of constituent representations, and the ability for their restructuring depending on context and top-down influences.

Connectionism offered an alternative, biologically inspired, approach to representation. The representations promoted under the paradigm of parallel distributed processing were based on a notion of multidimensional vector space, (Rumelhart, McLelland, 1986). They offered a much greater degree of flexibility and, most importantly, an inherent capability of coping with contextual information. Although there are very different mechanisms that fall under the scope of connectionism, nevertheless the common feature of many of them is that their operation can be summarised as some form of pattern recognition. Thus, in a sense they follow a conceptually similar direction as the newer artificial intelligence systems based on the pattern recognition paradigm (although the latter approaches the problem from a different perspective).

Yet another solution to the straightjacket of fixed representations was initiated by J. Holland in his work on classifier systems, (Holland, 1976), which gave rise to the now separate field of Genetic Algorithms. Here, the whole population of potential solutions, by imitating natural evolution, could change spontaneously and explore many alternative representations simultaneously.

Related to Genetic Algorithms are multi-agent systems, one of the main paradigms of Artificial Life, which offer both a great flexibility of possible representations and a potentially completely different perspective on the whole problem. Indeed, some authors, associated with AL research, have even more radical approach towards internal representation, e.g. Brooks posits the emergence of intelligent behaviour without internal representations altogether, (Brooks, 1987).

The more recent work of Winograd and Flores is in a similar spirit, (Winograd, Flores, 1987). Their approach fits into the framework of autopoietic systems of Maturana and Varela. The latter authors, (Maturana, Varela, 1980; Varela, 1997), coming from biological perspective, negate the existence of fixed representations and maintain that the key to understanding of both life and cognition is self-maintenance of living systems and their mutual interactions, which dynamically create consensual, or phenomenal, domains. Thus, in this view there are no fixed structures within the system corresponding to unchanging representations of the external world. If one can use the word ‘representation’ meaningfully at all, it should refer to dynamic constructs temporarily assigned to external events on the basis of interaction with the environment and especially with other similar systems.

The above `bird’s eye view’ presentation of different paradigms in Artificial Intelligence is by no means exhaustive, nor is it intended to reflect the ‘natural evolution’ of the associated concepts towards their ‘right’ form. This would be impossible because there is as yet no consensus on what representations and processing methods should be used as a basis of intelligent systems and most of the paradigms outlined co-exist and flourish with new work. The aim of the above presentation lies in the suggestion that, in spite of obvious and important differences, there is some degree of convergence of ideas in research coming from so many different perspectives. As pointed out in (Hofstadter, 1998) there would be hardly anybody in cognitive science nowadays that would opt for a strict separation or independence of cognition and perception, while researchers in connectionism invest also a lot of efforts into reproducing in their systems the desired features of symbolic systems.

It seems that a broad consensus is that the work should concentrate on obtaining systems in which emergence of representations of external events and the search for desired solutions to the problems would be coupled. Such systems should have very flexible structures allowing for highly adaptable, restructuring representations enabling the exploration of many alternatives.

The mode of processing would be, as stated above, linked with the creation of representations. It should allow for context sensitivity and should be more capable of exploring the emergent structure of representations than the general search methods used as inference engines of the early deductive systems. On the other hand, it should be more flexible then heuristic searches tailored to particular problems, or equivalently - their representations, used in later works. It seems that the emphasis should be put on distributed representations and nondeterminism as they allow for achieving at least some of the desired features.

One of the approaches that could fulfil such a broad characteristic is the Stochastic Diffusion Search (SDS). It was proposed as a part of a hybrid system, the Stochastic Diffusion Network (SDN), by Bishop, (Bishop, 1989), for solving an engineering problem in computational vision, namely location of eye features in facial images irrespective of their position. However, the nature of the problem solved by the system and a generic specification of SDS make it a more interesting generic technique, because it rests at an intersection of many of the ideas discussed above.

The following, brief description of SDS will make it possible to appreciate the position occupied by SDS among the paradigms discussed earlier in this chapter. It will also facilitate understanding of its relationship to several methods fitting broadly in these paradigms, which are discussed in more detail in Chapter 2. However, the detailed definition of SDS and a discussion of its most fundamental properties is postponed to Chapter 3.

In its most general formulation, SDS solves the best fit pattern matching problem, i.e. it tries to find in the search space a pattern that would fit to the predefined template, irrespective of its possible transformations. SDS consists of an ensemble of agents, each of which performs the search independently of others and, by evaluating random (micro-)features of the template at the chosen location, it poses hypotheses about the investigated potential solution. The probabilistic diffusion of information between the agents about positions of potential solutions establishes a weak, stochastic coupling, which nevertheless enables a rapid formation of dynamic clusters of agents representing the most promising solutions, with the biggest cluster corresponding to the best fit match.

From the brief description given above it should be clear that SDS possesses many of the features discussed in the proceeding paragraphs. The *subsymbolic* nature of the microfeatures used for the problem description places this technique somewhere between symbolic and connectionist representations. The operation based on loosely coupled, stochastic agents classifies the Stochastic Diffusion Search in both a family of random search methods, of which Genetic Algorithms are but one of the members, and a family of multi agent architectures, in which solutions of the problems are constructed collectively and agents enjoy a large degree of autonomy. SDS was successfully applied to the problems in shift invariant facial feature recognition (Bishop, 1989), motion tracking in video images (Grech-Cini, 1995) and for self-localisation in an autonomous navigation system, (Beattie, Bishop, 1998), proving empirically that it is an interesting and useful method.

The move away from fixed, rigid symbolic representations observed in some of the paradigms discussed earlier, has a price in the increase of complexity of the resulting systems. Classic symbolic systems had the advantage that, by construction, they could be formally analysed, their properties could be proven and their final results could be explained in terms of the system's operations.

This is much more difficult in systems which depart from this paradigm hence in all the systems discussed here. In spite of substantial efforts and partial successes, still the understanding of the operation of novel approaches comes more often from simulations than from a formal analysis. The problem lies in that these systems are considerably more difficult to analyse. This comes as no surprise, for example, in the field of Self-reproducing Automata, which is also interested in structural organisation of many simple components into a complex system, an original proof, created jointly by von Neumann and Burks, that it is possible to create a cellular automaton equivalent to a Turing machine and capable of self-reproduction was over 100 pages long, (Arbib, 1987).

A related problem is a lack of sufficiently widely accepted, formal frameworks which could enable their analysis and would allow for a formal characterisation of their similarities and differences.

As a result, there seems to be a tendency among the researchers exploring these new paradigms, to neglect the mathematical modelling of their systems, and rely instead solely on insights and simulations. These approaches are undoubtfully very useful in scientific endeavour in so complex domains, but nevertheless they should not replace the formal analysis but rather complement it. It seems then very important to equip new methods fitting into these paradigms with a mathematical analysis in order to characterise their fundamental properties.

This thesis presents an attempt at a detailed analysis of the Stochastic Diffusion Search. Although partial characterisations of the properties of SDS have been covered in (Bishop, 1989) and (Grech-Cini, 1995), here it obtains an in-depth, systematic treatment for the first time.

As stated above, the Stochastic Diffusion Search deserves a detailed analysis because of its links with so many areas of intensive research. It seems that it provides a flexible generic search method which, by sharing many features in common with the paradigms above outlined, may provide an optimal tool for an investigation of the role of dynamic representations in the intelligent information processing. It is by performing such analysis, augmented by suitable simulations, that we can get insights into how the system behaves.

Another rationale for this work is an investigation of a particular aspect of SDS's operation, namely the management of resource allocation, the latter being an interesting issue in its own right. As E. Horwitz expressed in (Selman, 1996) the resource allocation is an important issue in computationally intensive problems. This, convolved with the limited resources principle, whose role in surpassing the limitations of deductive systems has already been mentioned, prompts an investigation and analysis of methods which, like SDS, explore these principles. It is also hoped that, due to the generic formulation of the system, the model developed for its description, and methods of analysis presented in the thesis may be modified and transferred to other systems operating in a similar way to SDS, and so will help to better understand the underlying common principles behind the seemingly different approaches.

1.2. *An overview of the thesis.*

The second chapter of the thesis will present a more detailed overview of the existing systems having some conceptual overlap with the Stochastic Diffusion Search. In particular, the existing methods of invariant pattern matching will be discussed with techniques ranging from state of the art traditional computer science techniques to connectionist systems trying to imitate the information processing in the brain.

The algorithms performing random search or optimisation will be presented consecutively and compared with the way SDS operates and represents external objects. A similar presentation of the multi-agent architectures is given next. The purpose of Chapter 2 is to set the proper context for the Stochastic Diffusion Search, illustrate similarities and pinpoint differences between SDS and the algorithms discussed.

Chapter 3 will introduce a detailed definition of the Stochastic Diffusion Search, together with its Markov Chain model. It will also report on the fundamental properties of the system, which were proved on the basis of the latter. It will appear that SDS converges in a weak sense, i.e. its probability distribution converges to a steady state distribution. The ergodicity of the Markov Chain model enables the observed quasi-stability of SDS to be explained and justifies basing a definition of the halting criterion on a monitoring of the number of agents in the biggest cluster. This chapter will conclude with a presentation of time complexity of SDS, which can be proven to be sublinear in a special case and, as the simulations suggest, appears to retain this property in a general case.

Chapter 4 constitutes the basis for the resource allocation analysis of the Stochastic Diffusion Search. The new model, equivalent to the previous Markov Chain one, is a generalisation of a famous Ehrenfest Urn model of gas diffusion. Allowing for a direct modelling of the interactions between agents it enables the combinatorial explosion associated with the previous formulation to be escaped, and thus for a detailed analysis of the amount of resources - agents that SDS allocates to the best fit pattern. The discussion in this chapter is limited to the special case when only one object in the search space is similar to the template, but it allows for further elaboration presented in the following chapters.

Chapter 5 presents a generalisation of the Ehrenfest Urn model from the previous chapter to the case of two objects in the search space similar to the template. Due to the very definition of the problem of the best fit match the discussion of the allocation of resources may have two interpretations. First, it is possible to interpret this task in terms of searching for *the unique* solution. In this case, the analysis of resource management will be a starting point to a discussion of SDS's robustness to noise.

A second interpretation of the model comes forward, when one relaxes the assumption about uniqueness of the solution. In such a context Chapter 5 will provide an insight into how SDS processes different solutions.

The line of investigation associated with the interpretation of alternative solutions as mere disturbances will be further developed in Chapter 6, where the characterisation of a resource allocation in the case of a homogenous noise in the search space will be presented.

Even though more manageable than the direct Markov Chain model, the Ehrenfest Urn model results in solutions with very complex dependence on the parameters characterising the conditions of the search. It is therefore desirable to obtain a method of their approximate but simpler characterisation. Such an approximation is provided in Chapter 7, which also compares the quality of the approximate solutions across the parameter space of the search. It will appear that for wide regions of the parameter space the approximation is very good, and even in regions where it diverges from the exact solutions it still may very well describe the qualitative change of the resource allocation with the change of parameters.

The insights leading to the approximation method of Chapter 7 and the discussion of resource allocation in special cases discussed so far in the thesis poses further, more general questions which are taken up in Chapter 8. The work presented there shows alternative resource allocation schemes based on the original SDS. Although they share many similarities with the standard SDS, they differ in the underlying interpretation of the task to be solved, as discussed in Chapter 5. The so called context-free SDS is more similar to the standard SDS in the robustness to the noise and in very strong bias towards the best fit solution. On the other hand, the context-sensitive SDS will be more sensitive to noise but will allow for more even distribution of resources between competing solutions and thus it may better model interactions between several alternatives. The chapter will also present a characterisation of the resource allocation on the convergence time of all three systems.

The thesis concludes in Chapter 9 by summarising the work presented, and reflecting on its possible implications for future research.

CHAPTER 2

A CLASSIFICATION OF THE STOCHASTIC DIFFUSION SEARCH

2.1. *Introduction.*

The Stochastic Diffusion Search can be analysed and classified with respect to different criteria. One of the possible classification schemes is with respect to the task performed by the system. Thus SDS can be considered in the context of pattern matching and relating SDS to such algorithms will be a subject of the first part of this survey. Several techniques for the pattern matching in different data structures will be discussed. In particular we will present a Stochastic Diffusion Network, a hybrid system of which SDS is a search engine. Although the detailed definition of the Stochastic Diffusion Search will be deferred to the next chapter, the brief description provided in the exposition of SDN will highlight some important features and will enable us to see how SDS is related to other techniques presented throughout the whole chapter.

Pattern matching is closely linked with the specific domain by the use of data structures representing it (but see the possible generalisations of pattern matching algorithms discussed in the first part of the chapter). It is however possible to consider the operation of the system without referring to the particular problem domain. If analysed in this way, SDS appears to belong to a broad class of the random search and optimisation algorithms.

Random algorithms obtained an increasing interest during the last decades due to the inability of deterministic methods, based on a systematic search, or reasoning, of finding satisfactory solutions to many hard problems. It seems that there is an increasing awareness in the computer science community that, if one trades off the 100% accuracy for speed and flexibility, then random algorithms provide an interesting alternative. This concerns in particular random search methods. Many problems in computer science, artificial intelligence, technology, biological modelling etc. are directly reducible to finding optimal solutions according to some criteria. Random search algorithms are often faster and more flexible than deterministic algorithms. They are also more robust to local minima, which constitute a nontrivial problem for steepest descent algorithms. Many of these methods have been inspired by some natural, physical or biological, processes. However, the drawback of many such algorithms is a lack of formal analysis and thus of full understanding of their behaviour. In the next part of the chapter we will review the probabilistic search procedures and relate them to the Stochastic Diffusion Search.

Finally, the last section of the chapter will discuss SDS in the context of the architecture of the system. Probably the most general class of distinct architectures to which it belongs is the class of multi agent systems. Such systems can be contrasted with the traditional centralised approach of artificial intelligence to solving problems, in which there was a single stream of processing sequentially arriving at a solution. The recent rapid increase of interest in multi agent systems stems from the inability of coping, by classical artificial intelligence systems, with inherent brittleness, incomplete knowledge and background or context information. Multi agent systems are thought to be more flexible, robust and easier adaptable to the task modifications. They rely more heavily on the interactions between the agents and on an agent-human interaction and thus fit more naturally into the paradigm shift from algorithmic to interactive systems, (Wegner, 1997). Although the last decade witnessed an explosive increase of work in multi agent systems in both software and robotics, the beginnings can be traced to the 1950's and 1960's when the first work on modelling genetic evolution began, (Bremmerman, Anderson, 1991; Back, 1995), and when appeared the first cognitive models based on the concept of a demon, (Selfridge, 1958).

2.2. *Pattern matching - a survey.*

One of the most interesting and fundamental problems in pattern analysis is that of pattern matching. It is a problem of great importance and interest in its own right but also because of the numerous applications including text processing, image analysis, protein structure analysis and prediction, symbolic computation, automated theorem proving among others.

In general, in order to solve the problem of pattern matching, one needs to define representations corresponding to the scene as well as to the pattern. These representations should be defined over the set of all elementary features that characterise both scene and a pattern. The more complex features, resulting from relationships among the simple features, should be reflected in the structure of these representations. Thus, in this way the task of pattern matching is reduced to the one of searching and locating in a structure representing a scene a substructure corresponding to the pattern. Therefore two crucial components of the pattern matching task emerge: searching and recognising, the latter being determined by an imposed distance, or similarity measure (between the pattern representation and a corresponding substructure of the scene representation).

There have been numerous pattern matching algorithms developed in different contexts. The majority of them use some form of heuristics to enhance the search for the pattern in the particular domain but hence restrict applicability to other recognition tasks. Thus, in a quest for better methods, many researchers turned their attention to various pattern matching techniques observed in nature. However, systems developed so far are still far behind visual capabilities of simple animals. Therefore it seems that investigation of natural visual systems and incorporating their characteristic features into the artificial systems can be beneficial in terms of overall efficiency. Thus many techniques, especially connectionist networks, are inspired by architecture of a visual system.

However, the "reverse engineer" approach is not trivial. In spite of intensive research into human visual system, there are still more questions than answers. Other reasons for the difficulty of copying "mother Nature" follow from the fact that an attempt to its realisation implies facing all the fundamental problems of AI (due to interconnections between the visual system of a human, the rest of brain's machinery and higher order mental processes).

In spite of substantial interest in the last decades in constructing techniques capable of solving this task there are still several unresolved issues. One such problem is the difficulty of performing recognition of an object that has undergone various transforms in the search space. The transforms in general depend on the nature of the problem. They may be geometric, (e.g. translations, rotations or scale changes), or due to changes in the lighting conditions of an image or distortions due to noise. In the latter case one is posed with inexact matching problem, in which only part of the features characterising a given object may remain invariant.

In this section a general classification of different pattern matching techniques is presented together with a discussion of existing examples. Different classifications correspond to the nature of the technique or the specific problem that it solves.

2.2.1. *Exact pattern matching.*

All techniques that search for the exact copy of the pattern in the search space fall into this category. One can further divide this class with respect to the type of a search space, in which matching is being performed.

2.2.1.1. *String matching.*

Algorithms in this class solve the following problem:

> *Given a pattern p consisting of a single keyword and an input string s, answer 'yes' if* ***p*** *occurs as a substring of* ***s****, that is, if* ***s****=****xpy****, for some* ***x*** *and* ***y****; 'no' otherwise.*

The most well known algorithms discussed in this class will be explained below in some detail as they seem to be highly optimised solutions and allow extensions to other classes defined by the search spaces built of more general data structures.

Let us assume the convention **p**=$p_1...p_m$, **s**=$s_1...s_n$, m≤n.

The Karp-Rabin algorithm uses a hash function to lower the cost of comparing the pattern with each successive m-character substring, (Karp, Rabin, 1987). If **h** is a hash function mapping each m-character string to an integer and **h**($p_1...p_m$)≠**h**($s_k..s_{k+m-1}$) then it is certain that $p_1...p_m$ cannot match $s_k..s_{k+m-1}$. Otherwise one has to compare both strings character-wise to exclude a false match case. Karp and Rabin suggested for the hash function **h**(x)=x mod q, where q is an appropriately large prime number.

The m-character string $s_k..s_{k+m-1}$ can be transformed into the integer using the following formula

$$x_k = s_k b^{m-1} + s_{k+1} b^{m-2} + \dots + s_{k+m-1}$$

and treating each s_i as integer and choosing b to be an appropriate radix.

The Karp-Rabin algorithm finds an exact matching in time O(mn) in the worst case and in O(n+m) time in the expected case. It requires O(m) space.

The Knuth-Morris-Pratt algorithm pre-processes the pattern to construct an auxiliary table reflecting an internal structure of the pattern, which can help in speeding up the matching, (Knuth et al., 1977). It first reads the pattern and in O(m) time determines a table **h** that determines how many characters to slide the character to the right in a case of a mismatch during the pattern-matching process. Next, input string processing takes place. The underlying idea is that if one successfully matched the prefix $p_1...p_{i-1}$ of the keyword with the substring $s_{j-i+1}...s_{j-1}$ of the input string and $p_i \neq s_j$ then it would not be necessary to reprocess any of $s_{j-i+1}...s_{j-1}$ since this portion of the text string was the prefix of the keyword that we have just matched. Instead, pattern slides to the right a certain number of characters as determined by the table **h**. The table **h** has the property that h_i is the largest k<i such that $p_1...p_{k-1}$ is a suffix of $p_1...p_{i-1}$ (i.e. $p_1...p_{k-1} \neq p_{i-k+1}...p_{i-1}$) and $p_i \neq p_k$. If such a k does not exist then

h_i=0.

The Knuth-Morris-Pratt algorithm takes O(m+n) time and O(m) space to solve the exact matching problem.

The last algorithm discussed in this class was introduced by Boyer and Moore, (Boyer, Moore, 1977). This method compares characters in the keyword with these of an input string from right to left starting with p_m and s_m. If s_m does not occur in the keyword then there cannot be a match for the pattern beginning at any of the first m characters of the input. Therefore the keyword can be slid m characters to the right and one can try matching p_m and s_{2m}. This algorithm also pre-processes the keyword and calculates two auxiliary tables enhancing search.

Let us assume that one is about to compare p_m and s_k. Then it is possible to distinguish the following general cases:

- $p_m \neq s_k$; if the rightmost occurrence of s_k in the keyword is p_{m-g} then the keyword can be shifted g positions to the right to align p_{m-g} and s_k and then resume matching by comparing p_m with s_{k+g}.

In particular, if s_k does not occur in the keyword, the pattern is shifted m positions to the right and the algorithm resumes matching by comparing p_m with s_{k+m}.

- Suppose $p_{i+1}...p_m=s_{k-m+i+1}...s_k$.

If i=0 then there is a match. If, on the other hand, i>0 and $p_i \neq s_{k-m+i}$, then the two subcases need to be considered:

a) If the rightmost occurrence of the character s_{k-m+i} in the keyword is p_{i-g} then the algorithm shifts keyword g positions to the right to align p_{i-g} and s_{k-m+i} and resume matching by comparing p_m with s_{k+g}.

If it happens that p_{i-g} is to the right of p_i, i.e. g<0, then one should instead shift the pattern one position to the right and should resume matching by comparing p_m with s_{k+1}.

b) Suppose that suffix $p_{i+1}...p_m$ reoccurs as the substring $p_{i+1-g}...p_{m-g}$ in the keyword and $p_i \neq p_{i-g}$ (only the rightmost occurrence is considered). Then the algorithm aligns p_{i+1-}

$_g...p_{m-g}$ with $s_{k-m+i+1}...s_k$ and resumes search by comparing p_m with s_{k+g}.

The first table used by the algorithm is indexed by characters. It stores either the rightmost positions of the characters in the keyword or m, if the keyword does not contain a given character.

The second table is indexed by the positions in the keyword. For each of them it gives the minimum shift such that when p_m is aligned with s_{k+g}, the substring $p_{i+1-g}...p_{m-g}$ of the pattern agrees with the substring $s_{k-m+i+1}...s_k$ of the input string, assuming that p_i did not match s_{k-m+i}.

The Boyer-Moore algorithm solves the exact matching problem in O(m+n) time and O(m) space in the worst case. It is worth noting that the Boyer-Moore algorithm solves the problem in the expected time in O(n/m) and thus is sublinear on average.

2.2.1.2. *Two-dimensional pattern matching.*

The algorithms in this class solve the following problem:

> *Given the pattern representation as a two-dimensional array of symbols p_{ij}, $1\leq i, j \leq n$ and an input text representation as the second array t_{ij}, $1\leq i, j \leq m$, $n<m$ determine where (if at all) the pattern occurs as a subarray of the text.*

The first algorithm discussed here is the extension of the Knuth-Morris-Pratt (K-M-P) algorithm proposed by Bird, (Bird, 1977). The algorithm performs two nested loops indexed by row and column positions, j and k respectively, in the input text. Inside both loops it determines whether the pattern occurs in the text with its bottom right hand corner p_{nn} at the position t_{jk}. This operation is composed of row and column matching, respectively. Row matching is responsible for determining which row of the pattern, if any, matches a terminal substring of $t_{j1}...t_{jk}$. The idea is to consider each row of the pattern as a separate keyword and construct a pattern matching machine, matching simultaneously a number of patterns with a test string. In a column matching, the algorithm determines whether or not rows $\mathbf{p_1},...,\mathbf{p_{i-1}}$ occur immediately above row $\mathbf{p_i}$ in order to say whether or not the complete pattern is a subarray of the text. This is performed using again a Knuth-Morris -Pratt algorithm, working this time vertically.

The extended K-M-P algorithm has a running time O(n^2+m^2) and needs O(n^2+m) space.

The second algorithm is an extended version of the one-dimensional B-M algorithm, (Tarhio, 1996). The algorithm examines strips of r columns at a time, $^{m}/_{2} \leq r \leq m$. Instead of inspecting a single character at each step of the pattern, the algorithm examines a d-gram (a string of d characters). The text is split in $\lceil ^{(n-m)}/r \rceil$+1 strips of r columns, r≤m. Each strip is examined separately by applying a modification of B-M algorithm proposed by Horspool, as discussed in (van Leeuwen, 1990), to filter out potential matches. The latter are subsequently processed by a trivial brute force algorithm checking a position of **p** until a character mismatch is found or until a match of **p** is completed. Both, the filtration of the potential matches and the shifting of the pattern, are based on d-grams, i.e. a string of characters in a row, r+d≤m+1.

The extended B-M algorithm is $O(n^2/m^2 \lceil \log_c m^2 \rceil + cm^2)$ in expected time and in space $O(cm^2)$ in an alphabet of c characters.

In the worst case the extended B-M algorithm needs $O(m^2n^2)$ time but if one replaces the brute force algorithm by the extended K-M-P algorithm, then the algorithm becomes linear in the worst case whereas the expected time remains unchanged.

2.2.1.3. *Tree matching.*

Another interesting class of algorithms consists of tree matching algorithms. Trees are intensively used in symbolic AI because they enable encoding hierarchical dependencies and constitute a unified way of encoding knowledge as well as reasoning. Hoffman and O’Donnell present a survey of some algorithms and propose new ones, (Hoffman, O’Donnell, 1982). The one interesting property of tree matching algorithms described in their article is that they do not fit exactly into the framework of exact pattern matching algorithms. On the other hand they also do not belong completely to the approximate matching algorithms that will be described in the next subsection.

The definition of a tree is based on a finite ranked alphabet Σ of function symbols (constants are considered just as nullary functions). Function symbols define nodes of a tree and their ranks determine how many leaves grow from the node with a given symbol. The definition of a tree is recursive. First all nullary symbols are defined as trees and then any symbol $\mathbf{a}(\mathbf{t}_1,...,\mathbf{t}_q)$ (i.e. of rank q), whose arguments are trees is also a tree. Authors also define pattern trees in an analogous way to the definition of trees with the except that the alphabet Σ is now extended by a special nullary symbol *v* - a placeholder for any Σ-tree.

Thus a pattern tree **p** with k occurrences of the symbol *v* matches a subject tree **t** at node *n*, if there exist trees $\mathbf{t_1},...,\mathbf{t_k}$ such that the tree **p'**, obtained from **p** by substituting $\mathbf{t_i}$ for the i'th occurrence of *v* in **p**, is equal to the subtree of **t** rooted at *n*. Thus a matching problem can be defined as a finite set of pattern trees $\mathbf{p_1},...,\mathbf{p_m}$ and a subject tree **t**; its solution is a set of pairs (*n*,i), where *n* is a node in **t** and $\mathbf{p_i}$ matches at *n*. One can see that due to the placeholder symbol *v* the matching is exact with respect to the structure of the pattern tree, but does not need to be exact with respect to the information stored in nodes of this structure. Taking into account applications of these algorithms in which tree structures change dynamically (typically, matching is followed by replacement locally changing the subject tree) the authors designed algorithms able to respond incrementally to local changes in the subject in order to avoid repeated rescanning of the entire tree.

Due to their formulation Hoffman and O'Donnell treat pattern matching in trees as analogous to that of in strings and consequently all algorithms they propose are extensions of the K-M-P algorithm. They classify their algorithms in two main groups. The first class of algorithms are bottom-up pattern matchers. In these algorithms matching takes place by traversing the subject tree from leaves to the root. This method is a generalisation of the K-M-P algorithm. The algorithms in this class are characterised by a more expensive pre-processing but faster matching phase and a better response to local changes than algorithms from the second class. In the second, a top-down approach, the matching is performed by traversing the subject tree from the root to leaves. These methods, essentially, reduce a tree matching to a string matching problem. They have a better pre-processing time than bottom-up methods, but the matching times and update behaviour are inferior to that of the algorithms from the first class. Tree patterns are reduced to strings, which are matched along paths in the subject tree. The basic idea lies in the use of counters for co-ordinating the matches of different path strings. The counting however turns out to be also a limiting factor of the algorithm. For the most general case without any restriction with respect to the structure of the pattern tree, the bottom-up algorithm requires pre-processing of O(set^{rank+1} *sym patsize*) time and O(set^{rank} *sym*) time for matching. Here *patsize* is the sum of pattern sizes (each being the number of nodes in a corresponding pattern tree), *rank* is the highest rank in the alphabet, *sym* is the number of symbols in the alphabet and *set* is the number of different match sets generated.

The top down algorithm requires O(*patsize*) pre-processing time and O(*subsize suf*) matching time, where *suf* is a quantity depending on a structure of the pattern suffixes (at most equal to the maximum height of a pattern).

To summarise, all the algorithms in the exact pattern matching category are based on a systematic search of the search space - speed up is achieved through various ways of exploring the specific structure of the problem and of the data structures used for its representation. This may be contrasted with the approach to the solution of the problem by SDS, which is based on a random walk. It is also worth noting that the problem solved by SDS is much more difficult than the exact pattern matching. This is because in the latter case the constraint of searching only for an ideal solution imposes additional structure, which is used by the above algorithms during their operation, and which is absent in the best fit matching problem solved by SDS.

2.2.2. *Approximate pattern matching.*

As opposed to the algorithms searching for an exact copy of a pattern in a search space, algorithms in this class do not assume that such a copy exists. Instead, they search for its partial instantiation. Depending on the search space and an application there may be different definitions of such an instantiation.

2.2.2.1. *Approximate pattern matching in strings.*

A special subclass consists of algorithms searching for strings, (Galil, Giancarlo, 1988). In their review article Galil and Giancarlo survey both serial and parallel algorithms devised by themselves and other authors.

An example of a problem to solve is the following:

> *Given a text of length n and a string of length m, find substrings of the text that are at most k units apart from the pattern according to a given distance d and for a given integer $k \leq n$.*

The exact formulation depends thus on the definition of a distance measure. The two most popular measures used are Hamming and Levenshtein distances. Hamming distance is defined for strings of equal length as the number of positions with mismatching characters in given strings and approximate string matching in this case is referred to as matching with k differences.

Levenshtein, or edit, distance is defined for strings with arbitrary lengths as the minimal number of differences between the two strings (equivalently the minimal number of insertions and deletions necessary to transform one string into the other). In this case matching is referred to as a matching with k mismatches. The algorithms for approximate string matching fit into the following paradigm:

- pre-processing of the pattern and the text;

- analysis of the text.

Pre-processing consists either of gathering some information about the pattern and the text, which can be used for a fast implementation of primitive operations in the analysis step, or of constructing an automaton that accepts all strings at a distance at most k from the pattern.

An analysis consists either of scanning the text or of constructing an auxiliary table.

The algorithms using pre-processing use primitive operations that find the leftmost mismatch between any two given suffixes of the pattern and the text. They construct a suffix tree for either the string or the pattern or the string text$pattern (a concatenation of the text and the pattern).

Some of the algorithms analysing the text test each of its substrings t[i,i+m-1] trying to locate up to k mismatches between t[i,i+m-1] and the pattern. Therefore they constitute a natural extension of classic string matching algorithms. Other algorithms locate all occurrences with at most k differences of the pattern in the text through a table obtained by dynamic programming techniques. They constitute extension of the dynamic programming techniques for the computation of the edit distance between two strings.

The best serial algorithms for matching with differences as well as with mismatches have a time bound O(nk+(n+m)logm), but if k=Θ(m) they perform in time O(nm). The best parallel algorithm for k differences problem, using only a linear number of processors n, performs in time O(k+logm).

The most efficient algorithm for k mismatches problem is O(logm) in time with O(n*q*logm*loglogn) processors.

A sequential implementation of this algorithm would yield an O(n*q*logn*logm*loglogn) time algorithm and is better than the sequential algorithms when the size of the alphabet is small and k is very large.

2.2.2.2. *Feature location in images.*

Feature location in images constitutes a difficult visual problem for artificial systems. There has been a considerable effort focused on developing efficient methods for solving this problem. Location of a subimage in a bigger image is difficult because possible transformations of the feature can change representation of it in terms of intensity values and their spatial organisation. Therefore, in this type of a problem, one is faced with an approximate matching. However, in contrast with an approximate string matching, in this case one cannot specify in advance the maximal distance between the pattern and its representation. In fact the task is to find the best fit to the pattern, i.e. the partial instantiation which is closest to the original pattern.

2.2.2.2.1. *Classical vision systems.*

Several techniques have addressed the problem of inexact matching. Li, (Li, 1992), proposes approaching this problem by adopting Marr's, (Marr, 1982), point of view that an object-centred representation has to be derived for recognition. Li represented an object in terms of carefully chosen relations between its characteristic features. The feature relations he discusses are invariant to rotations, translations and scale changes, e.g. binary relations such as angles between lines, logarithm of the ratio of line lengths, ternary relations, such as angles in triangle defined by three points of the pattern etc. Next, he constructs a graph whose nodes are attributed with unary relations and constrained with higher order relations with the other nodes. The attributed relational structure obtained in this way is invariant to prespecified transforms and matching is formulated and solved as an optimisation problem by a relaxation labelling algorithm - a gradient descent based method. However, the use of gradient information may cause problems of convergence to local minima. In such a case the obtained solution is worse than optimal. In terms of image interpretation it may mean that the algorithm converged to some false pattern even though the original object was perfectly instantiated (e.g. subject to a rotation).

Shapiro and Haralick have proposed a similar approach in (Shapiro, Haralick, 1981), where they define a structural description of an object. The structural description of the object is very similar to attributed relational structure of Li with an exception that their primitives refer to object represented in the image and constitute its basic parts, whereas unary relations of Li refer to the visual representation of the object in the image.

To solve the problem of inexact matching Shapiro and Haralick propose and discuss the use of different tree searching algorithms like backtracking, backtracking with forward checking or backtracking and look-ahead. They conclude that for this problem the best method of all empirically checked was forward checking.

Vitanen (Vitanen et al, 1989) discusses a Chamfer matching (CMA) technique using a distance transform in order to generate a proper cost for matching a template. Distance transform is an iterative method for calculating an approximation of the true Euclidean distance from a given reference point. CMA is reported to be tolerant to noise and occlusion. Bergerfors extended it to using hierarchical resolution pyramids for stepwise execution: the matching process was first carried through in a lower resolution array with fewer starting points, positions and fewer feature points for calculation. The calculation is then continued for candidates in higher resolution until exact match was reached.

The method consists of:

- a scene analysis in which relevant features are detected;

- creating the distance image - each pixel is replaced by a value proportional to its distance from the nearest feature point;

- creating a model or a polygon - a co-ordinates list of representative points of the original pattern to be compared against the image;

- matching - transforming the polygon to different translated, rotated, scaled or tilted locations depending on the number of parameters in the recognition situation.

This means that matching is invariant to predefined transforms used for recalculating co-ordinates of the polygon.

Authors report that their method is of order n^x where n is a number of pixels in each dimension and x - the number of parameters.

To summarise, there is some similarity between the structural description methods of Li, (Li, 1992), and Shapiro and Haralic, (Shapiro, Haralick, 1981), and the Stochastic Diffusion

Search. The latter uses some form of a structural description in order to locate consecutive (micro)-features by an agent upon a successful activation on a matched feature, although features used in SDS are at sub-symbolic level. SDS, like Shapiro and Haralic's method, is also based on a search procedure, however at its hart is a global random search instead of a heuristic search used by these authors.

2.2.2.2.2. *Connectionist techniques.*

Pattern recognition has also been investigated in the neural network community in the context of visual processing models. One of the fundamental problems of early connectionist models was stimulus invariance. There has been no clear normalisation mechanism that would constitute an integral part of a model and would enable performing recognition task under such transformations.

Connex: eye feature location project.

In this part of the section five approaches to feature location will be described. In all cases the research has been performed on eye feature location inspired by the Connex programme. The same data images have been supplied by BT allowing easier comparison of obtained results. Below, the comparative summary of these approaches will be given underlying existing similarities in tackling the problem.

The general problem of locating an eye feature in the face images has been addressed in (Waite, 1992; Debenham, Garth, 1992; Hand et al, 1992; Lisboa, 1992). In (Vincent et al., 1992) a modification of this problem, a location and tracking of both eyes and mouth features in sequences of images, has been considered. Debenham has chosen for eye location a Radial Basis Function Network (RBFN) trained with Enhanced Restricted Coulomb Energy (ERCE), (Debenham, Garth, 1992). The rest of researchers have used a Multi Layer Perceptrons (MLP) with one hidden layer. A steepest descent algorithm for training was used and networks were trained to give a high response to inputs representing a desired feature and a low response otherwise. Vincent used in the hidden layer, second order neurons in order to facilitate complex decision boundaries (second order neurons have as a total input weighted sum of all incoming inputs as well as their squares, (Vincent et al, 1992). This allows them to form a closed decision surface - a hyperellipsoid, independently of dimensionality of the input data. One achieves equivalent behaviour with standard neurons, if the number of neurons in the hidden layer exceeds the dimensionality of the input data by one). Waite used a standard MLP but trained the network on the response function exp(-ax^2) instead of the discrete set {0,1} (An MLP can learn better smooth functions than discontinuous ones), (Waite, 1992).

The generally accepted strategy was to train the network by scanning images with a window of a preselected size. However, in four out of five cases researchers stated that the training was slow in the case of raw data training [Lisboa, (Lisboa, 1992), did not address the issue of training time at all]. Pre-processing of the input data was proposed as a general solution to this problem. Different methods of pre-processing were proposed: a multiresolution pyramid, (Vincent et al, 1992; Debenham, Garth, 1992; Hand et al, 1992), filtering, (Debenham, Garth, 1992), a principal component compression, (Waite, 1992), and a Gabor function representation, (Lisboa, 1992).

A multiresolution pyramid consists of different image representations, each with a different resolution. In the base level of the pyramid is the raw image and each next level contains a lower resolution representation of it. All researchers using this approach defined the resolution transformation between different levels of the pyramid to be an averaging 2x2 arrays of pixel's intensities. Hand used 3 levels of resolution, other researchers confined the pyramid to two levels only, (Hand et al, 1992).

The network initially is trained on each resolution. Next, a trained network is convolved over the image and pixels obtaining high response are chosen to locate most promising areas in the next, finer level of the pyramid. Thus, effectively in each of the levels, except the first one, the search space is confined to, usually, a small region. This results in a significant reduction of a processing time reported by all researchers using this method. An interesting use of the multiresolution pyramid, except that of reducing the search space size, is reported in (Hand et al, 1992). Here, the authors achieve speed up in training time by projecting weights learned in a coarser resolution level to the finer one. Thus, in this way the weights obtained in the coarser level are used for initialisation of the network trained on the next resolution level. This is intuitively explained by observing that the projected set of weights is a subspace approximation to the original solution, thus should be a good initial guess for the location of the global minimum.

Another benefit reported in (Hand et al, 1992) is a reduction in a number of false positives at the finest level of resolution, leading to an improved accuracy in locating the eye.

In fact, as Wechsler points out in (Wechsler, 1990), a multiresolution pyramid with such averaging constitutes another way of filtering (a low-pass filter). The resulting pyramid stands for a multichannel system where finer details tend to disappear in higher levels of the structure (a lower resolution). This is in accordance with the intuition that different features can be best detected when observed in optimal resolutions. This approach is interesting also because it seems that the human visual system performs a kind of multichannel frequency analysis, (Wilson Bergen, 1979; De Valois, 1979). However, authors do not dwell on the biological plausibility of this method.

Debenham additionally filters input data (a raw image) by ‘an unsharp filter’, (Debenham, Garth, 1992).

In order to speed up the training process Waite reduced the dimensionality of the input data by an orthogonal projection on the lower dimensional hyperplane spanned by the first m eigenvectors (corresponding to the m most significant eigenvalues) of the scanned window covariance matrix, (Waite, 1992).

Inspired by the research in the functional classification of neural cells, (Hubel Wiesel, 1962; Jones Palmer, 1987a, 1987b, 1987c), which suggests that some of neurons may act as edge detectors, and their response could be modelled by one- and two-dimensional Gabor functions, (Daugman, 1985; Marcelja, 1980; Pollen, Ronner, 1983), Lisboa used Gabor functions for an image representation, (Lisboa, 1992). Effectively, he trained a multilayer perceptron with coefficients of suitable Gabor expansions.

Finally (Vincent et al, 1992) and (Debenham, Garth, 1992) report using heuristics to improve further their techniques.

Vincent used his architecture for solving a problem of locating and tracking of features in image sequences. He makes his task easier by simultaneous location of both eyes and a mouth. Such a pattern is much better defined than a single eye feature. He uses moreover ‘a spatial and temporal pruning’. These are just heuristics respectively rejecting, in the same frame, implausible locations of eyes and mouth with respect to each other and confining the search space in the next frame to the regions of correctly located features in the previous one respectively.

Debenham and Garth, who observed that eyes are usually represented as regions darker than the surrounding area and consequently confined the search space to regions with the change in the intensity of the pixels, used similar heuristic.

Summarising these approaches all authors used rather standard neural structures in their experiments. In all cases there was a need for additional pre-processing of images to increase the speed and accuracy of the networks. They proposed different filtering methods (all commonly used in the past) enabling dimensionality reduction of the search space. The most popular filter was a multiresolution pyramid. Due to the use of this technique some obtained a reduction of a processing time, higher reliability of systems and an effect similar to that of focusing of an attention. However, in spite of biological inspirations of one of the authors and the biological plausibility of multiresolution pyramid as a model of a human visual system information processing, the resulting structures do not offer satisfactory solutions to the stimulus equivalence problem. Pre-processing does not take place in the connectionist part of their systems, thus both normalisation and focus of attention are obtained in a biologically implausible way. Therefore from a connectionist science point of view the proposed architectures stand conceptually on the level of early connectionist models and do not provide significant improvements with respect to the invariance problem. Partial responsibility for this fact bears the aim of the project, building an engineering system solving a particular task, which was not directed towards a biological plausibility of obtained systems.

Also worth noting is the fact that, although the task was to locate the desired feature in the image, no researcher did propose an interesting searching technique to speed up locating the feature. Instead the technique used was to scan a predefined window consecutively over the image or some of its regions. Although, several of researchers discussed the issue of a processing time, however there was no single work trying to characterise theoretically the time complexity of proposed method.

In contrast to the architectures discussed above, SDS uses a global search method to locate the best fit pattern, and thus avoids the problems facing the gradient based feedforward networks. The use of a systematic scanning of an image by the prespecified window adds a sequential component to the processing time of these nets. SDS, on the other hand is highly parallel and thus is not constrained by a exhaustive scanning method. The standard SDS is not based on a multi resolution pyramid, however an extension of it in this direction has been recently proposed in (Beattie, Bishop, 1998) in a successful attempt to use this technique for the localisation part of a navigation system.

We can note also that the methods discussed in this subsection are not supported by any formal analysis.

Hinton mapping.

One of solutions proposed to overcome the problem of stimulus invariance was Hinton mapping, (Hinton, 1981). Hinton proposed in his paper a PDP model capable of recognising objects regardless of their transformations by imposing a canonical frame of reference on objects. He obtains a viewpoint-independent description of the shape of an object by describing the spatial dispositions of the parts relative to this object-based frame. He proposed a parallel architecture capable of simultaneous discovering a representation of the shape and a representation of the object-based frame of reference. The model consists of object-based, canonical feature detectors and retinocentric detectors. Choosing an object-based frame of reference is equivalent to choosing a mapping from retinocentric features activated by the input pattern to object-based ones. It is assumed that retinocentric feature detectors are rich enough to be able to fit to an object regardless of its possible transformations. A mapping from retina-based features to object-based features is performed via mutual interaction between the two classes of detectors and a fixed set of so called mapping units, representing all possible transformations. This interaction is achieved by multiplicative connections from retinocentric and mapping units to object-based ones and feedback multiplicative connections from object-based and retinocentric units to mapping ones.

Initially all mapping cells are partially active. The activity pattern from retinocentric detectors is then sent via all possible mapping units to object-based units, which will become partially active. Among all these object-based features there will be some subsets corresponding to partial descriptions of familiar objects. Positive feedback from higher order units will reinforce appropriate object based features and these, together with corresponding retinocentric features, will send activity to mapping units. In this way the correct mapping will be discriminated, as ‘correct’ pairings will produce the greatest total input to the corresponding unit. The more active the mapping unit becomes, the more it contributes to the corresponding shape description, allowing for fast convergence of the system to the particular mapping, and the particular shape, description.

Hinton’s approach, considered as a metaphor of human vision processing, might have a partial support in the physiological studies performed by Hubel and Wiesel, (Hubel, Wiesel, 1962), suggesting that simple features such as bars with a particular orientation are recognised by single neurons. They found that complex patterns were activating more specialised neurons, which responded only to particular combinations of inputs from many of the neurons recognising simple features. This opinion was popularised further by a review article by Ballard, (Ballard et al, 1983), and still seems to have advocates amongst computer scientist, (Perrot, Hamey, 1991). However, Wechsler in (Wechsler, 1990) states that such an interpretation of single cells as edge detectors is unfounded. Wechsler claims that, in the light of a well established multichannel spatial frequency analysis in the work of Wilson and Bergen, (Wilson, Bergen, 1979), the results of Hubel and Wiesel can be easily accounted for. To support his claim, Wechsler quotes other results, e.g. (de Valois, 1979), reporting quantitatively the degree, to which a simple cell's response to bars could be predicted from its response to sine wave gratings.

Hinton suggests also extending his architecture towards such a hierarchical and distributed encoding of features and mappings as patterns of activity in many different units, in order to reduce the number of retinocentric and mapping units used by his system. However, he does not discuss the details of such an extension.

Hinton concentrates on the viewpoint independent recognition, without taking into account other possible transforms (nonrigid transforms, noise, etc.). Another drawback stated in (Rumelhart et al., 1986) is that Hinton's mapping allows only for serial attention modelling, whereas a system allowing a parallel focus of attention could explore simultaneous, mutual constraints between patterns that could model contextual dependencies.

In contrast to the Hinton's mapping, agents in SDS may, in principle, analyse whole classes of transforms. Due to a distributed processing of the search space and probabilistic character of agent's operation SDS can analyse many objects in parallel. For the same reasons local minima do not pose a problem, again in contrast to relaxation based methods.

Dynamic Link Matching.

In (Wiskott, von der Malsburg, 1996) the authors propose a system for recognition of objects from natural images. They concentrate on face recognition and claim that the system is inherently invariant with respect to translations and is robust with respect to rotations in depth and deformations. The architecture is based on the idea of dynamic links proposed by von der Malsburg in (von der Malsburg, 1981) as a hypothetical mode of information processing in the brain. The classical neural networks rest on the following assumptions, (von der Masburg, 1995):

- a short-term memory is encoded in the mean firing rate of neurons. The latter encode the basic symbols;

- the short-term memory arises from excitatory and inhibitory interactions between neurons;

- a long-term memory is coded in the (synaptic) weights of the connections linking neurons;

- the long-term memory is achieved through the mechanisms of synaptic plasticity.

The classical neural nets have been criticised for their problems with representation and processing of higher level symbols, (Fodor, Pylyshyn, 1988). Von der Malsburg, (von der Malsburg, 1981), identified this problem with the so called binding problem, posing the question of how distinct features of objects are bound together in order to form their internal representations. The dynamic link theory, addressing the binding problem, posits that coherent internal representations are created due to dynamic links capable of dynamically forming and destroying neural assemblies built of neurons corresponding to distinct features of the same object. The assemblies are defined by a synchronous firing of the constituent neurons. The characteristic feature of the theory is that, in effect, it introduces an additional degree of freedom into the information processing. Classical neural networks possess one degree of freedom corresponding to the activity of neurons encoding simple features. In the dynamic link theory an additional degree of freedom, the synchronous activity of assemblies, allows for defining higher order symbols out of simpler ones, a feature effectively used in a connectionist logical system proposed in, (Shastri, Ajjanagadde, 1993; Shastri 1995). The theory obtained an empirical support in recent neurobiological work of a number of researchers investigating synchronous oscillations in the cortex, e.g. (Engel et al, 1992; Abeles at al., 1994; Konig et al., 1996; Eckhorn, 1994).

The system discussed in (Wiskott, von der Malsburg, 1996) consists of two neural sheets of feature detectors modelled as Gabor wavelets and a memory containing templates of frontal views of faces. The system relies on self-organisation achieved due to fast synaptic plasticity and the topography preserving Mexican hat lateral connectivity within neural sheets. The architecture uses the winner-takes-all mechanism in order to sequentially rule out the least similar templates, leaving eventually the best fitting one to the input model.

As the authors admit the system presented in this work is very computationally intensive and, what is probably even more important, very slow. It requires a considerable amount of sequential steps to arrive to the solution, a feature that is in clear contradiction with the neurobiological data from the visual domain, (Thorpe, Imbert, 1989).

The Stochastic Diffusion Search, by contrast, stresses a highly parallel, distributed processing and is very fast. The similar feature in both models is the fact that, effectively, SDS operates on two types of information - a feature and its location, and thus implements a form of a bivariate processing which is also postulated in the dynamic link theory.

Sequence Seeking and Counter Streams Model of the Cortex.

In (Ullman, 1995), Ullman proposes an interesting model of the information flow in the cortex. The system encompasses considerations from a computational vision and a perceptual psychology. The author argues that the architecture of the cortex is highly suitable to perform the type of information processing envisaged by him and discusses his model in the context of the cortical connectivity.

One of the key ideas incorporated in the model is the approach to object recognition based on alignment, (Ullman, 1996). The objects in the external world rarely appear in exactly the same form as stored by the system’s internal representation. Usually, they are subject to various transforms, thus a need for invariant recognition. The basic idea behind the alignment approach to invariant object recognition is to compensate for the transform that makes the object and its internal representation differ. In other words, it is equivalent to finding an inverse transform, applying it to the object and matching the resultant image against the internal memory models. Thus, recognition is seen here as a search for an allowable sequence of transformations and the model that jointly minimise discrepancy between the object and the model. Ullman argues that such a (transform) sequence seeking is most suitable for a highly parallel architecture capable of simultaneous investigation of several competing alternatives. In his model the sequence seeking is a bi-directional process performed by the intertwined bottom-up and top-down processing performed by, respectively, ascending and descending pathways. The pathways are built of interconnected nodes, which consist of populations of many neurons. Thus the nodes may represent whole patterns of activity, corresponding in turn to hypothetical transformations. The input image and the memory models give rise to many sequences of possible transformations, which are processed simultaneously along separate and complementary paths. Any one stream of excitation in one pathway primes a trace in the complementary stream making it more excitable. Thus multiple streams originate from the input image and memory models in parallel. The complete sequence from the model to the input image is established once any ascending and descending tracks arrive at the corresponding nodes in the network. The arrival need not be simultaneous, because a decayed sequence leaves behind, for some time, a primed sequence in the complementary path. The direct links from the low level to high level nodes, omitting intermediate stages, could perform initial selection of memory models to be activated, thus reducing the number of initial hypotheses that would be considered by the system. Ullman posits that his system can take into account contextual effects by priming of some of the nodes on a longer, than the between-streams priming, time scales. The author claims further that, by enhancing all the streams passing through the primed node, the context would have broad effects on the processing, not dissimilar to the human capability of using contextual information. The local reinforcement rule would enable learning of the complete successful sequences. Ullman proposes that the rule, although similar to Hebbian learning, should differ from it in greater efficacy of synaptic modification of primed nodes as compared with corresponding changes in non-primed ones.

The sequence seeking architecture is very interesting and shares some key properties with the processing of information by the Stochastic Diffusion Search. However, although in both sources, (Ullman, 1995; Ullman, 1996), the author explains in detail the conceptual basis of the model and its neurobiological grounding, nevertheless there are many details that are omitted. It is not clear in what way the nodes can generate the transforms in the sequence-seeking stream. As the author notes the neural populations represented by nodes may overlap. This would suggest that the architecture does not have a modular character, i.e. the transforms are not a priori attributed to particular nodes. On the other hand, assuming standard model neurons as constituents of the whole architecture, it is not clear how one could distinguish between separate nodes on the basis of their functionality. This leaves the open question how the transforms are actually being generated. Unfortunately, the author does not give any details of the subpopulations of networks to which nodes do correspond.

The alignment approach to invariant recognition is very similar to the one taken in SDS, although they are realised somewhat differently. In particular, the recovery of the inverse transform in SDS can be achieved on the basis of a single (micro-) feature of an image, whereas the architecture proposed by Ullman suggests taking into the account a considerable amount of information for the same purpose. Also the emphasis on exploring the advantages of a highly distributed parallel processing of many simultaneous alternatives is similar in both systems. However, although Ullman stresses that one of the main considerations for the proposed architecture was the speed of processing, the learning rule based on Hebbian learning still needs many iterative steps for effective operation. It is not possible to make any comparison with SDS with this respect as the latter, at present, implements only an activation dynamics and thus do not contain any form of learning. Although the original SDS was designed with translation invariance as one of primary objectives it can be straightforwardly extended to account for a whole set of allowable transformations. Thus the functionality of both systems does not differ in any significant degree with respect to invariant processing. Ulmann mentions that on the basis of performed simulations the sequence seeking architecture compares favourably to simulated annealing or gradient descent, however he does not provide any formal analysis of the system, mentioning only that it is fairly robust with respect to local minima (suboptimal transform sequences), (Ullman, 1996).

The Stochastic Diffusion Network - a hybrid method.

The Stochastic Diffusion Network, (Bishop, Torr, 1992), has been proposed as a connectionist pattern recognition technique able to classify patterns invariant of transformation within a search space.

The Stochastic Diffusion Network (SDN) consists of the Stochastic Diffusion Search (SDS) engine, directing n-tuple RAMs, (Alexander, Stonham, 1979), into a search space. Search is performed by a competition-co-operation process between simple agents passing potential positions in the search space to n-tuples and diffusing information among other agents in order to explore the best intermediate solutions. It is this competition-co-operation process that enables a parallel focus of attention of SDS. SDS will be discussed thoroughly in the consecutive chapters.

SDN consists of an array of Minchinton cells (simple comparators transforming grey level data into binary data), n-tuple cells and SDS agents. Minchinton cells obtain grey level input data from positions pointed to in the images by SDS agents. Then they transform grey level data into binary data. Standard N-tuple cells (RAM), which receive binary data from Minchinton cells during learning, store the Boolean value in the address defined by the feature of the image defining the input data. However, in order to avoid saturation of the network and maintain a fast convergence of the Stochastic Diffusion Search, only features with the highest frequency of appearance as input data for a given RAM would be stored in that n-tuple at the end of a learning session. During testing, SDN operated in a similar fashion except that n-tuples acted as logic gates evaluating whether a presented micro-feature was similar to the one stored in the memory, thus they were responsible for the evaluation of positions pointed to by agents. In experiments reported in (Bishop & Torr, 1992) SDN was first presented with examples of eye features during training phase and then tested on unseen faces. The accuracy obtained was above 60% and authors report high time efficiency achieved in their experiments.

The Stochastic Diffusion Network can perform both serial as well as parallel focusing of attention and is robust to noise distortions and partial occlusions of the object. Transformation invariance is maintained mainly due to the Stochastic Diffusion Search capabilities of discovering partial matches.

2.3. *Random search and optimisation algorithms.*

Another way of looking at SDS is to analyse what it is doing in more abstract, domain independent terms. This leads us to the conclusion that SDS can be placed in the family of directed random search algorithms defined by Davies in (Davis, 1990) to include Simulated Annealing, (Kirkpatric, Gelatt, Vecchi, 1983), and Genetic Algorithms, (Holland, 1975). This class of algorithms includes also other probabilistic optimisation methods such as Extremal Optimisation, (Boettcher, Percus, 1999), Evolutionary Strategies, (Back, 1996), Evolutionary Programming, (Back, 1996), or Chemotaxis, (Bremmerman, Anderson, 1991).

2.3.1. *Mechanisms based on physical processes.*

Simulated Annealing is a general-purpose global optimisation technique for very large combinatorial problems, (Kirkpatric, Gelatt, Vecchi, 1983). Most of the development to date has been in the area of a large-scale combinatorial optimisation (problems with discrete variable parameters) but it has also been applied to optimisation over continuous variables. Simulated Annealing is based on the concept of physical annealing - the gradual cooling of a molten solid to produce a solid of minimum energy. In Simulated Annealing the set of parameters defining the function to be optimised (the parameter set) is stochastically adjusted, with adjustments that worsen the system performance by a factor (z) being accepted with a probability defined by the Boltzmann distribution:

$$P = \frac{1}{Z} e^{\frac{-z}{KT}}$$

where K is the Boltzmann constant and T is a parameter corresponding to temperature.

Simulated Annealing was also used in conjunction with Boltzmann Machines, (Ackley, Hinton, Sejnowski, 1985). It achieved a lot of attention as a technique of global optimisation. Its properties have been extensively studied in the framework of nonhomogenous Markov Chains, (van Laarhoven, Aarts, 1987). Although in theory it may escape from the local minima, the cooling scheme assuring asymptotic convergence to the global minimum is slow and in practice one has to trade off speed for a premature convergence.

As we stated above Simulated Annealing was inspired by the physical process of gradual and controlled cooling of a frustrated system in the thermal equilibrium. Extremal Optimisation, (Boettcher, Percus, 1999), on the other hand draws upon a model used to simulate the dynamics far from equilibrium, namely on the Bak-Sneppen model of evolution, (Sneppen et al., 1995). In spite of the connections between the Extremal Optimisation and random search procedures based on biological processes it was mainly discussed in physical literature and therefore we present it in this subsection.

Extremal Optimisation embeds the concept of self-organised criticality, (Bak, Tang Wiesenfeld, 1987), and can be described as progressive elimination of the least fit components. It is performed by selection of the lowest fitness species for adaptive changes. In contrast to Simulated Annealing, Extremal Optimisation applies no decision criteria whether to accept or reject new configurations. The authors report that Extremal Optimisation performs very well on selected benchmark problems and can be compared with much more elaborate optimisation techniques.

2.3.2. *Biologically inspired search algorithms.*

Among optimisation techniques developed on the basis of some biological processes we can make a distinction between algorithms inspired by a natural evolution and those based on the behaviour of some organisms.

2.3.2.1. *Evolutionary Algorithms.*

There is a whole family, Evolutionary Algorithms, (Back, 1995), of search and optimisation mechanisms having their origins in biological evolution. They include Genetic Algorithms, (Holland, 1975), Evolution Strategies - developed jointly by Bienert, Rechenberg and Schwefel in 1960’s, (Back, 1995), and Evolutionary Programming, (Fogel, 1962). The general idea underlying Evolutionary Algorithms is to evolve a population of individuals corresponding to the potential solutions of an optimisation problem and represented by appropriate parameter sets in order to obtain a globally optimal solution, where optimality criterion is encoded in the fitness function. The population is manipulated by a number of operations inspired by genetics, especially mutation and recombination, and undergoes a fitness-based selection process. There are some important differences between particular members of the Evolutionary Algorithms family, therefore we will describe them separately in turn.

Genetic Algorithms were introduced by Holland, (Holland, 1975), as a search algorithm based on a model of Darwinian natural selection and genetics. The parameter set describing a potential solution to a given problem is encoded as a binary string and a population of encoded parameter sets, or genes, is evaluated to determine their measure of fitness. New parameters sets are probabilistically derived from the old ones, the selection mechanism being a function of this fitness measure. Each such cycle of evaluation and derivation is defined as one cycle of evolution. In Genetic Algorithms a dominating role is assumed by recombination operator, of which there are many different variants, e.g. uniform crossover, m-point crossover to name a few. Mutation is considered mainly as a background operator and is realised as a bit inversion. There is a possibility to include into the algorithm simple constraints imposed by the optimisation task by choosing appropriate encoding. Genetic Algorithms are not self adapting - the parameters describing operation of the algorithms have to be set a priori and are kept fixed during the search. The interest in Genetic Algorithms grew quickly since their inception and resulted in very diverse mechanisms being investigated in the context of numerous applications, (Davies, 1991; Goldberg 1989). As a result the theoretical background behind them is rather weak. As Back points out the proofs of convergence with probability one exist only for simplified elitist form of Genetic Algorithms, (Back, 1995), and the rationale behind diverse versions of Genetic Algorithms is loosely rooted in natural evolution and numerous simulations.

Another example from this search family is Evolutionary Strategies. They differ from Genetic Algorithms in several aspects. First, they have a real-valued problem representation. Second, the main operator modifying the parameter sets is (Gaussian) mutation and not recombination. The latter, however is still present and is important for self-adaptation of Evolutionary Strategies. Self-adaptation is understood as incorporation into the optimisation process not only the parameter sets, describing potential solutions, but also of the main algorithm parameters and takes into account the local topology of the objective function. Recombination is used as a means of modification of the strategy parameters, i.e. standard deviations and rotation angles of normally distributed mutations. In contrast to Genetic

Algorithms the selection mechanism is deterministic in Evolutionary Strategies and it excludes the worst fit individuals from further evolution.

Yet another algorithm of this group is Evolutionary Programming. Similarly to Evolution Strategies it has a real valued encoding of the optimisation problem and uses a Gaussian mutation as the only genetic operator. In this respect it actually differs from both Evolutionary Strategies and Genetic Algorithms which both use recombination but put different emphasis on its importance. Evolutionary Programming shares with Genetic Algorithms lack of self-adaptation and selection mechanism probabilistic in nature rather than deterministic.

In spite of different attempts, Evolutionary Algorithms still lack the proper theoretical treatment with results concerning global convergence or the convergence rate limited only to the simplest forms of these algorithms, (Back, 1995).

The Evolutionary Algorithms are, similarly to SDS, based on a population dynamics exploring a biased random walk for finding the global optimum. The current form of SDS uses a discrete problem representation, so it is with this respect more similar to Genetic Algorithms than to Evolution Strategies. One of the main differences between these algorithms is however the evaluation of partial solutions. The Evolutionary Algorithms are based on an explicit representation of the fitness function and they determine the goodness of fit of potential solutions by a deterministic evaluation of corresponding fitness values. SDS, on the contrary, does not use explicit fitness function and the evaluation of an agent’s performance is based on local, or partial evaluation of a match between the pattern and a position in the search space. Moreover, this evaluation is based on using a random micro-feature selection. Probably an even more fundamental distinctive feature of SDS is the fact that, contrary to all Evolutionary Algorithms, the representation of the solution is formed collectively by a dynamic cluster of agents. In Evolutionary Algorithms every parameter set constitutes a full and static representation of the potential solution and both, plasticity and restructuring of representations, come as a result of a co-evolution of the whole population of the potential solutions. This is in a stark contrast to the way the representations are formed in the Stochastic Diffusion Search. In the latter system no readout can be made from a single agent because it does not contain a sufficient amount of data that would allow for a determination of a desired solution. Moreover, single agents are always in motion, switching between different solutions and thus allowing for a constant restructuring of the system. Only dynamic clusters of agents may constitute meaningful and stable representations although stable in this context does not imply static.

2.3.2.2. *Mechanisms based on organism's behaviour.*

Finally it is worth mentioning another stochastic optimisation procedure called Chemotaxis, (Bremermann, Anderson, 1991). Its operation resembles movement of a bacterium in the environment containing chemoattractants. A bacterium initially generates a random direction and continues movement in this direction until the concentration of a chemoattractant increases. If the concentration seaze to increase in this direction, then bacteria changes randomly a direction and moves further along a new path. This procedure allows bacteria to swim in the direction of gradient ascent of the chemoattractant. The Chemotaxis algorithm corresponds to optimisation of a function corresponding to the concentration of the chemoattractant in the medium. If the direction of gradient ascent is discovered, the algorithm moves along it and the step size is increased. Otherwise, the system reverts to the last solution and a new random direction is generated.

Although the algorithm, like the Evolutionary Algorithms, involves a random walk it is not based on a population of solutions and thus may get trapped in the local minima due to following the steepest ascent of the objective function. However, Bremmerman and Anderson report that it is relatively robust and often escapes local minima, (Bremmerman, Anderson, 1991). The authors provide interesting arguments in (Bremmerman, Anderson, 1991) and (Anderson, 1997) for the hypothesis that the random walk based algorithm underlies the learning in the brain and argue specifically that this role is played by Chemotaxis.

Chemotaxis, in the limit, has been described and analysed as a reaction-diffusion equation, (Alt, 1980).

2.4. *Multi Agent Architectures.*

The field of Multi Agent architectures is very rapidly expanding and encompasses very different concepts and technologies. This diversification resulted in a difficulty of formulation of a sufficiently general but at the same time useful definition of an agent. This explains why such a definition of an agent was formulated relatively late, (Franklin, Graesser, 1997).

The simplest characterisation of agent software is a program performing some prespecified task without explicit user input. The fundamental aspect of an agent's operation is its autonomy. Another crucial property of agents is their ability for "social" interactions between each other and with humans, (Wooldridge, 1995). An agent should also be able to react and respond to the changes in its environment. The above characteristics encompasses what is known as soft or weak agents, (Wooldridge, 1995). Hard agents are capable of adaptation and learning in order to improve their performance or to account for changes in the environment.

One of the research directions investigated within the broad framework of multi agent architectures and artificial life is, closely related to the agent's communication behaviour, an emergence of the functionality of a multi agent system. Due to interactions of agents with their environment, between themselves and with human users, and because of the considerable autonomy of a single agent's performance, the overall behaviour of the multi agent system cannot be easily deduced from the specifications of the tasks of single agents. The emphasis on the role of interaction and distribution of the overall task between relatively autonomous components, dynamically coupled to the environment, situates multi agent systems at the heart of, what Wegner calls, a paradigm shift from algorithmic to interactive systems, (Wegner, 1997). Wegner argues that such systems effectively surpass limitations of systems which can be described by the Universal Turing Machine formalism and thus offer a more powerful information processing paradigm.

2.4.1. *Pandemonium-type models.*

Probably one of the first systems that could be classified as a multi agent system was so called Pandemonium proposed in cognitive science by Selfridge, (Selfridge, 1958). Pandemonium consists of a hierarchy of sets of demons performing different tasks. The first set of demons record the first image of the external signal which is further analysed by the set of feature demons. The latter are specialised and search for particular features in the image. Selfridge used steepest descent in order to fix the weights determining a specialisation of a given demon.

In the next stage cognitive demons, observing the feature demons, recognise prespecified patterns. Cognitive demons signal presence of the features, found by feature demons, supporting the possibility of existence of their respective patterns. The more such features is present, the stronger is the signal of a given demon. Finally, a decision demon observes the cognitive demons ('listens' to their collective 'shouting', hence Pandemonium) and selects the demon producing the strongest signal. The pattern corresponding to the chosen signal is the one that the Pandemonium considers as the most likely given the observed evidence. Pandemonium is more powerful than the direct template matching because it hierarchically analyses the signal on increasing levels of complexity and can recognise the patterns despite possible transforms. As noted in (Lindsay, Norman, 1977), Pandemonium can learn if constituent demons would be made to gradually learn how to recognise the features corresponding to their corresponding patterns. Adding the contextual demons would extend the system to cope with background knowledge effects. However, it is necessary to observe, after (Lindsay, Norman, 1977), that the ability of Pandemonium to recognise patterns is constrained by the character of basic features that are extracted from the external signal by the pattern demons. In the basic Pandemonium this features are predetermined a priori by its designer.

It is worth noting that the pandemonium architecture had an influence on the creation of the multiple drafts metaphor of consciousness, (Dennett, Kinsbourne, 1992; Dennett, 1999).

The original system has been extended by Jackson in an attempt to propose an architecture implementing a mind on a computer, (Jackson, 1987). He proposes to extend the basic Pandemonium beyond perception and build a whole system of demons, some of which would process and propagate external and some - exclusively internal information. He envisages a stage metaphor in which only a few demons stay in the arena, causing majority to shout. The group in the arena is not static - the loudest demons from the crowd replace constantly the ones in the arena, which return to the crowd. In such a formulation the proposed model bears a similarity to another metaphor of consciousness - the global workspace theory of B. Baars, (Baars, Newman, 1992).

Jackson incorporates learning into the system by introducing variable links between demons in the arena, which are proportional to the time of their common activity in the arena. He proposes to explore this mechanism to achieve associativity, an ability of sequential processing and merging of demons in order to create demons corresponding to new concepts. Jackson maintains that such a system would mimic several aspects of human cognition, like flexibility, the speed of processing vast amounts of information on different levels of complexity, a seemingly nondeterministic operation, deduction, induction, continuous and smooth shifts between perception and complex actions etc.

Although Jackson does not discuss any practical realisation of his theory, recently it acquired an application in systems proposed by Franklin, (Franklin, 1999). The latter author proposed a computer architecture implementing Baars's global workspace theory of consciousness in which he incorporated a form of the pandemonium system envisaged by Jackson.

Another system of a pandemonium type has been recently applied to recognition of hand-written characters, (Larsen, Bundesen, 1996). The authors quote psychological and cognitive arguments supporting the hypothesis that “visual pattern recognition is based on position-wise comparison of stimulus patterns with memory representations”. The simplest scheme capable of such operations is based on template matching. This view is compatible with the alignment approach to the invariant recognition proposed by Ullman, (Ullman, 1995), and discussed above in the context of his sequence seeking architecture. Larsen and Bundesen discuss critically the common view about insufficiency of template matching based methods for recognition of real life patterns. They recognise that the recognition rates achieved by a traditional template matching approach cannot match the human performance but they claim that this is caused not so much by inherent problems of template matching as by its ‘uneconomical’ use. The traditional template matching uses information from the best matching template to perform classification of input. The authors maintain that by utilising the pandemonium architecture it is possible to extend this approach to use the degree of match with different templates as evidence, positive or negative, that can be used for any classification.

The model built by Larsen and Bundesen contains a number of demons corresponding to particular templates. An input character is at first normalised with respect to position and scale and convoluted with a two-dimensional Gaussian filter. Next, all demons determine the degree of matching between their corresponding templates and the character by finding a maximum of the correlation between the two characters as a function of some allowable displacement between the centres of gravity of both patterns. The next level of processing consists of cognitive demons, which are fully connected to feature demons. This input is a weighted sum of degree of matches determined by the feature demons. The response of the system is provided by the highest in the hierarchy decision demon detecting the ‘loudest’ cognitive demon.

The system initially learns the classification of digits. During the first pass through the training set each incorrectly classified pattern is added to the set of templates corresponding to the correct digit class. During consecutive passes no new templates are added. Instead, the weights linking the template and class demons are updated according to some form of the delta rule strengthening the role of useful demons and weakening it for the demons with small contribution to the classification process.

The progress of learning is evaluated on a separate test set. At the end of the training the demons corresponding to templates with the smallest contribution to the recognition of corresponding digit types are removed. Authors report attaining a recognition rate of 95% with about 28 templates per character class, which is short of human performance by only 2-3%. They could achieve the accuracy of 89%, equal to that of the traditional template matching, by reduction of the average number of templates per digit class to 8. This is a substantial increase in effectiveness, as compared to a traditional template matching system, which needed for this level of performance about 60 templates, (Larsen, Bundesen, 1992). From the analysis of the system's operation the authors conclude that its performance was most significantly affected by using, as a basis of classification, a collective response of many templates instead of the best template only.

Many models of pattern recognition are based on defining some canonical, independent dimensions of visual form, which constitute the basis for a description of any pattern. Larsen and Bundesen suggest that the template matching pandemonium model imply that this approach may be inappropriate because, although there may be many degrees of variation of visual form, they nevertheless may be neither universal nor mutually independent or orthogonal.

Smieja proposed a related architecture based on the pandemonium model, (Smieja, 1996). Agents used in this system comprise of feedforward neural networks adapting to solve a given task with the use of a standard backpropagation. The system solves the problems by dynamic modularization of the input space.

An agent in fact comprises of two sub-networks and, as a result, it produces a vector output. One component of this vector corresponds to the standard output and the other - to the 'self-assessment' (a confidence function) by the agent of its own performance. This extension makes agents reflective. In other words, agents produce additional signal, which can constitute a basis for the decomposition of the input space into regions of their competence. This in turn allows the Pandemonium to allocate dynamically the most suitable agent to processing particular input data. The decomposition of the input space between agents is achieved automatically and simultaneously with their learning of the solution for corresponding domains of expertise. Thus the pandemonium architecture proposed by Smieja uses in fact two degrees of freedom, the concept first proposed to the connectionist community by von der Malsburg. Only the part of the message produced by an agent corresponds to the standard output and has an externally interpretable meaning in terms of the contribution to a problem's solution. The other part is used for the internal organisation of system's operation - the 'self-assessment' information is mainly used for the internal co-ordination of different agents.

This architecture was used by the author to approximate piecewise continuous functions, for which problem the above mentioned architecture obtained automatically a division of the input space between different agents and performed better than a uniform feedforward network. Smieja discusses also the application of the architecture to Boolean and classification problems, in which domains however he changes the basic architecture of the agent and the method of operation of the whole system - the dynamic decomposition of the input domain. In fact he proposed three different methods of decomposition, depending on whether the system is to approximate a mapping between continuous spaces, a mapping from low-dimensional input space into Boolean output space (classification) or mappings from a high-dimensional binary input space into Boolean output space (parity).

For all problems reported by the author the pandemonium architecture converged to the solution much faster than the feedforward network trained with a backpropagation. In contrast to the system used by Larsen and Bundesen the agents in Smieja's pandemonium were allocated exclusively to the separate regions of the problem domain, thus the functionality of the system was achieved mainly through a competition for the domains of expertise between agents. Smieja reports that for classification problems, like the one described in (Larsen, Bundesen, 1996), it is indeed more advantageous to base the operation of Pandemonium on a co-operation of agents, or as he puts it on a team processing, in which agents produce together a range of possible answers used subsequently for the final classification.

The classification of the problems into the ones that allow solution methods based on a domain decomposition and the others supporting collective, or team, processing is very interesting and Smieja validates his claims with carefully chosen experiments illustrating his conclusions. However, he does not provide any formal model of the discussed architecture, relying primarily on insight and simulations in order to derive its properties.

2.4.2. *Genetically inspired multi agent systems.*

Finally, we will describe an example of multi agent architectures inspired by certain natural systems, namely multi agent architectures based on genetics. In fact, all Evolutionary Algorithms fall into this category of multi agent systems. However, as they were discussed already in the context of random search and optimisation methods, we will present here another, related architecture proposed for the image segmentation, (Liu, 1998).

In order to identify different homogenous regions in an image, agents of this system perform various actions in response to the local statistical properties of images defined at the pixel level. The pixels, whose neighbourhood satisfies certain predefined constraints, are called feature pixels. Thus the goal of the agents is to identify the homogenous regions by labelling the feature pixels. The repertoire of an agent's actions consists of marking the feature pixels, self-reproduction, diffusion or death and is described below:

- image pixels labelling - as a consequence of a discovery of a feature pixel an agent marks this pixel with a label;

- self-reproduction - after finding and labelling a feature pixel, an agent produces a number of offspring agents which will explore the vicinity of its location;

- diffusion -an agent failing to discover a feature pixel moves in a direction determined from its successful parent but at a randomly generated distance;

- ageing - as a result of a failure to discover a feature pixel an agent decreases its finite life span by a unit. An agent, whose life span decays to zero, dies and disappears from the population.

The evolution of an agent on the image can be described as a biased random walk, in which bias is introduced via determination of a direction of motion from the directions of successful parents and a requirement of remaining in the vicinity of a parent's location. The method of selection of parent agents is based on a fitness function and thus resembles selection methods used in Evolutionary Algorithms. The end of operation of an algorithm is marked with decay of all agents.

Liu presented two experiments in image segmentation in which the proposed method successfully discovered borders and homogenous regions in the brain scan images and detected edges in artificial images. As in many cases discussed in this chapter also here the formal analysis of the system's performance is lacking.

In summary, clearly SDS shares many similarities with the architectures discussed in the last section. It is probably more interesting to point out a few important differences. In the majority of architectures presented here agents are more specialised than in SDS, in which agents have a uniform functionality. The architectures of the systems enforce more local processing than is the case in the Stochastic Diffusion Search, where the system has an access to the whole search space which agents can freely explore. It follows that, in spite of their universal functionality, SDS's agents seem to be more autonomous in that, at any given moment they may process different external input and, what is more important, the exact form of this processing is not fixed but in part determined by agent's history. Indeed, the difference between most of architectures and SDS is not unlike the difference in a mode of processing between standard neural networks and Genetic Algorithms. This view is supported also by the fact that, in contrast to models of Selfridge, Larsen and Bundesen, or Smieja, agents in SDS are not adaptive nor the system as a whole, in its current form, can learn.

2.5. *Conclusions.*

In this chapter three possible classifications of SDS were presented, each stressing different aspect of the algorithm. SDS can be considered as a system performing pattern matching, and as such, it solves probably the most difficult form of this problem, namely the best fit matching. The Stochastic Diffusion Search, due to its generic formulation, can be used in different search spaces irrespective of the constituting them data structures.

The character of the problem solved by the Stochastic Diffusion Search together with the probabilistic operation of the algorithm enable its characterisation as a random search method. The main similarity between SDS and many algorithms in this class is a population-based operation and a biased random walk character. The distinctive feature of SDS in this context is the indirect use of a measure of fitness. As opposed to many random search procedures, in which the fitness function is evaluated completely, the evaluation of the potential solution in SDS is incomplete and probabilistic. This contributes to the flexibility of the algorithm and is largely responsible for its ability to adapt and achieve the best fit solution.

Finally, it is possible to assign SDS to a class of multi agent architectures. However, in spite of many similarities, it can also be distinguished from many of the architectures, discussed in the last section of this chapter, by a higher level of autonomy of agents.

From the presented review it should be clear that SDS is an interesting algorithm standing at the intersection of many important research areas. Many of the systems presented here are not supported by a formal analysis of their properties, and their characteristic usually follows from empirical evaluation and, often superficial, similarity to some natural processes. This, however cannot be a satisfactory situation. There is a strong need to develop formal theories behind such methods. It is especially important to obtain a formal analysis in the case of the Stochastic Diffusion Search, because of the universal character of the algorithm. Partial results were already obtained in works of Bishop and Torr, (Bishop, Torr, 1989), and Grech-Cini, (Grech-Cini, 1995), but one needs a more thorough characterisation of the algorithm. In the remaining chapters of this thesis such a formal analysis will be provided. Introducing a mathematical model will make it possible to derive many properties of SDS and will lead to characterisation of its behaviour in many search conditions. Next chapter will introduce a detailed definition of SDS together with its Markov Chain model. This will facilitate a discussion of the most fundamental properties of the system and will constitute a starting point for further analysis.

CHAPTER 3
THE STOCHASTIC DIFFUSION SEARCH AND ITS MARKOV CHAIN MODEL

3.1. *Introduction.*

This chapter provides a definition of the Stochastic Diffusion Search and explains in detail the fundamental mechanism underlying its performance. It also introduces a Markov Chain model of SDS and will discuss some of the most fundamental properties of the algorithm that can be proven on its basis, namely its convergence and time complexity. It will appear that SDS converges to a fixed state in the special case and, in general, it achieves a statistical equilibrium. In fact, although in such an equilibrium the Markov Chain model of SDS should visit all possible states of its state space, in practice it stays in a quasi equilibrium restricted to a relatively small fraction of the state space.

The algorithm, without any pre-defined knowledge about the domain of the problem achieves a linear time complexity, which situates it among the best search algorithms.

3.2. *Stochastic Diffusion Search.*

The purpose of the Stochastic Diffusion Search is to locate a predefined data pattern within a given search space or, if it does not occur, its best instantiation. Other possible partial instantiations of the data model are treated as noise or disturbances influencing the search.

An example of the task SDS is to solve could be locating of a familiar person in a 2D image of a crowd. Due to, for example, occlusions the instantiation of the template constitutes only a partial match. The task is to locate this person in spite of other partial matches present in the scene (e.g. other people with similar clothing and postures, etc.). In general this is a nontrivial problem for an artificial visual system.

The situation described above can be generalised to other potential settings - the search can be performed in an n-dimensional space, where objects are represented by collections of points representing micro-features - basic building blocks of the space. One can also imagine that the search space is represented by a graph, in which case one is interested in locating a pattern represented as another sub-graph.

The search space as well as data model are defined in terms of the basic micro-features, from which they are both comprised. Thus, macroscopic features of objects from a search space can be effectively defined in terms of appropriate micro-features and their relationships. Micro-features can be thought of as single pixel intensities, if the search space is a bit map image, or can constitute some higher level properties like vertical and horizontal lines, angles, semicircles etc. if the search space and the target are described in terms of these properties, or they can be letters (with the search space being a text), or nodes of a graph (the search space would be a graph with a pre-specified neighbourhood). The locations of micro-features common to the object in the search space thus constitute partial solutions to the search.

The search is performed in parallel by a pre-specified number of elements called agents. Each agent acts autonomously, and in parallel with the others tries to locate the position of the target in the search space. The position of the target is represented as co-ordinates of a pre-defined reference point in the target's description. The transmission, or diffusion, of information (the exchange of co-ordinates of potential reference point locations) enables agents to communicate with each other and to allocate dynamically computational resources depending on the outcome of the search.

Depending on their performance in the search agents can become 'active', if they point to potentially correct positions within the search space, otherwise they remain 'inactive'. All agents have an access to the search space and to the description of the target object.

The search can be considered as a competitive co-operative process in which all agents independently seek for partial solutions. The latter, once found, compete to allocate other agents to their positions.

In this way competition transforms smoothly into co-operation, when more and more agents are attracted to explore the potential fit to the data model. This competition for co-operation assures that all potential positions of the object will be examined independently, with the most promising one, over a number of iterations, attracting most of the computational resources. Thus, the correct position of the best fit to the data model will emerge from independent, parallel exploitation of different potential positions in the search space by gradually disregarding misleading partial matches. From this principle it follows that agents would cluster over interesting positions in the search space as soon as first agents pointing to these positions spread information to others. The mechanism responsible for this diffusion of information is probabilistic in nature and consists of assigning activity to the agents and, based on this activity, selection of agents for communication.

The Stochastic Diffusion Search involves several stages and can be succinctly summarised in the algorithmic form as follows:

```
INITIALISE(AGENTS);
WHILE NOT TERMINATE(AGENTS) DO
    TEST(AGENTS);
    DIFFUSE(AGENTS);
END;
```

The remainder of this section will describe details of the operation of SDS and the algorithmic steps stated above.

The Stochastic Diffusion Search effectively performs the best possible match between existing objects in the search space and the description of the target object. It follows that SDS is able to find the target if it exists in the search space, otherwise it will locate an object with the most similar description to the target.

Initially all agents are randomly initialised to reference points in the search space (e.g. if the task to solve is to locate a word ABRACADABRA in Encyclopaedia Britannica, then agents will point to possible positions of the first ‘A’. The simplifying assumtion is that a word ABRACADABRA does occur in Encyclopaedia Britannica). They are also initially all set as inactive.

Next, each of the agents independently performs a probabilistic check of the information at the reference point, by comparing a random micro-feature from the target object (e.g. one of the agents could have chosen a left-most letter ‘R’ from ABRACADABRA) with a corresponding one in the search space (i.e. the third letter to the right of the reference point). If the test is successful then the agent becomes active otherwise it remains inactive.

In effect, the activity of an agent gives an indication of the possibility that it is pointing to the correct position. However, due to the partial test (only one letter is checked at a time), the possibility of false positives (i.e. activation on non-target e.g. a word like 'CURVATURE' - the third letter to the right of the reference point is 'R' as in the target) is not excluded nor is the probability of false negatives (i.e. failing to activate on the best match to the target in the case when the ideal instantiation of the target does not exist in the search space -e.g. if 'ABRACADABRA' were misspelled). In this way the Stochastic Diffusion Search can escape from local minima corresponding to objects partially matching the description of the target.

Subsequently in the diffusion phase each, and only, inactive agent selects, at random, another agent with which it will communicate. Depending on whether the chosen agent is active or not the choosing agent will point either to the same reference point as the active one or, if the chosen agent is inactive, will randomly reinitialise its position. Active agents do not sample other agents for communication but they also undergo a new testing phase and, depending on the outcome, may retain activity (if an agent points actually to 'ABRACADABRA' then it will remain active and will always point towards this position regardless of the letter chosen for the testing phase) or become inactive (an agent pointing to 'CURVATURE' would become inactive if it chose for example the sixth letter to the right of the reference point for the next testing phase).

This process iterates until a statistical equilibrium is achieved, which is the basis of the halting criterion used in (Bishop, 1989). The halting criterion monitors the maximal number of agents pointing to the same position in the search space (the more letters a given word has in common with 'ABRACADABRA' the more likely the agent pointing to it is to remain active, and therefore to attract other agents to this word via the diffusion phase). If the number of agents in this cluster exceeds a certain threshold and remains within certain boundaries for a number of iterations then it is said that SDS have reached its equilibrium state and the procedure is terminated.

Even though agents act autonomously and there is only a very weak form of probabilistic coupling, it nevertheless enables agents to develop a co-operative behaviour. In order to discuss the basic properties of the Stochastic Diffusion Search the next section will introduce a model of SDS based on Markov Chain theory (Kemeny, Snell 1976). This will also faciliate an introduction of a notation that will be used in subsequent chapters where further analysis of SDS will be presented.

3.3. *Markov Chain Model.*

In the most general case, the Stochastic Diffusion Search is supposed to locate the target, or the template, or if it does not exist in the search space its best instantiation. Therefore from now on the object sought by SDS will be referred to as the target.

Let the search space size be M (measured as a number of possible locations of objects). Let the probability of locating the target in a uniformly random draw be p_m and let the probability of locating the sub-optimal object (one sharing to some extent common features with the target) be p_d.

Let the probability of a false positive be p^+ and the probability of false negative be p^-. Assume that there are N agents. The state of the search in the nth step is determined by the number of active agents pointing towards the position of the target and active agents pointing towards the false positives (the number of non-active agents will be equal to the difference between the total number of agents and the sum of these two numbers). This is because, by assumption, only active agents are considered as carrying potentially useful information and effectively they influence the search directions of all other agents. Also the strong halting criterion uses only information from active agents.

Thus in effect there is a finite number of discrete states each characterised by the pair of two natural numbers. The Stochastic Diffusion Search changes its state in a random manner and the possible future evolution of SDS can be influenced by the past only via the present state (agents are memoryless and the information about the past evolution of SDS is contained in its current configuration) thus effectively it can be modelled by a Markov Chain.

In order to specify the Markov Chain model it is necessary to construct the transition matrix. Let the state of the search in the nth step, denoted X_n, be specified by a pair of integers (v,b), where v denotes a number of active agents pointing towards the target and b the number of active agents pointing towards false positives.

If at the nth step an agent is active and points towards the target then it will become inactive with probability p^-, otherwise it will remain active.

Similarly, an active agent pointing towards the false positive will remain active with probability p^+, otherwise it will become inactive.

The one step evolution of the non-active agent is determined first by the outcome of the diffusion phase and then by the testing phase. One of its possible evolutions is described here. During the diffusion phase a non-active agent will choose an active agent pointing towards the target with probability v/N and then will remain active with probability $1-p^-$. The other possibilities follow in an analogous way and are best summarised in Figure 3.1.

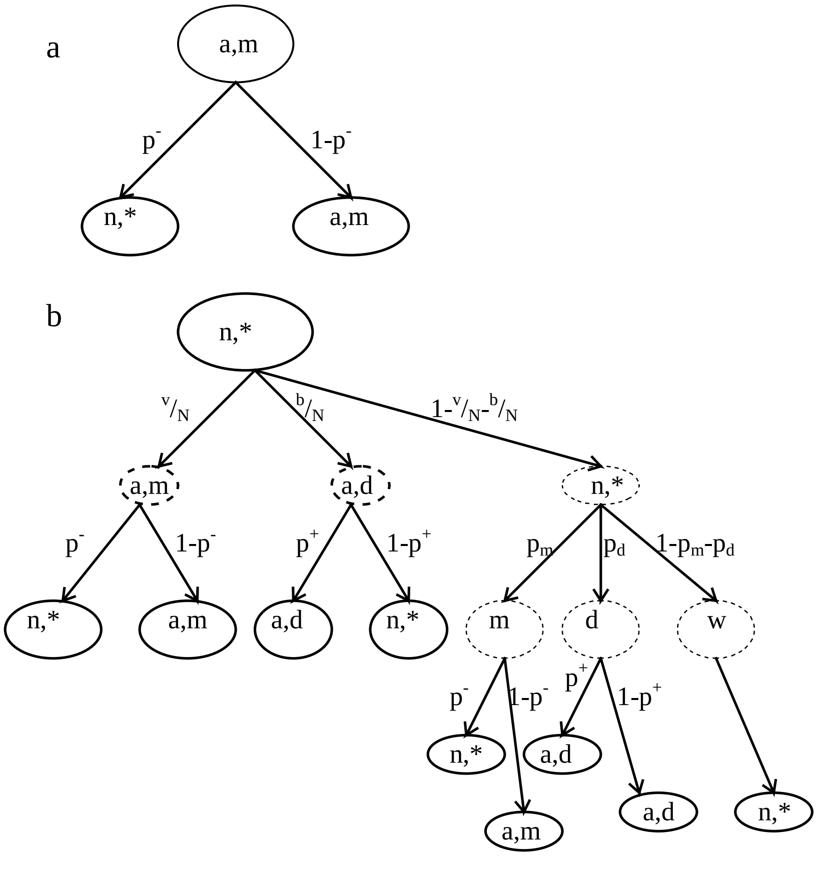


Figure 3.1 (a) One step evolution of an active agent pointing to the correct position. (b) One step evolution of an inactive agent. Probabilities of changing states are given next to appropriate arrows. Intermediate steps are marked by dashed ovals. One step evolution of an active agent pointing to a disturbance is analogous to (a).

It is apparent that a transition from one state to another can take place in many different ways, depending on the performance of all agents (e.g. number of non-active agents can increase by one, because during one iteration an active agent pointing towards a false negative failed the test phase or two active agents pointing to the target became inactive and one inactive agent became active and so on). The one step probabilities of transition from one state to another result from summing probabilities of all possible ways that this particular transition can be achieved, see Figure 3.2.

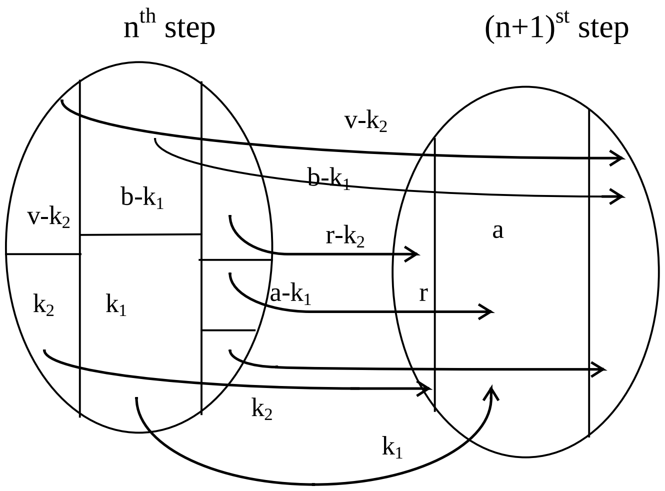


Figure 3.2 Diagrammatic illustration of a one step evolution of the Stochastic Diffusion Search. In the *n^th^* step there are *v* active agents pointing to the correct solution and *b* active agents pointing to the disturbances. In the *(n+1)^st^* iteration these numbers change to *r* and *a* respectively.

The exact formula is given below

$$P\left\{X_{n+1} = (r,a)\middle|X_n = (v,b)\right\} = \sum_{k_2}^{\min(v,r)} \sum_{k_1}^{\min(b,a)} Bin\left(k_2, p^-\right) Bin\left(k_1, p^+\right) Mult\left(k_1, k_2, r, a, v, b\right)$$

where

$$Bin(k_2, p^-) = \binom{v}{k_2}(1 - p^-)^{k_2}(p^-)^{v-k_2}$$

$$Bin(k_1, p^+) = \binom{b}{k_1}(p^+)^{k_1}(1 - p^+)^{b-k_1}$$

$$Mult(k_1, k_2, r, a, v, b) = \binom{N - v - b}{r - k_2} p_{ab}^{r-k_2} \binom{N - v - b - r + k_2}{a - k_1} p_{af}^{a-k_1}(1 - p_{ab} - p_{af})^g$$

and

$$p_{ab} = \frac{v}{N}(1 - p^{-}) + (1 - \frac{v}{N} - \frac{b}{N}) p_m (1 - p^{-})$$

$$p_{af} = \frac{b}{N} p^{+} + (1 - \frac{v}{N} - \frac{b}{N}) p_d p^{+}$$

$$g = N - r - a - v - b + k_1 + k_2$$

and double summation in the above formula is over such k_1, $k_2 \geq 0$, that $g \geq 0$.

The term Bin(k_2,p^-) denotes the probability that k_2 out of v active agents pointing towards the target will remain active after the test phase and v-k_2 agents will become inactive. Similarly, the term Bin(k_1,p^+) denotes the probability, that k_1 out of b active agents pointing towards false positives will remain active after testing and b-k_1 of them will become non-active. The term Mult(k_1,k_2,r,a,v,b) expresses the probability of r-k_2 out of N-v-b inactive agents starting to point towards the target and passing the test phase, a-k_1 out of N-v-b-r+k_2 agents starting to point towards false positives and become active and remaining agents staying inactive.

Finally observe that the above formula can be extended for the cases when p^-, p^+ are equal to zero by calculating the limit of the transition probability with p^-, p^+ tending to zero respectively.

Let S denote a given search space. Let f_n^s denote the number of active agents pointing to the same position s in the search space S in the n^{th} iteration. It is easy to see that the following condition is fulfilled: $\sum_{s \in S} f_n^s \leq N$, where N is the total number of agents.

Let z_n denote the maximal number of active agents in the n^{th} iteration pointing to the same position, $s_n^z \in S$ in the search space, i.e. $z_n = \max_{s \in S}(f_n^s)$. Then, from (Bishop, 1989), the definition of convergence of the Stochastic Diffusion Search, is formulated in Definition 1:

Definition 1. Strong halting criterion.

The Stochastic Diffusion Search has reached equilibrium, if

$$\underset{a,b>0}{\exists}(2b < N \wedge b + a \leq N \wedge a - b \geq 0)\underset{n_0}{\exists}\underset{n>n_0}{\forall}\left(\left|z_n - a\right| < b\right)$$

and the solution is the position pointed at by z_n.

Thus the Stochastic Diffusion Search will have reached an equilibrium if there exists a time instant n_0 and an interval (specified by a and b) such that after n_0 the maximal number of agents pointing to the same position will enter and remain within the specified interval. Intuitively, the competitive co-operation process will lead to the allocation of most agents to the best-fit position.

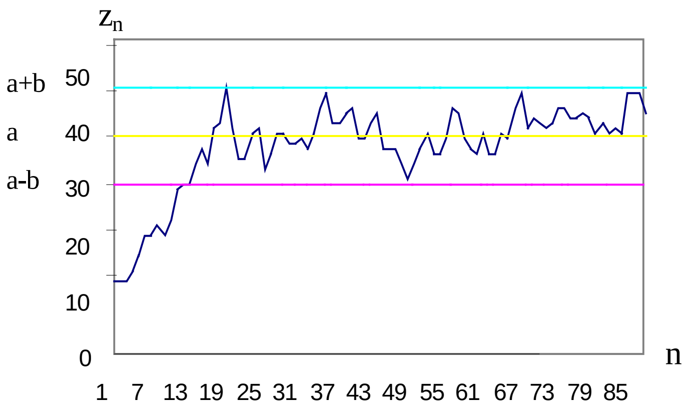


Figure 3.3 A hypothetical evolution of the Stochastic Diffusion Search depicting stabilisation as defined by the strong halting criterion. *n* denotes the number of iterations and z_n the maximal number of active agents pointing to the same position in the search space.

It may also be noted, that the above definition does not require convergence of the process to a fixed state. Indeed, the interval specified by a and b defines a tolerance region. All fluctuations of the maximal number of agents pointing to the same position in the search space are discarded as not important, if they occur within this interval. The conditions for a and b exclude the trivial case in which one would ask only, that $0 \leq z_n \leq N$.

Figure 3.3 shows a hypothetical evolution of the Stochastic Diffusion Search stabilising in the sense of the strong halting criterion. From the graph one can see that the search stabilises around the value a=40 and fluctuates within the band 2b=20.

In (Nasuto, Bishop, 1999) it has been shown that in the case of ideal instantiation of the target in the search space these two parameters do not play a critical role.

3.4. *Convergence of the Stochastic Diffusion Search.*

In (Nasuto, Bishop, 1999) the convergence of SDS has been analysed in two separate cases. First, the case when there existed the ideal instantiation of the target in the search space was inspected.

In the presence of the target in the search space the testing phase for agents pointing to the target becomes deterministic (there is a perfect match, so no agent pointing to this position can ever fail the test).

Let the position of the model in the search space be denoted as s^m. Recall that in the Markov chain model of the Stochastic Diffusion Search the presence of the object in the search space is equivalent to setting p^- to zero.

Thus it holds for $p^-=0$, that

$$P\left\{\lim_{n\to\infty} z_n = N\right\} = 1 \cdot$$

Moreover, $P\{s_n^z=s_m\}=1$, *where* $z_n=max_{s\in S}(f_n^s)$.

For a proof of this statement see (Nasuto, Bishop, 1999). The above proposition proves the intuition that in the presence of the target in the search space all agents will eventually converge on its position. Thus one can see that indeed in this case the parameters a and b do not influence the convergence of SDS.

In the situation when the target is not present in the search space the following result can be proven:

Given $p^-\neq 0$ the strong convergence criterion does not hold in the Stochastic Diffusion Search.

From the proof of the above statement, see (Nasuto, Bishop, 1999), it follows, that in the case of $p^{-}\neq 0$ the model of the Stochastic Diffusion Search is an ergodic Markov chain, (Kemeny, Snell, 1976). Therefore it is easy to see that SDS fulfils another, weaker convergence property:

Given $p^{-}\neq 0$, the Stochastic Diffusion Search converges in a weak sense, i.e.

$$(\exists a > 0)(\lim_{n \to +\infty} Ez_n = a) \cdot$$

A proof of the above property can be found in (Nasuto, Bishop, 1999).

The above characterisations show that in the most general case the Stochastic Diffusion Search convergence has to be understood as approaching an equilibrium in a statistical sense. It means that even after reaching a steady state all possible configurations of agents pointing to the best instantiation of the target as well as to disturbances occur infinitely often according to limiting probability distribution (however some of them may occur very rarely). In practice with appropriate estimates for halting parameters a and b the algorithm will stabilise for a long enough period thus enabling termination.

Also the convergence in the weak sense is crucial for the adaptability of SDS. Effectively SDS allocates a certain amount of computational resources to the best solution found so far. Remaining resources explore the search space attempting to discover other potential solutions. A given optimal solution can become sub-optimal in two ways - either its fit to the target decreases because the similarity criteria change over time or a new, better solution appears in the search space over the course of time. In both cases SDS will be able to find a new optimal solution (due to the agents exploring the search space) and once found, it will rapidly reallocate most of resources towards the new optimal solution.

The case when $p^{-}\neq 0$ constitutes a difficult problem with regard to defining the convergence zone of the Stochastic Diffusion Search, characterised by the parameters a and b. Namely a and b are related to the ability of agents to point towards the best instantiation of the target, (i.e. they are negatively correlated to the probability of false negative) but the latter is not known in advance in the most general case. The issue of determining the parameters a and b in terms of the parameters of the search will be addressed in the consecutive chapters. Although, this is of a great theoretical interest, there remain an issue of setting the parameters a and b for the monitoring the convergence of the algorithm in practical applications. The possible solution, for the user of SDS, is to assume the minimal acceptance level for the object to be recognised and to estimate suitable values of a and b off-line from this acceptance level.

3.5. *Time complexity.*

In (Nasuto, Bishop, Lauria, 1998) the convergence rate of the Stochastic Diffusion Search was characterised in the case when the target existed in the search space and there was no noise. However, the results obtained there should extend over a range of cases in which both of these assumptions were relaxed. This conclusion follows from properties of transition probability matrix of underlying Markov Chain model.

The time complexity estimates were obtained for a synchronous mode of operation of SDS. By this is understood a dynamical mode of operation of the Stochastic Diffusion Search in which all agents perform their tasks and update their states simultaneously. It was found out, using the Markov Chain model introduced above, that the convergence rate of the Stochastic Diffusion Search is $O(-N\log(1-\frac{1}{M}))$ for *M>M(N)* and $O\left(-\log\left(\frac{1}{N}\left(1-\frac{1}{M}\right)\right)\right)$ for M<M(N), where M(N) is given by

$$M(N) = \left\lfloor \frac{N^{\frac{1}{N-1}}}{N^{\frac{1}{N-1}}-1} \right\rfloor$$

(for proof see (Nasuto, Bishop, Lauria, 1998)).

As a corollary it follows that the convergence time of SDS is $O(\frac{M}{N})$ for M>M(N), whereas for M<M(N) the search time grows non-linearly with increase of the search space size and is $O(\frac{1}{\log N})$, (Nasuto, Bishop, Lauria, 1998).

Although one could expect from the theory to find the clear saturation of the average search time for sufficiently small search space sizes, it is not confirmed in an experiment, in which SDS were to find a pre-specified string of letters of an alphabet in a large string - the search space, Figure 3.4.

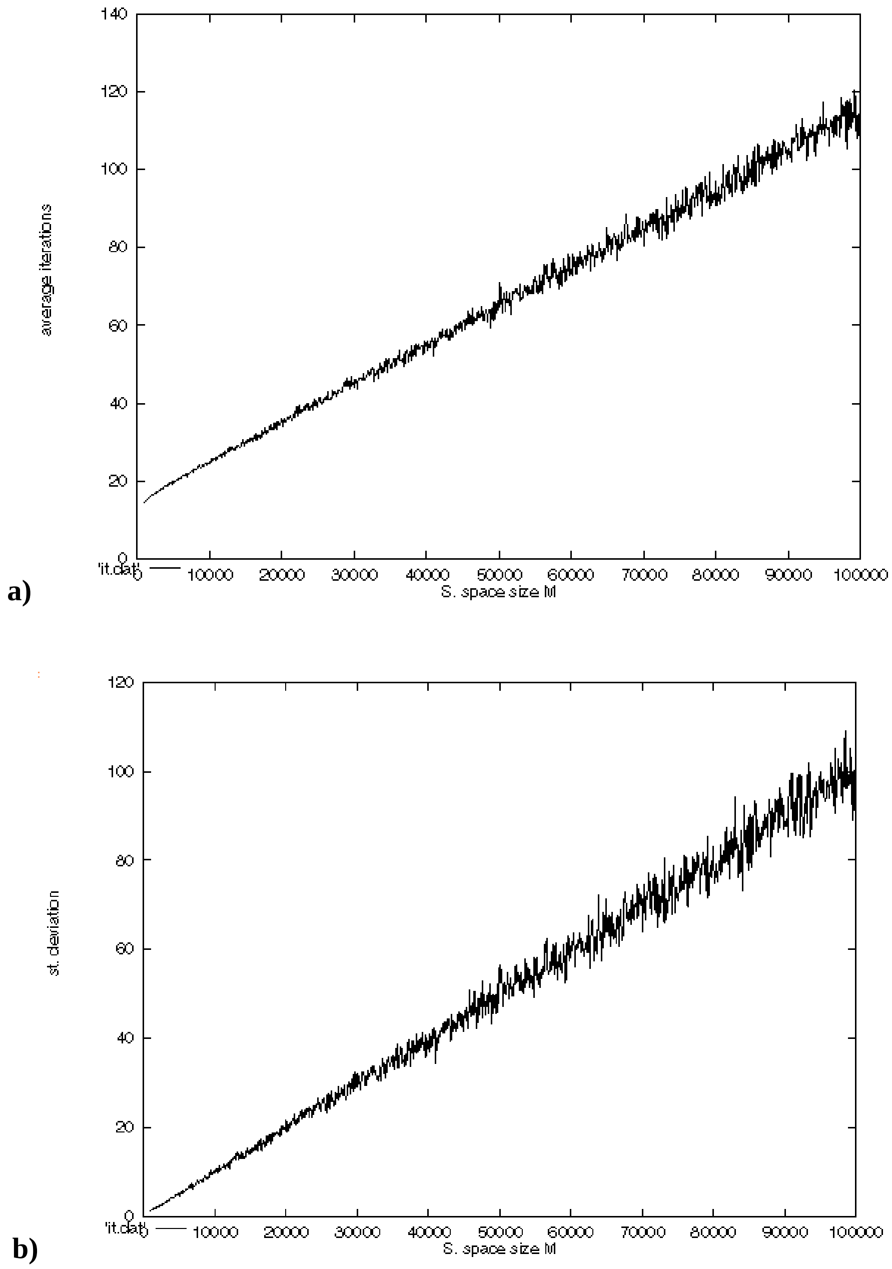


Figure 3.4 Time complexity in a noiseless search space. (a) Mean convergence time of SDS, n, versus search space size M; (b) a standard deviation of the convergence time, n, versus search space size, M. In both panels the average is calculated over 1000 runs; N=1000 agents.

Although the curve of mean convergence times approaches very quickly the asymptotic straight line, Figure 3.4 (a), but there is no easily detectable saturation of SDS convergence time for small search spaces as theory would predict. The reason for this is the fact, pointed out in (Varga, 1962), that asymptotic rate of convergence reflects the behaviour of an iterative scheme for sufficiently large number of iterations which obviously is not the

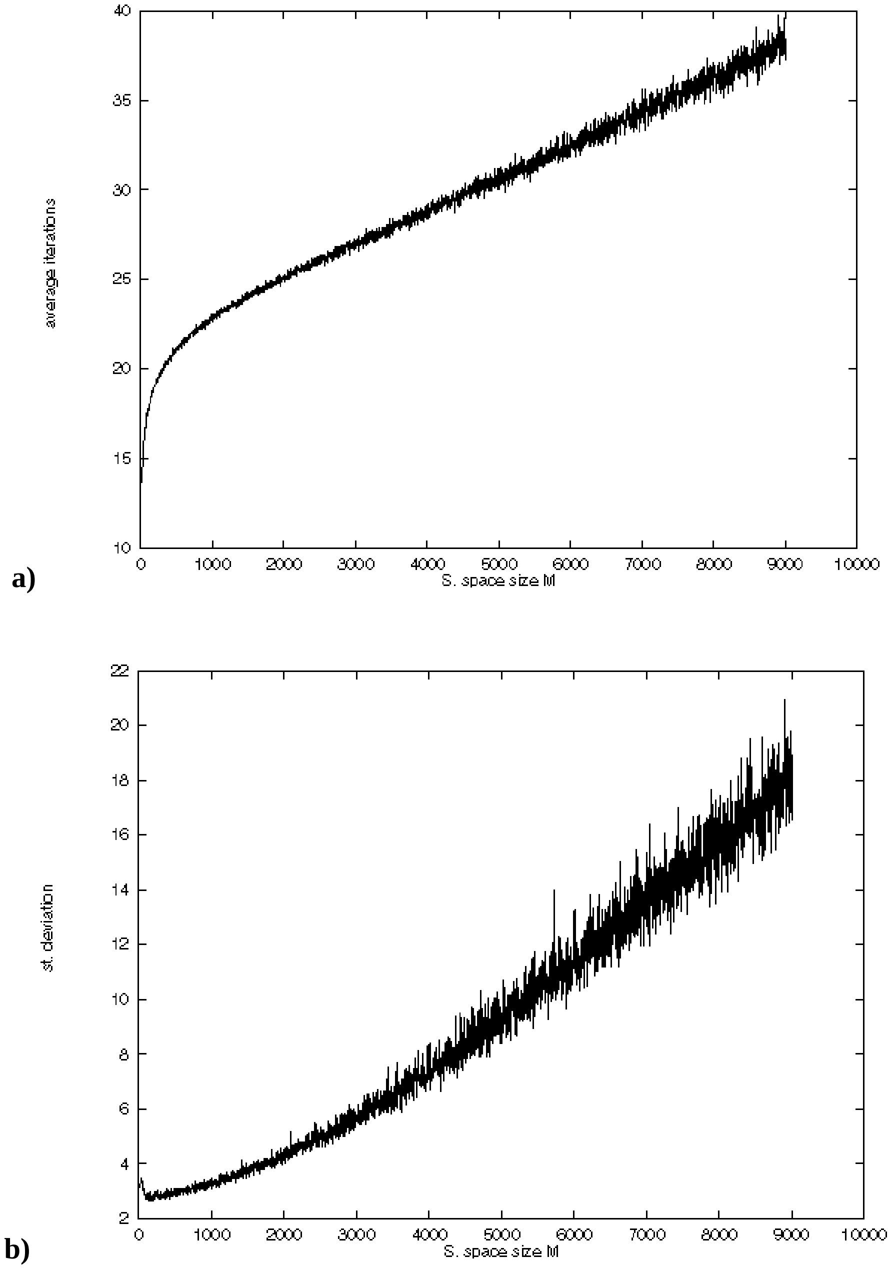


Figure 3.5 Time complexity in a noisy search space. 10% of positions, p_d=0.1, in the search space are occupied by distractors with 60% overlap, p^+=0.6, with the template. (a) Mean convergence time of SDS, n, versus search space size M; (b) a standard deviation of the convergence time, n, versus search space size, M. In both panels the average is calculated over 1000 runs; N=1000 agents.

case for small search space sizes. It may be noted that the standard deviation of the convergence time shows a similar linear trend, the ratio of the standard deviation to the mean being close it follows that, in a majority of cases, SDS convergene times are scattered close to the expected value, however there are cases when convergence is very slow (or very fast) causing this pattern of behaviour.

In order to assess the influence of noise a simulation was performed, in which a moderate number of disturbances was inserted in the search space, see Figure 3.5. In Figure 3.5 one can observe an initial non-linear dependence of the convergence time on the search space size, which changes clearly into linear trend for sufficiently high search spaces. Also the standard deviation, after initial non-linear (concave) dependence on the search space follows a linear trend. From these experiments it follows that overall functional dependence of the convergence time on the search space size is preserved also in presence of noise. One may also note, comparing Figure 3.5 with initial segment of Figure 3.4, that the search time increases in the noisy search space.

In summary one can divide the plane M×N into 2 regions:

1. Region I, in which convergence time depends sub-linearly on the search space size;

2. Region II, in which convergence time increases non-linearly with the search space size but is bounded above.

3.6. *Conclusions.*

In this chapter the Stochastic Diffusion Search was presented as a standalone, connectionist technique capable of efficiently searching for the best fit to a given object in the search space. In contrast to most connectionist search techniques, SDS is capable of finding either the object or - if it does not exist in the search space, its best instantiation without getting trapped in local minima. Experiments performed illustrate that the use of SDS is not restricted to the visual domain for which it was first developed, but can be successfully applied to text string matching problems. The algorithm should be easily extendible to a wide range of possible applications.

The weak convergence of the Stochastic Diffusion Search effectively makes it a truly adaptive algorithm. By such it is understood an algorithm capable of tracking the best solution in a dynamic environment. SDS belongs to this class of algorithms because effectively, even in a statistical equilibrium, agents can be classified as belonging to two classes - one class supporting the solution found so far and the second class exploring the rest of the search space. Thus SDS proposes the best solution found so far but it is still capable of finding a better solution if such would appear somewhere in the search space due to its changes. From theoretical considerations it follows that SDS can be successfully applied in solving problems which are not static but change dynamically in time. The success of the Stochastic Diffusion Search in solving such problems, (Grech-Cini, 1995), follows from its adaptability, the latter being the result of the weak convergence of SDS.

The characterisation of the mean convergence time suggests that the latter depends linearly on the search space size and thus, in terms of time complexity, situating SDS among the best search algorithms, (van Leeuwen, 1990). Also it follows from experiments that SDS is relatively robust to the amount of noise in the search space with respect to both accuracy and convergence time.

The characterisation of the Stochastic Diffusion Search, reported here, did not include the equilibrium behaviour of the clusters corresponding to the potential solutions. In fact, it would be very difficult to obtain such characterisation from the current Markov Chain model, because its state space consists of (M+1)*(M+2)/2 states, i.e. it grows quadratically with the number of agents.

The problem of large state space is quite common in discrete Markov Chain theory. Many researchers address this problem by investigating the conditions, under which it is possible to replace the original Markov Chain with a large state space by another one with a significantly reduced space, obtained from the original one by partitioning it and aggregating, or lumping, all the states belonging to the same partition together. There are different forms of aggregation: lumpability, weak lumpability, (Kemeny, Snell, 1976), exact lumpability, (Schweitzer, 1984; Kim, Smith, 1995), pseudo aggregation, (Rubino, Sericola, 1989), approximate lumpability, (Buchholz, 1994). However, these notions are applicable only to Markov Chains, for which there exists a partitioning of the state space fulfilling some particular property, or for special classes of Markov Chains, e.g. nearly decomposable Markov Chains, (Cao, Stewart, 1985).

Unfortunately, none of the above mentioned techniques of a state space reduction is applicable to the Markov Chain describing SDS. Thus, in consequence, it is necessary to reformulate the model presented in this chapter, in order to address the problem of equilibrium behaviour of the Stochastic Diffusion Search. Such a characterisation of SDS is highly desirable because it would allow to get an insight at the resource allocation management of this algorithm.

In consecutive chapters another (equivalent) model of SDS will be constructed. The model will be related to the classic model from Statistical Physics, namely the Ehrenfest Urn model of gas diffusion. This model will make it possible to prove several properties of the steady state resource distribution by SDS and its analysis will lead to interesting generalisations of the algorithm.

CHAPTER 4

THE BEST INSTANTIATION AND NO NOISE IN THE SEARCH SPACE

4.1. *Introduction.*

This chapter will present an extension of the analysis of the Stochastic Diffusion Search to the case when the target does not exist in the search space. The analysis will characterise the stationary probability distribution of the activity of agents and, as a result, also its first two moments. The chapter will discuss regions in the parameter space with different behaviour of the algorithm. The analysis will also constitute a starting point for further analysis when the assumption about no noise in the search space will be gradually relaxed. The analysis will be based on the observation that the Markov Chain model of the Stochastic Diffusion Search, presented in Chapter 3, can be reformulated in terms of an Ehrenfest Urn model.

This classical model was introduced by Ehrenfests in 1907 to resolve the apparent paradox in the context of statistical mechanics. The paradox concerned the emergence of irreversible macroscopic processes described by the changes of thermodynamic variables (e.g. entropy) from reversible microscopic Newtonian dynamics of single particles. The explanation given by Ehrenfests will acquire an interesting interpretation in terms of information processing in Stochastic Diffusion Search. The next section will first introduce briefly the original Ehrenfest Urn Model and its generalisation due to Whittle, (Whittle, 1986), which will lead to a discussion, presented in the subsequent section, of how the model of SDS arises from the nonhomogenous extension of the generalised Ehrenfest urn model. Section 4.4 of this chapter will characterise the long term behaviour of an agent in the Stochastic Diffusion Search. This, and the consecutive section, describing the steady state probability distribution of agents, will form the basis of the characterisation of SDS’s resource allocation. This characterisation will be illustrated first through a numerical example presented in Section 4.6 and developed further in Section 4.7 discussing the resource allocation as a function of parameters characterising the search conditions. Section 4.8 will present simulations of SDS illustrating and complementing the properties inferred from the generalised Ehrenfest Urn model and the last section will present the summary of the chapter.

4.2. Ehrenfest Urn Model and its generalisation.

It was not clear at the beginning of this century how from time reversible equations of classical mechanics one could obtain irreversibility in the conclusions from the statistical mechanics description, which was in principle derivable from classical mechanics. The model proposed by Ehrenfests was addressing this issue, (Ehrenfest, Ehrenfest, 1907). The model consists of two urns containing in total N balls. At the beginning there are k balls in the first urn and N-k in the other. A ball is chosen at random from a uniform distribution over all N balls and is placed in the other urn. The process relaxes to the stable state. In this state it remains most of the time in a quasi-equilibrium which corresponds to approximately equal numbers of balls in both urns, subject to small fluctuations. Although the original interpretation of the irreversibility in Ehrenfest model is disputable, nevertheless it is possible to account for macroscopic irreversibility in terms of the same model, (Whittle, 1986).

Whittle discusses a generalisation of a classical Ehrenfest urn model, which consists of a collection of N simultaneous and statistically independent 2-state Markov chains governed by the same transition matrix, (Whittle, 1986),

$$\begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{bmatrix},$$

Thus in this formulation a ball moving between the urns corresponds to a single Markov Chain. Its transition probabilities between the two urns: f_{12} and f_{21} are in general not equal.

4.3. Ehrenfest Urn Model of the Stochastic Diffusion Search.

It will be argued that SDS can be recast as a nonhomogenous version of the Ehrenfest urn model described in (Whittle, 1986). In order to extend the model of Stochastic Diffusion Search it is necessary to investigate a one step evolution of a single agent. Consider a noiseless search space in which there exist a unique object with a non-zero overlap with the template - a desired solution. However, it is not assumed here that the solution is an ideal instantiation of the target, thus $p^{-}> 0$. Assume that in the nth iteration there are m active agents. Then, with the notation introduced in Chapter 3 one step evolution of an agent is described by the following transition probability matrix:

$$P_n = \begin{array}{c c} & \begin{array}{cc} (a,m) & (n,*) \end{array} \\ \begin{array}{c} (a,m) \\ (n,*) \end{array} & \begin{bmatrix} 1-p^- & p^- \\ p_1^{\ n} & 1-p_1^{\ n} \end{bmatrix} \end{array},$$

where

$$p_1^{\ n} = \tfrac{m}{N}(1-p^-) + (1-\tfrac{m}{N})p_m(1-p^-),$$

and, as before, the active agent pointing to the correct solution is denoted by *(a,m)* and an inactive agent by *(n,*)*.

It is apparent that the above stochastic process modelling an evolution of a given agent is a nonhomogenous Markov Chain. Nonhomogeneity of this stochastic process stems from the fact that the entries of the second row of the probability transition matrix P are not constant in time but change as search progresses thus making the matrix P_n time dependent. This is because the probability of an inactive agent becoming active in the next iteration depends on the normalised number of active agents in the previous iteration. This time dependence of entries of P reflects the interaction in the whole population of agents. However, the description of an agent’s evolution in terms of a nonhomogenous Markov Chain makes it possible to consider it, at any iteration, as effectively independent, so it decouples an agent from the population. This is because in a given iteration, due to the synchronicity of the search, once agents underwent testing and their activity status is established, they do not influence each other in this iteration (their evolution will affect elements of P in the next iteration only). In other words, during a single iteration the whole information about the rest of population is contained in the entries of the transition probability matrix which is identical for all the agents in the population. Thus one step evolution of agents, although governed by identical probability transition matrix is, conditionally on the past, statistically independent. This leads immediately to the conclusion that the evolution of the whole population of agents can be described as a nonhomogenous, discrete Markov Chain. Indeed, one can write a one step probability transition matrix for the process formulated in such a way. Assuming that, in iteration n, there are l active and N-l inactive agents, an entry of the one step transition probability matrix has the form:

$$P\left(X_{n+1} = m \,|\, X_n = l\right) = \sum_{k=0}^{\min(m,l)} \binom{l}{k} \left(1-p^-\right)^k \left(p^-\right)^{l-k} \binom{N-l}{m-k} \left(p_1^{\ n}\right)^{m-k} \left(1-p_1^{\ n}\right)^{N-l-(m-k)}$$

The above equation shows that indeed the Ehrenfest Urn model of the Stochastic Diffusion Search is equivalent to the Markov Chain model introduced in Chapter 3.

4.4. Long term behaviour of the Stochastic Diffusion Search.

In order to investigate the evolution of this process it is convenient to analyse the long term behaviour of a single agent. The following proposition will be proven:

Proposition 4.1 The sequence *$\{P_i\}$* of stochastic matrices describing the evolution of an agent in the Stochastic Diffusion Search is weakly ergodic (see Appendix 2 for a definition of a weak ergodicity).

Proof. The proposition will be first proven for $p^- > 0$. Recall that

$$P_n = \begin{array}{c} \\ (a,m) \\ (n,*) \end{array} \begin{array}{c} \begin{array}{cc} (a,m) & (n,*) \end{array} \\ \begin{bmatrix} 1-p^- & p^- \\ p_1^{\;n} & 1-p_1^{\;n} \end{bmatrix} \end{array}, \tag{4.1}$$

where

$$p_1^{\;n} = \tfrac{m}{N}(1-p^-) + (1-\tfrac{m}{N})p_m(1-p^-). \tag{4.2}$$

Define after (Seneta, 1973):

$$\begin{aligned} \lambda(P_n) &\overset{d}{=} \max_j \left\{ \min_i p_{ij} \right\} = \\ &= \max\left\{ \min\left\{1-p^-, p_1\right\}, \min\left\{p^-, 1-p_1\right\}\right\} \end{aligned}$$

Rearranging (4.2) leads to

$$p_1^{\;n} = (1-p^-)\big((1-p_m)x + p_m\big),$$

where x denotes the average activity in SDS. It is clear that $p_1^{\;n}$ is a linear function of x and as 0<x<1, it follows that

$$p_1^{\;n} \in \left[p_m\left(1-p^-\right), 1-p^-\right]$$

so

$$p_1{}^n \leq 1 - p^- \quad \Rightarrow \quad \min\left(p_1{}^n, 1 - p^-\right) = p_1{}^n$$
$$\Rightarrow \quad \min\left(p^-, 1 - p_1{}^n\right) = p^-$$

and therefore

$$\lambda(P_n) = \max\left(p_1{}^n, p^-\right) \cdot$$

Consider a series $\sum_{i=1}^{\infty} \lambda(P_n)$. This series is divergent because

$$(\forall n \geq 0)\left(\lambda(P_n) \geq p^-\right) \Rightarrow \sum \lambda(P_n) \geq \sum p^-$$

The last series diverges and therefore the weak ergodicity of $\{P_n\}$ follows from theorem 4.8 in (Seneta, 1973), (theorem 1, Appendix 2).

When $p^- = 0$ the same argument applies with the lower bound series taking the form $\sum_{i=1}^{\infty} p_m$. This is because from (4.2) it follows that p_m is a lower bound for $p_1{}^n$. □

Next, the following proposition will be proven:

Proposition 4.2 $\{P_n\}$ is asymptotically stationary (see Appendix 2).

Proof. The above assertion can be proven by formulating the problem in geometric terms and showing that appropriately defined mapping has a fixed point. Consider the subset K of a space R^6 defined as

$$K = X \times M_2^{p^-},$$
$$X = \left\{(p, 1-p) \,\middle|\, 0 \leq p \leq 1\right\},$$
$$M_2^{p^-} = \left\{ \begin{bmatrix} 1-p^- & p^- \\ p_{21} & 1-p_{21} \end{bmatrix} \middle| \begin{array}{l} p_{21} = p\left(1-p^-\right) + (1-p)p_m\left(1-p^-\right) \\ \quad = \left(1-p^-\right)(1-p_m)p + \left(1-p^-\right)p_m, \\ \quad 0 \leq p_m, p^- \leq 1, (p, 1-p) \in X \end{array} \right\}$$

X is a set of two dimensional probability vectors *(p, 1-p)* and $M_2^{p^-}$ is the set of two dimensional stochastic matrices with fixed first row and components of the second row being linear functions of *p*. All points of K can be attained by varying the parameter *p* so by definition K is a one dimensional subset of R^6. As K can be thought of as a Cartesian product of one dimensional closed intervals, it follows that K is convex and compact as a finite Cartesian product of convex, compact sets.

Define a norm in R^6

$$\| \; \| = \| \; \|_X + \| \; \|_M$$

where $\| \; \|_X$ and $\| \; \|_M$ are l_1 norm in R^2 and l_1 induced matrix norm in $M_2^{p^-}$ respectively.

Define the mapping

$$S\colon K \to K\,,$$
$$S(p, P) = \left(pP, P_{pP}\right),$$

where

$$p = (p, 1-p),$$
$$P = \begin{bmatrix} 1-p^- & p^- \\ p_{21} & 1-p_{21} \end{bmatrix},$$
$$P_{pP} = \begin{bmatrix} 1-p^- & p^- \\ p_{21}^* & 1-p_{21}^* \end{bmatrix},$$

and

$$pP = \left(p^*, 1-p^*\right) = \left(p\left(1-p^-\right) + (1-p)p_{21}, pp^- + (1-p)(1-p_{21})\right),$$
$$p_{21} = p\left(1-p^-\right) + (1-p)\left(1-p^-\right)p_m = (1-p_m)\left(1-p^-\right)p + p_m\left(1-p^-\right), \quad (4.3)$$

It follows that

$$\begin{aligned} p^* &= p\left(1-p^-\right)+(1-p)\left(\left(1-p_m\right)\left(1-p^-\right)p+p_m\left(1-p^-\right)\right)= \\ &= \left(1-p^-\right)\left[p+\left(1-p_m\right)(1-p)p+(1-p)p_m\right] \\ &= \left(1-p_m\right)\left(1-p^-\right)p(2-p)+p_m\left(1-p^-\right) \\ p_{21}^* &= \left(1-p_m\right)\left(1-p^-\right)p^*+p_m\left(1-p^-\right) \end{aligned} \tag{4.4}$$

S acts on both components of a point from K returning probability distribution and stochastic matrix obtained as a result of one step evolution of nonhomogenous Markov Chain corresponding to the one step evolution of an agent.

It is possible to prove that S is continuous in K. In order to show this, choose an $\varepsilon > 0$ and fix an arbitrary point in K, (q,Q) and chose another point (p,P), such that

$$\left\|(q,Q)-(p,P)\right\| \le \delta .$$

Note that

$$\begin{aligned} &\|q-p\|_X = |q-p|+|1-q-1+p| = \\ &= |q-p|+|-q+p| = 2|q-p| \end{aligned}$$

and that using (4.3)

$$\begin{aligned} &\|Q-P\|_M = \left\|\begin{bmatrix} 1-p^- & p^- \\ q_{21} & 1-q_{21} \end{bmatrix} - \begin{bmatrix} 1-p^- & p^- \\ p_{21} & 1-p_{21} \end{bmatrix}\right\| = \\ &= \left\|\begin{bmatrix} 0 & 0 \\ q_{21}-p_{21} & p_{21}-q_{21} \end{bmatrix}\right\| = \left|q_{21}-p_{21}\right| = \\ &= \left(1-p^-\right)\left(1-p_m\right)|q-p| = \tfrac{1}{2}\left(1-p^-\right)\left(1-p_m\right)\|q-p\|_X \end{aligned}$$

Together the above give

$$\left\|(q,Q)-(p,P)\right\| = \|q-p\|_X + \|Q-P\|_M = \left[1+\tfrac{1}{2}\left(1-p^-\right)\left(1-p_m\right)\right]\|q-p\|_X \ .$$

Now

$$\begin{aligned} &\left\|S(q,Q)-S(p,P)\right\| = \left\|\left(qQ,Q_{qQ}\right)-\left(pP,P_{pP}\right)\right\| = \left\|\left(qQ-pP,Q_{qQ}-P_{pP}\right)\right\| = \\ &= \|qQ-pP\|_X + \left\|Q_{qQ}-P_{pP}\right\|_M \end{aligned}$$

Both terms in the above equation will be considered separately.

$$\begin{aligned}
&\|qQ - pP\|_X = 2\left|q^* - p^*\right| = \\
&= 2(1 - p_m)(1 - p^-)\left|q(2-q) - p(2-p)\right| = 2(1 - p_m)(1 - p^-)\left|2(q-p) - (q^2 - p^2)\right| = \\
&= 2(1 - p_m)(1 - p^-)\left|2 - (q+p)\right|\left|q - p\right| = (1 - p_m)(1 - p^-)\left|2 - (q+p)\right|\|q - p\|_X
\end{aligned}$$

Similarly

$$\begin{aligned}
&\left\|Q_{qQ} - P_{pP}\right\|_M = \left|q^*_{21} - p^*_{21}\right| = \left|(1 - p_m)(1 - p^-)(q^* - p^*)\right| = \\
&= \tfrac{1}{2}(1 - p_m)^2(1 - p^-)^2\left|2 - (q+p)\right|\|q - p\|_X = (1 - p_m)(1 - p^-)\left|2 - (q+p)\right|\|Q - P\|_M
\end{aligned}$$

Finally it follows that

$$\begin{aligned}
&\|S(q,Q) - S(p,P)\| = (1 - p^-)(1 - p_m)\left|2 - (p+q)\right|\left[1 + \tfrac{1}{2}(1 - p^-)(1 - p_m)\right]\|q - p\|_X = \\
&(1 - p^-)(1 - p_m)\left|2 - (p+q)\right|\|(q,Q) - (p,P)\|
\end{aligned}$$

Continuity of the operator S follows from the fact that for $\delta = \frac{\varepsilon}{2(1-p^-)(1-p_m)}$ one obtains

$$\begin{aligned}
&\|S(q,Q) - S(p,P)\| = (1 - p^-)(1 - p_m)\left|2 - (p+q)\right|\|(q,Q) - (p,P)\| \le \\
&\le 2(1 - p^-)(1 - p_m)\|(q,Q) - (p,P)\| \le \varepsilon
\end{aligned}$$

Thus, by the Birkhoff-Kellogg-Schauder fixed point theorem, (Saaty, 1981), (see Appendix 2, theorem 3), it follows that S has a fixed point in K. This property implies that the sequence {P$_i$} of stochastic matrices is asymptotically stationary. □

Finally, it is possible to establish the desired result:

Proposition 4.3 The sequence *{P$_{i\,i}$}* is strongly ergodic (see Appendix 2).

Proof. Strong stationarity of {P$_i$} follows as a consequence of the above propositions, and theorem 4.11 in (Seneta, 1973), (see Appendix 2, theorem 2). □

Remarks.

1. The proof of Proposition 4.2 goes without change also in the case of p^{-}=0. Thus this model is more general than ergodic Markov chain model in that it allows to prove ergodicity of SDS in a more general case. This confirms the result of the homogenous Markov Chain model and partially invalidates results obtained in (Grech-Cini, 1995), in that it is not true in general that SDS would converge to a stable state only if p^{-}<0.5.

2. In spite of the above remark the value p^{-}=0.5 does play a role. The Birkhoff-Kellogg-Schauder fixed point theorem ensures only the existence of a fixed point but not its uniqueness. Thus Proposition 4.2 alone is not sufficient to establish strong ergodicity which requires uniqueness of the limit probability distribution vector. The latter follows from the weak ergodicity established in Proposition 4.1.

However, it is possible to obtain a stronger result using fixed point theory only, if one puts an additional condition on parameters of the search. Namely from the last inequalities of the proof of Proposition 4.2 it follows that it is possible to formulate a sufficient condition for S to be a contraction. Thus, it is sufficient to require that

$$2\left(1-p^{-}\right)\left(1-p_{m}\right)<1 \cdot$$

When this condition is fulfilled the uniqueness of the limiting probability distribution vector follows directly from the Banach fixed-point theorem. It follows, that if p_m is sufficiently small one obtains

$$\begin{aligned}&1-p_{m}\approx 1\Rightarrow 2\left(1-p^{-}\right)\leq 1\\&\Rightarrow p^{-}\geq \tfrac{1}{2}\end{aligned}$$

3. Actually the discrete dynamical process formulation used in the proof of Proposition 4.2 is an idealisation of the evolution of a single agent in SDS. In other words, one can interpret the evolution of SDS as a computer simulation of that process in which the real value of the parameter p is replaced by its estimate from a finite population sample.

4. It is important to stress the difference between the two models of SDS. The first model, presented in Chapter 3, included in its state space all possible configurations of active and inactive agents. Therefore the whole information about the environment influencing single

agent was contained in the model either as the parameters or in probabilities of transitions between states. On the other hand, the second model concentrated on a description of the behaviour of a single agent. The remaining agents were treated as an external environment influencing behaviour of a given agent. This influence was reflected in the nonhomogeneity of the probability transition matrix, which depended on the activity of the environment. This difference resembles the situation from control theory where one considers open and closed systems. A system is an open loop system, if the results of its action are fed back implicitly by appropriate reaction of the environment, which however is not considered to be a part of the system. Any open system can in principle be transformed into a larger, closed loop system, which includes a model for the environmental response. In this closed loop system the output of the system is explicitly fed back. The Ehrenfest urn model of SDS would correspond to an open loop system model and the first model to a closed loop model.

5. The nonhomogenous Ehrenfest urn model of SDS is a particular example of a discrete chain with complete connections in a wider sense, (Iosifescu, Theodorescu, 1970). In this model the transformation used to calculate the value of the parameter in the next step is in part determined by the very matrix of transition probabilities which depends on this parameter. Thus the transition matrices and the parameters are coupled.

6. In the matrix P describing the probability transition of a single agent only second row depends on the parameter p and thus is changing over time; the first row remains unchanged. This corresponds to the fact that, conditional on an agent being active in a given iteration, its evolution is independent of the state of other agents in the population. However, this assumption can be relaxed, in effect leading to the dependence of the first row of matrix P on the parameter p. It will be shown in Chapter 8 that this modification will bring a desirable, in some situations, effect of more even distribution of resources.

4.5. *Calculation of the equilibrium probability distribution of SDS.*

The previous section established that the evolution of a single agent in the Stochastic Diffusion Search is asymptotically ergodic, which effectively makes it possible to characterise its long term behaviour. In other words, a stochastic process describing a time evolution of a single agent will visit states of its state space with frequencies, which can be characterised via the limiting probability distribution. This characterisation will be consequently used in constructing the steady state probability distribution of the whole ensemble of agents used in the search. The resource allocation management by the Stochastic Diffusion Search will be inferred from the ensemble steady state distribution and characterised in terms of the parameters describing the search space and the best fit solution.

First it is necessary to find the limit probability distribution implied by the strong ergodicity of a nonhomogenous Markov Chain describing evolution of a single agent. To this end one has to find the solution of the equation

$$S(p,P) = (p,P)$$

This amounts to solving a system of two equations

$$\begin{cases} \pi P = \pi, \\ p_1 = \left(1 - p^-\right)\pi_1 + \left(1 - p^-\right)\left(1 - \pi_1\right)p_m \end{cases} \tag{4.5}$$

The first equation is an equation for an eigenvalue 1 of a two by two stochastic matrix P for which the eigenvector has the form

$$\pi = \left(\pi_1, 1 - \pi_1\right) = \left(\frac{p_1}{p_1 + p^-}, \frac{p^-}{p_1 + p^-}\right) \tag{4.6}$$

This makes it possible to find the solution in the special case when p_m=0, which implies that there is no solution in the search space. It follows, that the initial distribution of agents is concentrated entirely on the inactive state and from the latter it follows that p_1=0 so, as expected,

$$\pi = (0, 1)$$

i.e. an agent will always remain inactive.

To find the solution in the general case, assume that p_m>0. From equation (4.6) and the second equation of (4.5) it follows that

$$\pi_1\left[\left(1 - p^-\right)\pi_1 + \left(1 - p^-\right)\left(1 - \pi_1\right)p_m + p^-\right] = \left(1 - p^-\right)\pi_1 + \left(1 - p^-\right)\left(1 - \pi_1\right)p_m$$

which after rearrangement leads to a quadratic equation in π_1

$$\left(1 - p^-\right)\left(1 - p_m\right)\pi_1^2 + \left[2\left(1 - p^-\right)p_m + 2p^- - 1\right]\pi_1 - \left(1 - p^-\right)p_m = 0$$

This can be written in the form

$$\left(1-p^{-}\right)\left(1-p_{m}\right)\pi_{1}^{2}-\left[2\left(1-p^{-}\right)\left(1-p_{m}\right)-1\right]\pi_{1}-\left(1-p^{-}\right)p_{m}=0$$

This equation has two solutions because the condition

$$\left[2\left(1-p^{-}\right)\left(1-p_{m}\right)-1\right]^{2}+4\left(1-p^{-}\right)^{2}\left(1-p_{m}\right)p_{m}\geq 0$$

is always fulfilled. This solutions are as follows

$$\pi_{i}=\frac{2\left(1-p^{-}\right)\left(1-p_{m}\right)-1\pm\sqrt{\left[2\left(1-p^{-}\right)\left(1-p_{m}\right)-1\right]^{2}+4\left(1-p^{-}\right)^{2}\left(1-p_{m}\right)p_{m}}}{2\left(1-p^{-}\right)\left(1-p_{m}\right)},\qquad i=1,2$$

Straightforward analysis of the above solutions implies that only one of them can be regarded as a solution to the problem. Namely, the desired equilibrium probability distribution is

$$\begin{aligned}\pi=(\pi_{1},\pi_{2})=&\\ &\left(\frac{2\left(1-p^{-}\right)\left(1-p_{m}\right)-1+\sqrt{\left[2\left(1-p^{-}\right)\left(1-p_{m}\right)-1\right]^{2}+4\left(1-p^{-}\right)^{2}\left(1-p_{m}\right)p_{m}}}{2\left(1-p^{-}\right)\left(1-p_{m}\right)},\right.\\ &\left.\frac{1-\sqrt{\left[2\left(1-p^{-}\right)\left(1-p_{m}\right)-1\right]^{2}+4\left(1-p^{-}\right)^{2}\left(1-p_{m}\right)p_{m}}}{2\left(1-p^{-}\right)\left(1-p_{m}\right)}\right)\end{aligned}\tag{4.7}$$

Modelling SDS via evolution of single agents, as proposed in the model, implies the distinguishability of agents. From this it follows that the state space of the ensemble of Markov Chains corresponding to N agents consists of N-tuples x = (x_1 , ..., x_N) where the ith component takes on a value 1 or 0 depending on whether the ith agent was active or not. Because of strong ergodicity of Markov Chains describing evolution of single agents and their conditional independence, discussed at the end of Section 4.3 in conjunction with showing equivalence between the Ehrenfest Urn model and the Markov Chain model of Chapter 3, there will be a unique asymptotic limit probability distribution given by (Whittle, 1986):

$$\Pi(x)=\pi_{1}^{a[x]}\pi_{2}^{N-a[x]},\tag{4.8}$$

where a[x] denotes number of active agents corresponding to the state x and (π_1,π_2) is the two state asymptotic distribution corresponding to the strong ergodicity of a Markov Chain describing a single agent.

From Propositions 4.1-4.3 it follows that the Markov Chain corresponding to the evolution of a single agent is asymptotically homogenous, i.e. the limit $\lim_{i \to \infty} P_i$ exists. Denote this limit by P. Thus this process behaves more and more like a homogenous Markov Chain as it approaches the equilibrium. Therefore, as of main interest in this study is the steady state behaviour of SDS, one can, with good approximation, characterise the equilibrium probability distribution of SDS via analysis of a time homogenous Markov Chain with probability transition matrix P. Therefore in order to establish the equilibrium probability distribution of SDS one has to abandon the distinguishability of agents implied by the construction of the generalised nonhomogenous Ehrenfest urn model. This means that it is necessary to consider an aggregate process X_n=a[x]$_n$, in which all configurations corresponding to the same number of active agents are lumped together. This aggregate process is reversible, as it is derived from a reversible Markov Chain [every time homogenous, two state Markov Chain is reversible, (Whittle, 1986)]. It is also a Markov Chain because the aggregation procedure corresponds to a maximal invariant under permutations of agents, which preserve the statistics of the process, (Whittle, 1986, theorem 3.7.2). This can be seen from the fact, that permuting two arbitrary agents in the same state (both active or inactive) does not affect the probability distribution of the process and the lumping procedure described above establishes equivalence classes of states which have the same probability. Therefore summing equation (4.8) over all configurations x corresponding to the same number of active agents a[x] one obtains a probability distribution of SDS:

$$\pi(n) = \binom{N}{n} \pi_1^n \pi_2^{N-n} \tag{4.9}$$

which is a binomial distribution.

Equation (4.9) describes the steady state probability distribution of the whole ensemble of agents used in the Stochastic Diffusion Search. It describes probabilities of finding agents in different configurations of states, which implies different possible distributions of resources by SDS. Thus the resource allocation of SDS in the steady state can be characterised via the expected distribution of agents. The deviations from the latter by the former can be characterised in terms of the standard deviation of the ensemble probability distribution.

From equation (4.9) it follows immediately that the expected number of active agents in the equilibrium will be

$$E[n] = N\pi_1 \tag{4.10}$$

In fact the most likely state, n, given by the maximum of the binomial distribution will be an integer number fulfilling following inequalities, (Johnson, Kotz, 1969),

$$(N+1)\pi_1 - 1 \leq n \leq (N+1)\pi_1$$

This implies that for sufficiently large N the expected number of active agents is a good characterisation of their actual most likely number.

Similarly the standard deviation defined for binomial distribution as

$$\sigma = \sqrt{N\pi_1\pi_2} \tag{4.11}$$

will be used as a measure of variability of the number of active agents around its steady state.

In fact E[n] is not sensu stricto an equilibrium of the system. From strong ergodicity it follows that eventually all possible configurations of agents will be encountered provided that one would wait sufficiently long. However, as in the case of the system with two containers with gas, although all states are possible, nevertheless some of them will appear extremely rarely (e.g. state in which all the particles of the gas would concentrate in one container only). In fact, the system will spend most of the time fluctuating around the expected state, which thus can be considered as a quasi-equilibrium.

The above discussion closely parallels the reasoning motivating Ehrenfests in using their model for a discussion of an apparent contradiction between the reversible laws of micro-dynamics of single particles and irreversibility of thermodynamical quantities.

The next section will illustrate some of these points with a numerical example.

4.6. Numerical example.

Consider the Stochastic Diffusion Search with N=1000 agents, a probability of hitting the best instantiation of the target p_m=0.001 and a probability of a false negative p^-=0.2.

Equations (4.7) - (4.10) imply that $\pi_1 \approx 0.75, \pi_2 \approx 0.25, E[n] \approx 750, \sigma \approx 13.7$. Therefore the fluctuations are well concentrated around the mean value with the standard deviations not exceeding 1.5% of the total number of agents.

It follows from the Markov Chain theory, (Iosifescu, 1980), that, for the ergodic Markov Chain, the mean return time to a state j is inversely proportional to the equilibrium probability of this state, i.e.

$$m_j = \frac{1}{\pi(j)}$$

In the example considered, the mean return time to the state in which all agents are inactive is then

$$m_0 = \frac{1}{\pi(0)} \approx \frac{1}{(0.25)^{1000}} \propto 10^{602}$$

One can see that indeed this state is on average visited extremely rarely.

Consider the behaviour of the Stochastic Diffusion Search with the same parameters characterising the search space but with smaller number of agents.

Let N=10, then $E[n] \approx 7.5$ and $\sigma \approx 1.7$ so in this case the variability of the number of active agents around the quasi equilibrium raises to about 17%. Also the mean return time to the state with all agents inactive decreased in this case

$$m_0 = \frac{1}{\pi(0)} = \frac{1}{(0.25)^{10}} \propto 10^6.$$

In general, if one denotes

$$\tau = \log m_0$$

then varying only the number of agents leads to

$$r = \frac{\tau_0^1}{\tau_0^2} = \frac{\log \pi^1(0)}{\log \pi^2(0)} = \frac{N_1 \log \pi_2}{N_2 \log \pi_2} = \frac{N_1}{N_2}$$

so

$$N_1 = rN_2 .$$

Thus the ratio of mean return times to the ‘all agents inactive’ state is equal to the ratio of the numbers of agents involved in the search. Thus in the above example r=100.

For comparison assume that one performs N completely random draws of potential positions from the search space and, upon all draws, checks a random microfeature at that position. The probability of recognising the best solution k times in such a procedure is given by

$$p[X = k] = \binom{N}{k}\left[p_m\left(1 - p^-\right)\right]^k\left[1 - p_m\left(1 - p^-\right)\right]^{N-k},$$

where the standard notation from the model is used.

For N=1000, p_m=0.001 and p^-=0.2, one has $p[X = 0] \approx 0.449$, $p[X = 750] \propto 10^{-2081}$ and for N=10 $p[X = 0] = 0.992$, $p[X = 7] \propto 10^{-20}$. Thus, in both cases, in completely random draws the probability of being completely unsuccessful is orders of magnitude higher than in the Stochastic Diffusion Search. This indicates that the diffusion of information in SDS changes radically the way the algorithm performs the search. The probabilities of the random draws procedure to be completely unsuccessful are very large whereas for SDS they are extremely small. On the contrary, the probability of obtaining a group of successful draws of comparable size to the largest cluster of active agents in SDS is very small indicating that the solution proposed by the Stochastic Diffusion Search is highly reliable.

4.7. *Characterisation of the resource allocation as a function of fit of the global solution.*

The foregoing sections discussed how to derive the agents distribution by SDS and its variability from the Ehrenfest Urn model. They characterised the resource allocation in terms of the expected size of the active agents cluster and its standard deviation. However, the latter quantities depend on the parameters characterising the search space.

This section will characterise the behaviour of the average number of active agents as a function of the parameter p^-, characterising the possibility of false negative response of agents for the best instantiation of the target. Figure 4.1 illustrates this relationship for the value of p_m=0.001 and N=1000 agents and Figure 4.2 shows the two dimensional plot of the average number of active agents as a function of both parameters p_m and p^-.

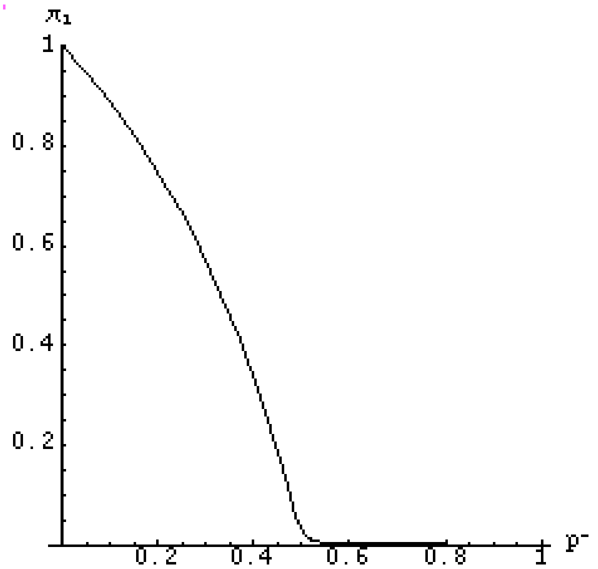


Figure 4.1 The normalised average number of active agents in SDS as a function of the parameter p^-. Plot obtained for N=1000 and p_m=0.001.

Figure 4.1 implies that the number of active agents decreases nonlinearly with an increase of the false negative parameter p^- and reaches very small values around p^-=0.5. Thus, two regions of different characteristics of the resource allocation by SDS can be inferred from Figure 4.1. The first one is for p^-<0.5, where the cluster of active agents constitutes a significant part of the total amount of resources, and the second one is for p^->0.5, where the amount of active agents is orders of magnitude smaller.

From the fact that the number of agents in SDS is always finite it follows that for a given total number of agents there exists such value of p^- that, above it, the actual number of active agents is almost always zero. This seems to confirm an estimate obtained in (Grech-Cini, 1995). However, π_1 as a function of p^- is everywhere positive in [0,1). It follows that for any p^->0 there exists a finite number of agents N such, that $\lfloor N\pi_1 \rfloor > 0$, where $\lfloor x \rfloor$ denotes the greatest integer smaller than x.

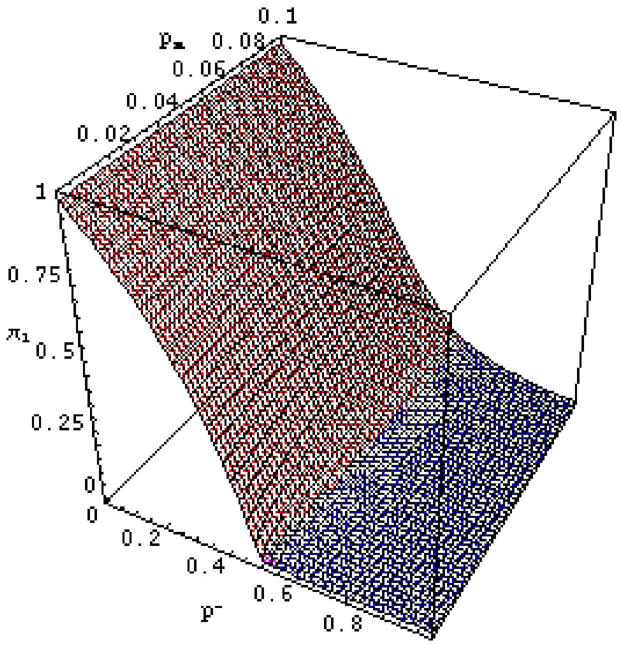


Figure 4.2 The normalised mean number of active agents as a function of both the false negative parameter p^- and the probability of hit at the best instantiation p_m; plotted for N=1000.

Figure 4.2 shows the normalised mean number of active agents as a function of both the false negative parameter p^- and the probability of locating the best instantiation p_m. From the

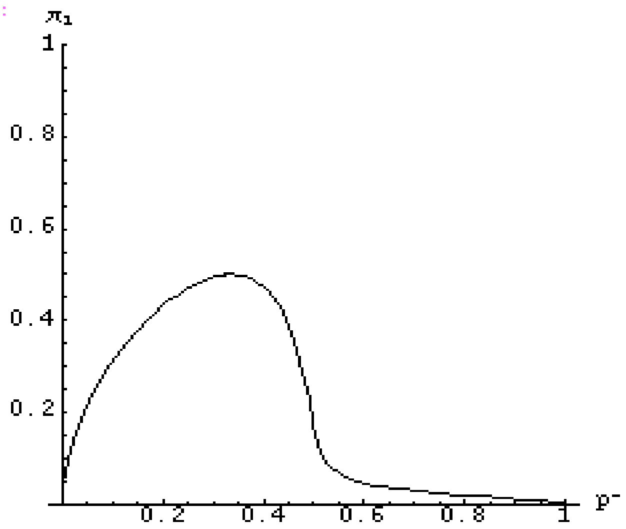


Figure 4.3 The rescaled standard deviation of the number of active agents calculated from the model for N=1000 agents and p_m=0.001; the scaling factor is $\alpha = N^{\frac{-1}{2}} \approx 0.0316$.

inspection of this figure it follows that changing p_m does not alter significantly the dependence of the mean number of active agents on the parameter p^-. The change, resulting from an increase of p_m, can be summarised as a smoothing out the boundary between two regions of behaviour of SDS clearly visible in Figure 4.1.

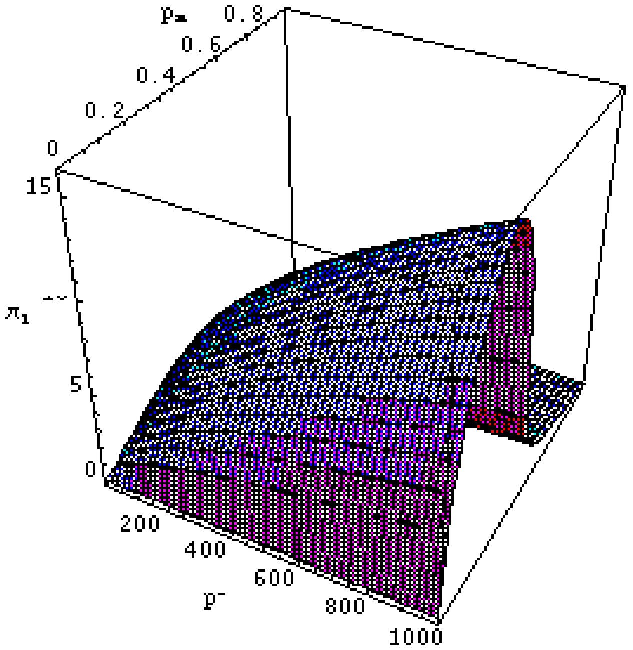


Figure 4.4 The standard deviation of the number of active agents as a function of the total number of agents N in SDS and of false negative probability p^-; plotted for p_m=0.001.

Similarly, it is possible to investigate, using the equations (4.7) and (4.11), dependence of the standard deviation on parameters of SDS. Figure 4.3 illustrates the behaviour of the standard deviation as a function of p^-, for p_m=0.001 and N=1000 agents and Figure 4.4 shows the 3D plot of the standard deviation as a function of p^- and N.

From Figure 4.3 one can deduce that standard deviation is also a non-linear function of p^- first increasing quickly with p^- increasing from 0 to around 0.4, where the normalised standard deviation is largest, and then rapidly decreasing for p^- increasing from 0.4 to 0.6. When p^- increases further from 0.6 to 1 the normalised standard deviation decreases more steadily to 0. Figure 4.3 corresponds in fact to a cross-section of Figure 4.4 along the line of a fixed number of agents.

4.8. Simulations.

A number of simulations of the Stochastic Diffusion Search were run in order to compare the theoretical estimates characterising quasi equilibrium with the actual behaviour of the system. The simulations reported here were run with N=1000 agents, p_m=0.001 and p^- assuming values of 0.1, 0.2, 0.5 and 0.7 respectively. For calculating the estimates of the expected number of active agents and its standard deviation SDS was run over 2000 iterations. In all cases the first 500 samples were discarded as a burn-in period. This method was suggested in (Gilks et al, 1996) in order to avoid the bias caused by taking into account samples from the evolution when the process is far from the steady state.

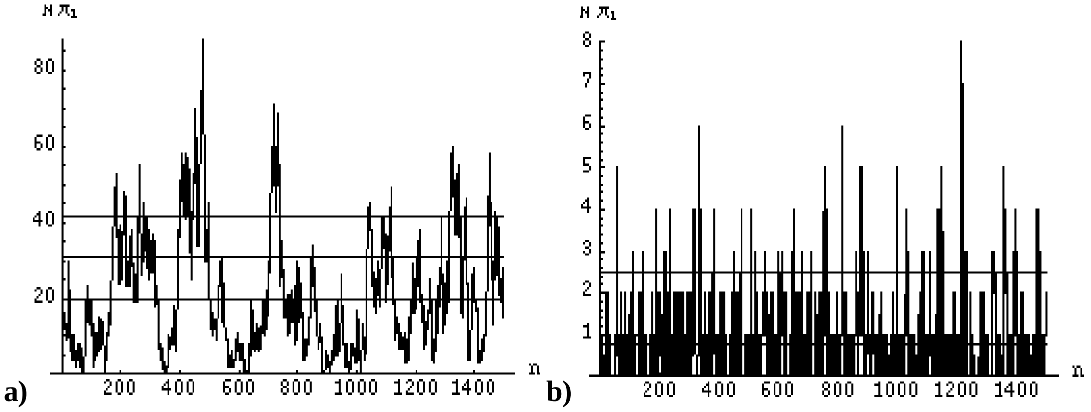


Figure 4.5 Evolution of the number of active agent in SDS with N=1000 agents and p_m=0.001. The false negative parameter p^- is 0.5 (left panel) and 0.7 (right panel). The straight lines correspond to the average activity predicted by the model surrounded by the +/-2 standard deviations band.

Average activity	Standard deviation
889.02	10.06
749.61	15.22
20.57	15.48
0.81	1.09

Table 4.1 Average activities and standard deviations estimated from the 1500 samples of SDS activity. N=1000, p_m=0.001 and p^- changes from 0.1, 0.2, 0.5 to 0.7 respectively (top to bottom).

However, the size of the burn-in period is, in general, a difficult issue because it is related to the estimation of the convergence of a given Markov Chain to its steady state probability distribution. In the Markov Chain Monte Carlo practice, the number of iterations needed to obtain reliable estimates of statistics is often of the order of tens of thousands and the burn-in periods lengths can also be considerably long (Gilks et al, 1996). The particular values of the burn-in periods, used in this thesis, were chosen on the basis of visual inspection, as a heuristic remedy against the bias of estimates.

This procedure resulted in a better agreement between predicted and estimated variances in half of the cases reported here. The results are summarised in Table 4.1 and Table 4.2 and in Figure 4.6 and Figure 4.5.

Average activity	Standard deviation
888.06	9.97
750.1	13.69
30.65	5.45
0.75	0.86

Table 4.2 Average activities and standard deviations of SDS predicted by the model. N=1000, p_m=0.001 and p^- changes from 0.1, 0.2, 0.5 to 0.7 respectively (top to bottom).

One can conclude that the expected number of active agents and two standard deviations, as calculated from the model, correspond to the empirical parameters a, b used in (Bishop, 1989)

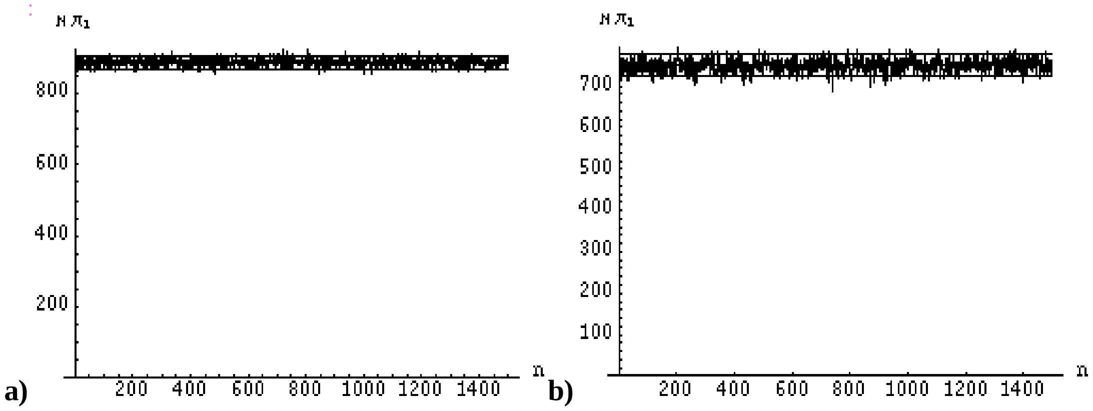


Figure 4.6 Evolution of the number of active agents in SDS with N=1000 agents and p_m=0.001.
(a) The false negative parameter p^- is 0.1; (b) p^-=0.2. The straight lines correspond to the average activity predicted by the model, surrounded by the +/-2 standard deviations band.

to define statistical stability of SDS and thus defining the halting criterion. Therefore the Ehrenfest Urn model provides a firm ground for the calculation of these parameters and for the characterisation of their dependence on the parameters of the Stochastic Diffusion Search and the search space.

4.9. *Conclusions.*

In this chapter a new model of the Stochastic Diffusion Search was introduced, which effectively was a nonhomogenous generalisation of the Ehrenfest Urn model. This model made it possible to prove ergodicity of SDS independently of the first Markov Chain model.

The resource allocation management of SDS was characterised by finding the steady state probability distribution of the Ehrenfest Urn model of SDS and observing that the actual distribution of agents is well described by two first moments of this distribution.

It thus appeared that the resource allocation by SDS corresponds to its quasi-equilibrium state characterised by the state, in which the Stochastic Diffusion Search would spend most of the time and the stability region - a range of fluctuations of the actual number of active agents. Thus the quasi equilibrium state is well characterised by the expected number of active agents and the stability region, in which SDS will fluctuate - by two standard deviation bands around expected number of active agents. Finding explicit expressions for these quantities made it possible to characterise their dependence on the parameters of the search - a total number of agents, probability of false negative and a probability of randomly locating the best instantiation in a random draw. It follows, that the analysis outlined in this chapter described the dependance of the resource allocation on the parameters describing the search conditions. This analysis also provides a theoretical background for the halting criterion introduced in (Bishop, 1989).

The model discussed in this chapter allowed for a slightly more general analysis than the first model based on a homogenous Markov Chain due to relaxation of the assumption about a presence of the ideal instantiation of the target in the search space. However, it still retained the assumption of the noiseless search space. The consequtive chapters will present a generalisation of the model, allowing first for existence of a single distractor and then discussing further generalisation to the arbitrary number of distractors.

CHAPTER 5

EXTENDED EHRENFEST URN MODEL OF SDS

5.1. *Introduction.*

Chapter 4 described an analysis of the Stochastic Diffusion Search in the case of a noise-free search space. This chapter will extend the Ehrenfest urn model to incorporate the influence of noise on SDS. It will present a simple extension, in which in the search space there exist the best solution and a single distractor with a strictly smaller overlap with the target than that of the solution. From this perspective the results presented here will constitute a starting point for the characterisation of SDS behaviour in the presence of noise. In particular it will appear that SDS is very sensitive to the difference between the best fit pattern and a distractor and allocation of resources is very strongly biased towards the former.

However, the model will cover also the case, when both the “best” fit pattern and the “distractor” have the same overlap with the target. Thus, it will appear that the distinction between the best fit and a distractor may be redundant and the results of this chapter could be interpreted in a different manner. On this interpretation, one would be interested in analysing how SDS distributes agents among several possible solutions which are not necessarily equally similar to the template. It will follow that, although in theory SDS would form clusters corresponding to all sub-optimal solutions, the very strong bias towards the best fit solution will nevertheless result in an emergence of serial processing by the algorithm.

As it was the case in the proceeding chapter the resource allocation of the Stochastic Diffusion Search will follow, irrespective of possible interpretation of results, from steady state probability distribution, its statistics and their dependence on the parameters of the model.

The next section will discuss the extension of the Ehrenfest Urn model to the two patterns case and the consecutive section will present a detailed discussion of dependence of its solutions on the parameters characterising SDS. This will enable one to discuss, in section 5.4, the steady state probability distribution, leading to a characterisation of the SDS’s distribution of resources. The subsequent two sections will respectively consider several features of the Stochastic Diffusion Search illustrated by appropriate simulations and discuss the results of this chapter.

The characteristics of SDS described here will enable further extension, presented in the next chapter, allowing for its steady state behaviour analysis in the case of many distractors present in the search space.

Although it will be, in principle, possible to solve the extended model, however, the exact solutions are very complex and do not reveal the dependence of the equilibrium behaviour of SDS on parameters of the search. For this reason it will be more informative to find an approximate solution, which will make it possible to investigate how the parameters affect the steady state and will also provide a quantitative estimation of the true steady state behaviour. The approximate solution for the extended model will be a subject of Chapter 7.

5.2. Exact extended Ehrenfest urn model.

In order to define an extended model it is necessary to introduce a notation for parameters characterising the distractor. Thus, denote the probability of random location of the distractor by p_d and the probability of the agent undergoing a successful test of the microfeature while pointing to this distractor as p^+. A single agent can be in this new setting attracted to either the best solution or to the distractor thus contributing to the two alternative clusters of agents. Overall, the evolution of a single agent will be described by a probability transition matrix, which is a straightforward extension of the corresponding matrix from Chapter 4 and has the following form

$$P = \begin{array}{c} \\ (a_m, m) \\ (a_d, m) \\ (n, *) \end{array} \begin{array}{c} \begin{array}{ccc} (a_m, m) & (a_d, m) & (n, *) \end{array} \\ \begin{bmatrix} 1 - p^- & 0 & p^- \\ 0 & p^+ & 1 - p^+ \\ p_1 & p_2 & 1 - p_1 - p_2 \end{bmatrix} \end{array} \tag{5.1}$$

where

$$\begin{aligned} p_1 &= \tfrac{m_1}{N}\left(1 - p^-\right) + \left(1 - \left(\tfrac{m_1}{N} + \tfrac{m_2}{N}\right)\right) p_m \left(1 - p^-\right), \\ p_2 &= \tfrac{m_2}{N} p^+ + \left(1 - \left(\tfrac{m_1}{N} + \tfrac{m_2}{N}\right)\right) p_d \, p^+, \end{aligned} \tag{5.2}$$

and m_1, m_2 denote, respectively, populations of agents concentrated on the best solution and on the distractor.

One can see that in fact the only extension to the model of Chapter 4 is an introduction of terms corresponding to the second cluster of agents.

The theorems from Chapter 4 and their proofs hold also in this more general case with minor adaptations. Therefore one is again looking for a steady state probability distribution which is a solution to the equation

$$\begin{aligned} \pi P &= \pi, \\ \pi &= (\pi_1, \pi_2, \pi_3) \end{aligned} \tag{5.3}$$

The latter is equivalent to the system of equations

$$\begin{cases} \pi_1(1-p^-) + \pi_3 p_1 = \pi_1 \\ \pi_2 p^+ + \pi_3 p_2 = \pi_2 \\ \pi_1 p^- + \pi_2(1-p^+) + \pi_3(1-p_1-p_2) = \pi_3 \end{cases} \tag{5.4}$$

where

$$\sum \pi_i = 1, \pi_i \geq 0, \quad i = 1, 2, 3.$$

From the equation of (5.4) one obtains

$$\begin{aligned} \pi_1 &= \frac{p_1}{p^-}\pi_3, \\ \pi_2 &= \frac{p_2}{1-p^+}\pi_3, \\ \pi_1 + \pi_2 + \pi_3 &= 1 \end{aligned}$$

which leads to the following system of five equations with four unknowns

$$\begin{aligned} \pi_1 &= \frac{p_1(1-p^+)}{p_1(1-p^+) + p_2 p^- + p^-(1-p^+)}, \\ \pi_2 &= \frac{p_2 p^-}{p_1(1-p^+) + p_2 p^- + p^-(1-p^+)}, \\ \pi_3 &= \frac{p^-(1-p^+)}{p_1(1-p^+) + p_2 p^- + p^-(1-p^+)}, \\ p_1 &= \left[\pi_1 + \{1-(\pi_1+\pi_2)\} p_m\right](1-p^-), \\ p_2 &= \left[\pi_2 + \{1-(\pi_1+\pi_2)\} p_d\right] p^+ \end{aligned} \tag{5.5}$$

From the above system one can make the following observation. From the first equation of (5.5) it follows that probability of finding, in an equilibrium, a single agent locked to the target is affected by the equilibrium probability of agent’s transition from inactive state to “the active pointing towards the target” state and by the probability of failure in the testing phase of the agent in “the active pointing towards the distractor” state. There is an analogous dependence in the case of the second equation of (5.5) expressing the equilibrium probability of finding an agent locked to the distractor. In both cases the dependence of the equilibrium probabilities on the corresponding parameters, i.e. π_1 on 1-p^- and π_2 on p^+, is expressed via corresponding probabilities of flow from inactive state into the active states. From the above it follows that in equilibrium both clusters of active agents will influence each other, so effectively it is possible to compare their behaviour to a system of two connected containers with fluid - a change in the parameters characterising one of the containers will affect the amount of liquid in the other and vice versa.

5.3. Characterisation of the solutions of the extended Ehrenfest Urn model.

Although it is possible, in principle, to reduce the system (5.5) to cubic equations in single variables π_1 and π_2, the solutions are complicated and do not reveal the nature of dependence of SDS behaviour on the parameters. Therefore instead of using this method of analysis, it will be more informative to concentrate on the mapping describing a one step evolution of the Stochastic Diffusion Search. The equation (5.3) leading to the above system (5.5) is in fact a fixed point equation of this mapping. Define a mapping

$$\begin{aligned} &S: X \to X, \\ &S(\pi) = \pi P \end{aligned} \tag{5.6}$$

where

$$\pi \in X = \left\{ (\pi_1, \pi_2, \pi_3) \middle| \sum \pi_i = 1, \forall i \;\; 0 \le \pi_i \le 1 \right\} \subset R^3$$

and P was defined in (5.1)-(5.2).

From the above it follows that

$$\begin{aligned} \pi_{n+1} &= S(\pi_n) = \pi_n \mathbf{P} = \left(\pi_1(1-p^-) + \pi_3 p_1, \pi_2 p^+ + \pi_3 p_2, \dots\right) = \\ &= \left(\pi_1(1-p^-) + \pi_3(\pi_1 + \pi_3 p_m)(1-p^-), \pi_2 p^+ + \pi_3(\pi_2 + \pi_3 p_d)p^+, \dots\right) = \\ &= \left((1-p^-)(\pi_1 + \pi_1\pi_3 + \pi^2{}_3 p_m), p^+(\pi_2 + \pi_2\pi_3 + \pi^2{}_3 p_d), \dots\right) \end{aligned}$$

where the obvious expression for the third co-ordinate of the new probability vector was omitted.

Using the fact that $\pi_3 = 1-\pi_1-\pi_2$, after straightforward algebraic operations one can write

$$S(\pi_1,\pi_2) = \left(f_1(\pi_1,\pi_2),\, f_2(\pi_1,\pi_2),\, f_3(\pi_1,\pi_2)\right), \tag{5.7}$$

where

$$\begin{aligned}
f_1(\pi_1,\pi_2) &= (1-p^-)\big[-(1-p_m)\pi_1^2 + p_m\pi_2^2 + (2p_m-1)\pi_1\pi_2 - 2p_m\pi_2 + \\
&\quad + 2(1-p_m)\pi_1 + p_m\big],\\
f_2(\pi_1,\pi_2) &= p^+\big[p_d\pi_1^2 - (1-p_d)\pi_2^2 + (2p_d-1)\pi_1\pi_2 + \\
&\quad + 2(1-p_d)\pi_2 - 2p_d\pi_1 + p_d\big],\\
f_3(\pi_1,\pi_2) &= 1 - f_1(\pi_1,\pi_2) - f_2(\pi_1,\pi_2)
\end{aligned} \tag{5.8}$$

In the above derivations the definition of S has been changed slightly, taking into account that the space of all three dimensional discrete probability distributions is a subset of a two dimensional plane in R^3. This reformulation will make it possible to get insight into the evolution of the Stochastic Diffusion Search. In fact, one can plot the mapping (5.7) for different values of parameters characterising SDS.

Figure 5.1 shows the set of all three dimensional discrete probability distribution vectors X and its image under the mapping S corresponding to the best solution with p^-=0.2 and a distractor with p^+=0.1, p_m=p_d=0.0001.

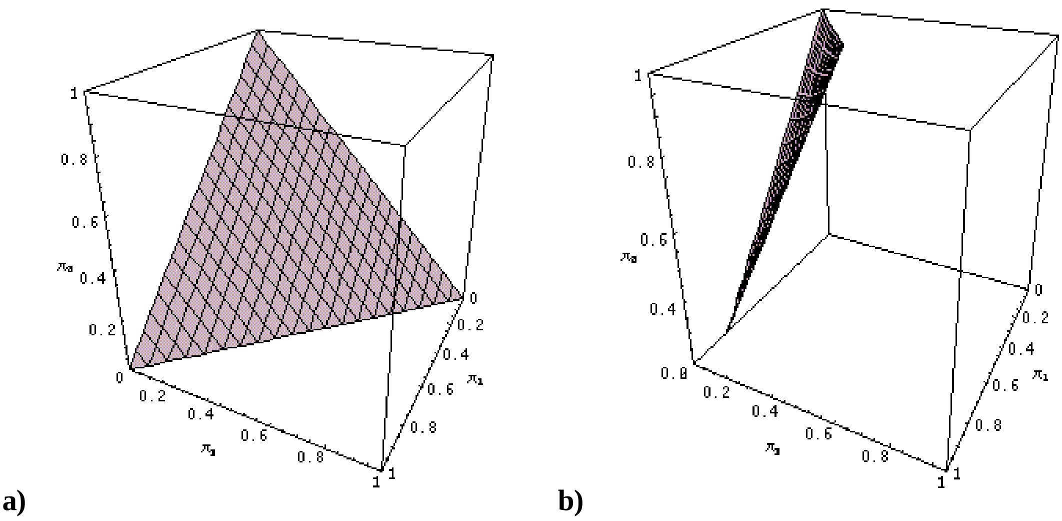
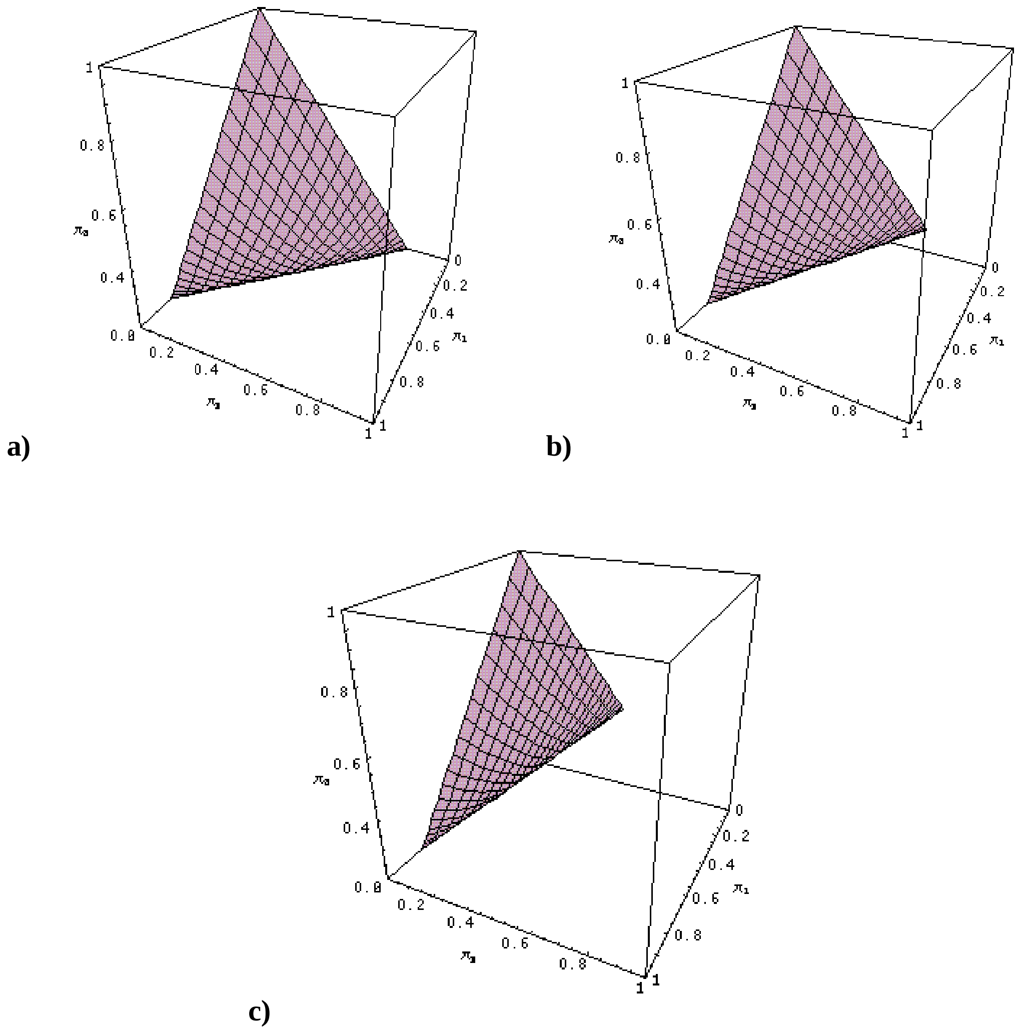


Figure 5.1 The set of all three dimensional discrete probability distribution vectors (a) and its image under the mapping S with parameters p^-=0.2 and a distractor with p^+=0.1 (b); p_m=p_d=0.0001.

From (5.8) it follows that for $p_m=p_d$

$$f_1(\pi_1,\pi_2)\frac{1}{1-p^-} = f_2(\pi_2,\pi_1)\frac{1}{p^+}$$
$$f_2(\pi_1,\pi_2)\frac{1}{p^+} = f_1(\pi_2,\pi_1)\frac{1}{1-p^-}$$

and so



a)

b)

c)

Figure 5.2 Images of X under the mapping S corresponding to one step evolution of the Stochastic Diffusion Search for $p_m=p_d=0.0001$ and (a) $p^-=0.2$, $p^+=0.8$, (b) $p^-=0.2$, $p^+=0.7$, (c) $p^-=0.2$, $p^+=0.5$.

$$
\begin{aligned}
f_1(\pi_1,\pi_2) &= \frac{1-p^-}{p^+} f_2(\pi_2,\pi_1) \\
f_2(\pi_1,\pi_2) &= \frac{p^+}{1-p^-} f_1(\pi_2,\pi_1)
\end{aligned}
\tag{5.9}
$$

It follows from (5.9) that knowing the evolution of the iterations of S starting from a given initial probability distribution vector, it is possible to retrieve evolution of S for the probability vector which is symmetric with respect to the plane x=y with respect to the original one. In other words, orbits of S are symmetrical subject to a component-wise rescaling. In particular, in the case of $1-p^-=p^+$ orbits of S, corresponding to the symmetrical initial probability distribution vectors, are symmetrical. These properties can be observed for an initial step of the evolution of S in Figures (5.2) and (5.3) showing images of X under the mapping S for various values of parameters p^- and p^+.

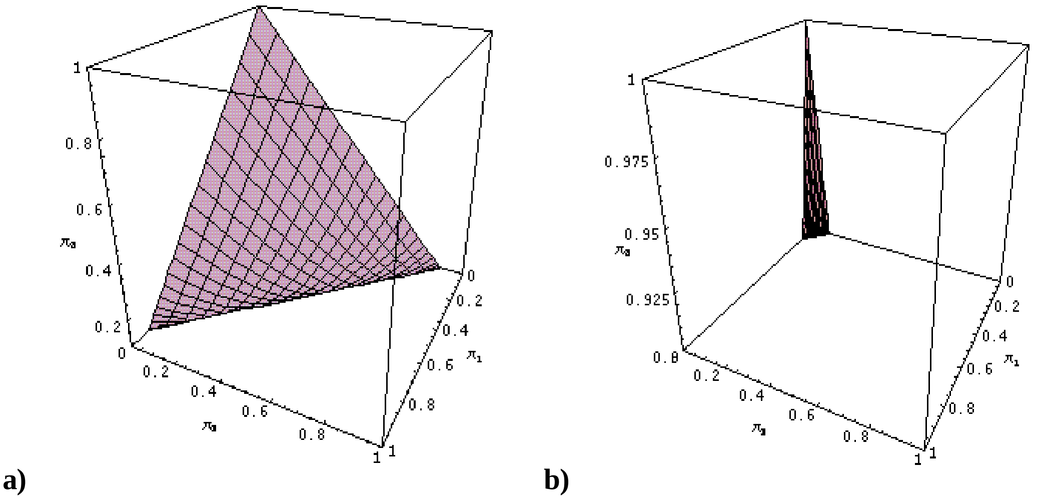


Figure 5.3 Images of X under the mapping S corresponding to one step evolution of the Stochastic Diffusion Search for $p_m=p_d=0.0001$ and (a) $p^-=0.1$, $p^+=0.9$, (b) $p^-=0.9$, $p^+=0.1$.

From Figure 5.1 - Figure 5.3 it follows that the mapping S contracts the domain X to a set of smaller area. This is to be expected because of the contractive property of S, which was used to prove convergence of SDS in Chapter 4. One can intuitively characterise a contraction by the ratio of a suitably chosen measure of a domain of a mapping to the measure of its image under this mapping. In the current case both the domain and its image under S are subsets of a two dimensional plane in R^3 and therefore a suitable measure is equivalent to an area of a set. This leads to the conclusion that Figure 5.1 - Figure 5.3 suggest that the lower a sum of probabilities of an agent's activation on either of the patterns (given by 1-p$^-$+p$^+$) is, the more pronounced is the contraction of S.

In order to formalise the above mentioned intuitions a concept of a dissipative system will be introduced after (Serra, Zanarini, 1990):

Definition. 5.1.

A two dimensional system is called dissipative, if through the iterations there is a contraction of the areas in the plane.

The condition for dissipativity of a system is defined via linearisation.

Namely, if

$$x^{i+1} = f\left(x^i\right)$$
$$f = \left(f_1, f_2\right), \quad x = \left(x_1, x_2\right)$$

then the linearisation of f is defined as

$$x^{i+1} = Jx^i,$$

where J is a Jacobian matrix of f and f is called dissipative if the absolute value of the determinant of the Jacobian J is smaller than one ($|\det(J)| < 1$).

One can apply the definition (and condition) of a dissipative system to the mapping S, defined in equations (5.7), (5.8), because it is effectively a two dimensional mapping due to the constraint of normalisation of probabilities. Thus, one has

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial \pi_1} & \frac{\partial f_1}{\partial \pi_2} \\ \frac{\partial f_2}{\partial \pi_1} & \frac{\partial f_2}{\partial \pi_2} \end{bmatrix}$$

where

$$\begin{aligned}
\frac{\partial f_1}{\partial \pi_1} &= \left(1 - p^-\right)\left[2\left(1 - p_m\right)\left(1 - \pi_1\right) + \left(2p_m - 1\right)\pi_2\right], \\
\frac{\partial f_1}{\partial \pi_2} &= p^+\left[\left(2p_m - 1\right)\pi_1 - 2p_m\left(1 - \pi_2\right)\right], \\
\frac{\partial f_2}{\partial \pi_1} &= p^+\left[-2p_d\left(1 - \pi_1\right) + \left(2p_d - 1\right)\pi_2\right], \\
\frac{\partial f_2}{\partial \pi_2} &= p^+\left[\left(2p_d - 1\right)\pi_1 + 2\left(1 - p_d\right)\left(1 - \pi_2\right)\right]
\end{aligned} \tag{5.10}$$

Substituting partial derivatives (5.10) into the definition of a determinant of the Jacobian and rearranging the terms, one obtains

$$\left|\det(J)\right| = 2p^+\left(1 - p^-\right)\left|1 - \left(p_m + p_d\right)\right|\left[1 - \left(\pi_1 + \pi_2\right)\right]\left[2 - \left(\pi_1 + \pi_2\right)\right]$$

Thus the condition for the mapping S to be dissipative is

$$4p^+\left(1 - p^-\right)\left|1 - \left(p_m + p_d\right)\right| < 1 \tag{5.11}$$

One can see that (5.11) confirms an intuition following Figure 5.1 - Figure 5.3, in that the total probability of activation of an agent indeed has to be sufficiently small for the mapping S to be a contraction. The inequality (5.11) is fulfilled in particular, if the following condition is true

$$4p^+\left(1 - p^-\right) < 1$$

which is a two dimensional counterpart of the corresponding condition discussed in remark 2 after Proposition 4.3 in Chapter 4.

Figure 5.1 - Figure 5.3 show the general action of the mapping S on the set of all possible initial probability distribution vectors, but do not reveal the long term behaviour of the iterations of S, which in fact correspond to the evolution of the Stochastic Diffusion Search in time. In time SDS tends to the equilibrium state which corresponds to S converging to the limit probability distribution vector. In order to calculate the stationary probability distribution vector a fixed point equation (5.3) would have to be solved which, in general, leads to very complex expressions for the solutions. However, a simple form of the solutions can be obtained in a special case.

Namely, if $1\text{-}p^-=p^+=p$ and $p_m=p_d=a$, one can rewrite (5.3), taking into account (5.6)-(5.8), as a system of two equations

$$p\left[-(1-a)(\pi_1-1)^2+a(\pi_2-1)^2-(1-2a)\pi_1\pi_2+1-a\right]=\pi_1$$
$$p\left[a(\pi_1-1)^2-(1-a)(\pi_2-1)^2-(1-2a)\pi_1\pi_2+1-a\right]=\pi_2 \tag{5.12}$$

This case corresponds to the search space with two patterns with equal probability of uniform sampling from the search space and equal overlap with the target. Solving the equation (5.12) amounts to considering special cases of parameters p and a and reducing (5.12) to a linear or quadratic equation. The calculations are straightforward and lead to the following cases:

1. $p = 1$

In this case the equilibrium probability distribution vector is given by (π_1, 1-π_1, 0). This case is special because there are infinitely many equilibrium probability distribution vectors forming a straight line. The condition $p = 1$ corresponds to two ideal instantiations of the target in the search space. As expected, all agents in the SDS will ultimately become active and the distribution of agents between the two targets will be constant and dependent on the initial conditions.

2. $p = 0$

The equilibrium probability distribution vector is given by (0, 0, 1). As expected, this condition corresponds to the situation in which there are no patterns similar to the template in the search space.

3. $\frac{1}{2} \le p < 1,\ a = 0$

The equilibrium probability distribution vector is given by

$$\left(\pi_1, 2-\frac{1}{p}-\pi_1, \frac{1}{p}-1\right) \tag{5.13}$$

Again the fixed points of S are situated on the straight line. This is another formal case similar to the second one; here the probability of finding a pattern is null.

4. $p \ne 1,\ p \ne 0,\ a = \frac{1}{2}$

The equilibrium probability distribution vector is given by

$$\left(\frac{p}{2}, \frac{p}{2}, 1 - p\right) \tag{5.14}$$

This is yet another formal special case, corresponding to the search space filled completely with patterns with equal overlap with the target.

5. $p \neq 1,\ p \neq 0,\ a \neq \frac{1}{2},\ a \neq 0$

This is the most general case which covers most of the situations encountered in the Stochastic Diffusion Search. The equilibrium probability distribution vector is given by

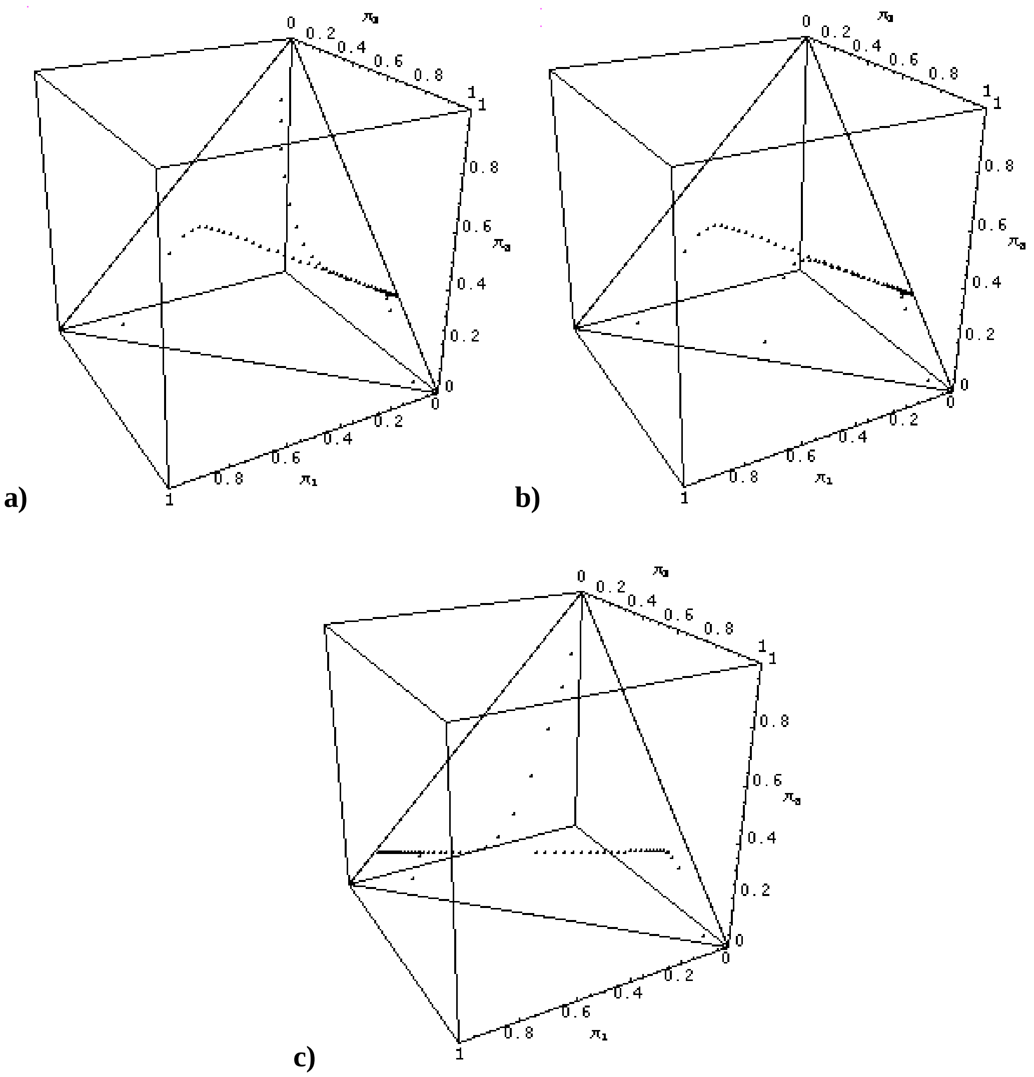


Figure 5.4 Iterations of the mapping S corresponding to different parameters or initial probability distribution vectors. Consecutive points correspond to consecutive iterations of S. In (a) and (b) values of parameters are the same, p^-=0.3, p^+=0.8, p_m=p_d=0.0001, but the initial probability distributions are different; in (c) values of parameters are p^-=0.1, p^+=0.8, p_m=p_d=0.0001 and the initial probability distributions are the same as in (a).

$$\begin{aligned}\pi_1 &= \frac{2(1-2a)-\frac{1}{p}+\sqrt{4(1-2a)\left(1-\frac{1}{p}\right)+\frac{1}{p^2}}}{4(1-2a)},\\ \pi_2 &= \pi_1,\\ \pi_3 &= 1-2\pi_1\end{aligned} \tag{5.15}$$

The last case confirms the intuition about SDS that, for two patterns with equal overlap with the target, the equilibrium probabilities of finding an agent activated by either of them will be equal.

In general, the solution to the equation (5.3), although possible in principle, is very complex. Thus, it is more instructive to consider the consecutive iterations of S, corresponding to the evolution of the Stochastic Diffusion Search.

Figure 5.4 - Figure 5.6 show orbits of the mapping S corresponding to different initial probability distribution vectors. From Figure 5.4, Figure 5.5 one can see that, as expected, irrespective of parameters or initial conditions, iterations of S always converge to a single point. However, the orbits of S depend significantly on the parameters of the search as well as on the initial probability distribution vector. Depending on the parameters and initial probability distribution vectors some of the orbits follow straight to the limit point, e.g. in Figure 5.5 (a), (b), whereas others converge first to a common valley, along which they subsequently approach the equilibrium, e.g. Figure 5.4. Yet another behaviour is visible from Figure 5.5 (c), (d), in which orbits approach equilibrium probability distribution vector along different trajectories.

In Figure 5.4, Figure 5.5 consecutive points correspond to consecutive iterations of S. Thus, in these figures, S approaches rapidly a neighbourhood of the equilibrium probability distribution vector. However, for some settings and initial probability vectors the convergence towards the equilibrium can be very slow. Figure 5.6 shows one such case. The first iteration of S moves the current probability distribution vector into the valley leading towards the equilibrium, but consecutive convergence towards the equilibrium point is extremely slow.

In fact, distance between each two points, except the first two, corresponds to 60 000 iterations of S.

However, in the calculations performed, for the same settings of parameters but different initial probability distribution vectors no such slow convergence have been observed.

The case in Figure 5.6 is special in the sense that the settings of the parameters correspond to the search space with two patterns with the same overlap with the target.

The solution to the system (5.12) for these parameters is, from case 5, given by the vector (0.444446, 0.444446, 0.111108).

From numerical calculations used in conjunction with plotting all the graphs in this section, it follows that the first iteration brings S to the probability distribution vector with the sum of the first co-ordinates closely approximating the sum of the first two co-ordinates of the equilibrium probability distribution vector. Thus, even though the actual equilibrium is approached very slowly, nevertheless the estimation of the probability of finding an agent

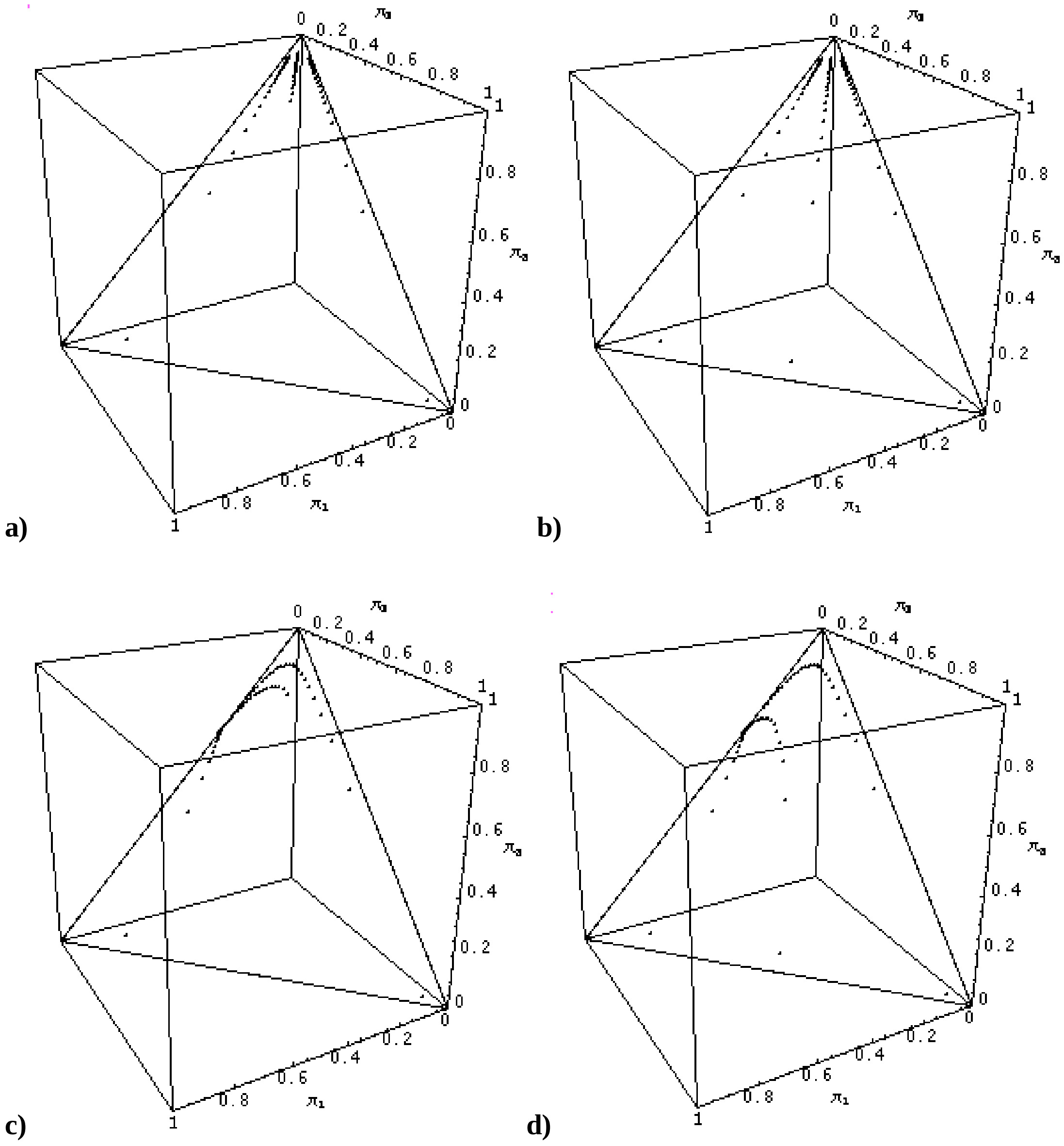


Figure 5.5 Iterations of the mapping S corresponding to different parameters or initial probability distribution vectors. Consecutive points correspond to consecutive iterations of S. In (a) values of parameters are $p^-=0.5$, $p^+=0.5$, $p_m=p_d=0.0001$ and the initial probability distribution vectors are as in Figure 5.4 (a); (b) values of parameters are $p^-=0.5$, $p^+=0.5$, $p_m=p_d=0.0001$ and the initial probability distribution vectors are as in Figure 5.4 (b); (c) values of parameters are $p^-=0.4$, $p^+=0.45$, $p_m=p_d=0.0001$ and the initial probability distribution vectors are as in Figure 5.4 (a); (d) values of parameters are $p^-=0.4$, $p^+=0.45$, $p_m=p_d=0.0001$ and the initial probability distribution vectors are as in Figure 5.4 (b).

active on any of the two patterns is rapid.

The possible reason of the slow convergence of S to the equilibrium is that the parameter a, describing the probability of a success in a uniformly random sampling of a pattern from the search space is small. However, as could be seen from the case 3, for a=0, there are infinitely many fixed points forming a segment contained in X consisting of points with the third co-ordinate constant and equal, in the present case, to 0.11(1).

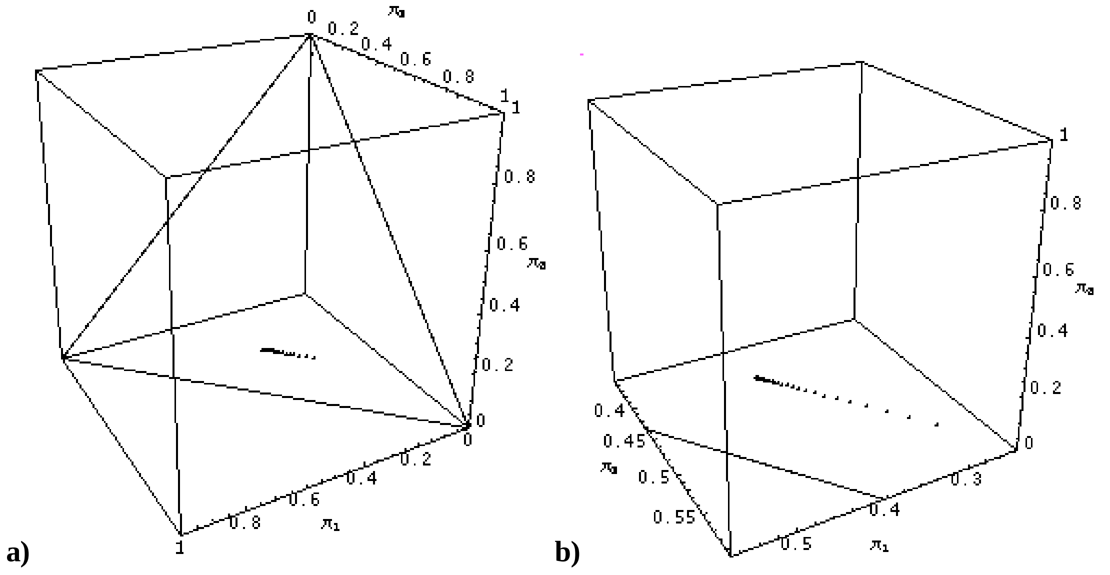


Figure 5.6 (a) Points on the trajectory of the mapping S. The distance between any two consecutive points (excluding the first two) corresponds to 60 000 iterations of S. The parameters are p^-=0.1, p^+=0.9, p_m=p_d=0.0001 and the initial probability distribution vector is (0.02, 0.04, 0.94); (b) zoom on the lower part of the trajectory shown in (a).

It follows, that very small value of the parameter a results, in this case, in the situation being intermediate between those described by the cases 3 and 5, i.e. iterations of S acting on points with a third co-ordinate constant and equal to the third co-ordinate of the equilibrium point move very slowly towards the unique fixed point.

From the above figures it follows that, except for the case of equal corresponding parameters characterising the two patterns in the search space similar to the target, consecutive iterations of S converge to the boundary of X. In fact, the asymptotic ergodicity of the Ehrenfest Urn model implies that the equilibrium probability distribution vector must have strictly positive co-ordinates. This is indeed the case here, however, for most of the settings of parameters the distribution of probabilities in the equilibrium probability distribution vector is strongly biased, resulting in an ill-conditioned problem. The equilibrium probabilities of finding an agent being activated by one of the patterns differ by several orders of magnitude, as can be observed in Figure 5.7. The panels in Figure 5.7 describe the same surface for different values of parameters p_m and p_d.

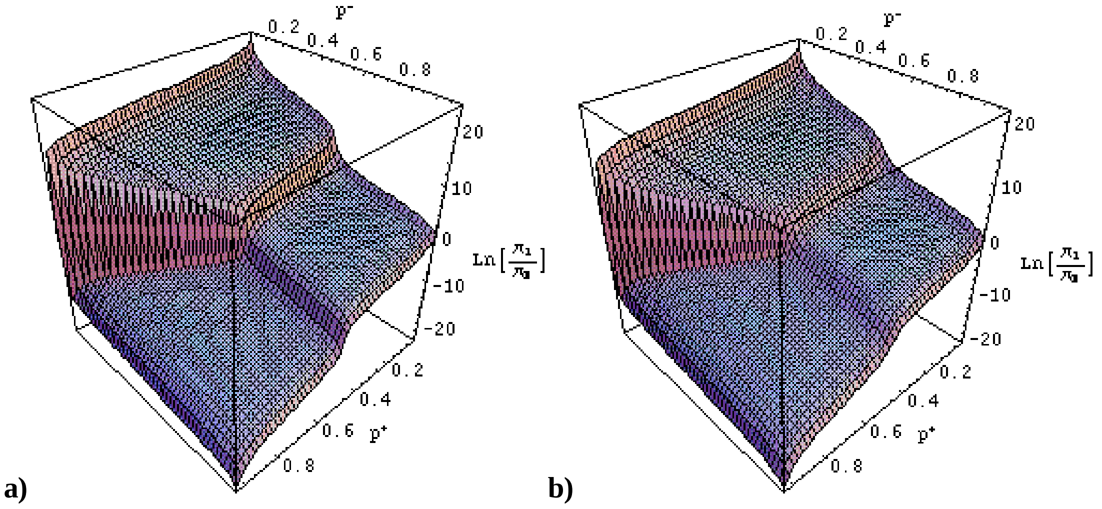


Figure 5.7 Each point on these surfaces, corresponding to different settings of probabilities of successful testing, denotes order of magnitude of difference between equilibrium probabilities of finding an agent activated by one of the two patterns in the search space;(a) p_m=p_d=0.0001, (b) p_m=p_d=0.001.

The surface described in Figure 5.7 (a) is symmetrical with respect to the line lying in the plane $1\text{-}p^-=p^+$ and going through a point (1,0,0). This follows from the fact that, in a formulation of the Ehrenfest Urn model, π_1 and p^- were associated with the particular pattern in the search space, which was assumed to be the best fit, whereas π_2 and p^+ were correspondingly associated with a distractor. However, a distinction between the best instantiation and a distractor is arbitrary and if one assumes, that both p^- and p^+ can take on any values between 0 and 1, then, depending on their values, either one or the other will be the best fit to the target, hence the symmetry. One can also see that almost everywhere the surface is quite flat, with three regions of very steep change of magnitude, corresponding to either of the parameters being equal to about 0.5 or their sum being equal to 1. It is also possible to notice that, except for the region, in which $1\text{-}p^-$ and p^+ are less than 0.5 or $1\text{-}p^- = p^+$, there is a significant difference in orders of magnitude between both probabilities π_1 and π_2. This explains the convergence of S to the boundaries of X in Figure 5.2 - Figure 5.5. This, together with the steepness of the slope of the surface in the region in which $1\text{-}p^- = p^+$ and $1\text{-}p^- > 0.5$, suggests also that SDS will make a sharp distinction between the best instantiation and a distrator by allocating orders of magnitude more agents to the former than to the latter. From numerical calculations it follows that this is true even if the difference in overlaps with the target pattern is less than 1%. This means that SDS is particularly insensitive to signal to noise ratio, in cases when the signal - the best fit pattern, is strong enough.

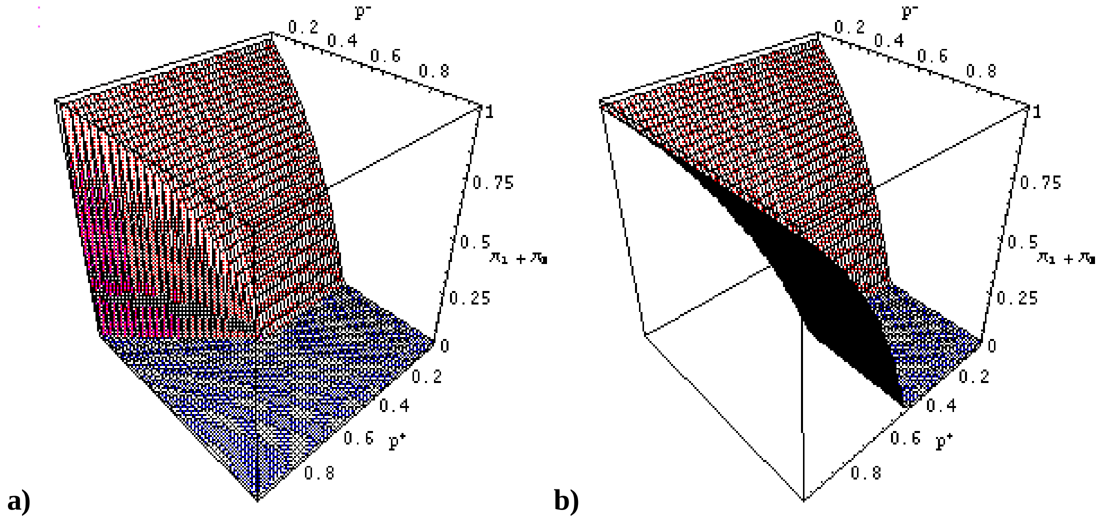


Figure 5.8 Equilibrium probability of finding an agent activated by the first pattern (a) and the total probability of finding an agent activated, i.e. a sum of π_1 and π_2 ; p_m=p_d=0.0001.

The signal to noise ratio sensitivity is still good in the region where the signal is weaker, however the corresponding change in orders of magnitude between probabilities π_1 and π_2 is less pronounced and can be considered approximately linear in p^+ in the neighbourhood of points where $1-p^- = p^+$.

Because of such big differences in the orders of magnitude between π_1 and π_2 it is impossible to include them in a meaningful way in a single figure. However, due to symmetricity and difference in magnitudes, it is sufficient to show graphs of one of them alone and of their sum. Figure 5.8 (a) shows π_1 as a function of p^- and p^+ and Figure 5.8 (b) shows the sum of π_1 and π_2.

It is worth mentioning that the proceeding discussion of these figures was biased by the implicit assumption about a uniqueness of the solution in the search space. However, if one relaxes this assumption then SDS is faced with a problem of exploration of multiple solutions (in the particular case discussed in this chapter there would be only two of them). One can see that, unless both solutions are virtually equally similar to the template, SDS would quickly switch, on a system level, to serial processing mode by forming the unique largest cluster. It would be also very biased towards the ‘stronger’ solution, supporting a ‘weaker’ solution could occur practically only in the transient phase. Only for virtually equally ‘strong’ solutions, their corresponding cluster representations would be of comparable size, thus only in such a case SDS would support more parallel form of processing.

Another conclusion, following from the results presented in this section and fitting in their second interpretation discussed above, concerns the way the Stochastic Diffusion Search processes the search space in order to find solutions. The search can be characterised as an exploration, when resources are distributed over different regions of the search space or as an exploitation, when they are concentrated in a particular part of the search space in order to test the quality of found solution. In this terms SDS strikes a balance between exploration and exploitation favouring the latter over the former. More specifically, once a ‘reasonable’ solution is found (e.g. with $p^{-}<0.5$) SDS quickly switches from exploration to exploitation by concentrating a large proportion of its resources on this solution. This bias towards the exploitation of the search space is of course a result of a very high sensitivity of the resource allocation to the difference in ‘strength’ between the potential solutions.

5.4. Equilibrium probability distribution of agents in Stochastic Diffusion Search.

The procedure of calculating the equilibrium probability distribution of a single agent in Stochastic Diffusion Search was a straightforward extension of the nonhomogenous generalised Ehrenfest urn model described in Chapter 4. Similarly the state space of the whole ensemble of processes describing evolution of N agents in SDS consists of N - tuples $x = (x_1,...,x_N)$, where the i^{th} component has a value 1, 2, or 0 depending on whether the i^{th} agent was active and pointing to the first pattern, or active and pointing to the second pattern, or inactive.

It follows, from the strong ergodicity of a Markov Chain corresponding to the evolution of a single agent, that the unique asymptotic limit probability distribution is given by

$$\Pi(x) = \pi_1^{a_1[x]}\pi_2^{a_2[x]}\pi_3^{N-a_1[x]-a_2[x]} \tag{5.16}$$

where $a_1[x]$, $a_2[x]$ denote numbers of active agents, pointing to the first and second patterns respectively, corresponding to the state x and (π_1, π_2, π_3) is the equilibrium probability distribution vector found from the fixed point equation (5.3).

In order to characterise the steady state behaviour of the Stochastic Diffusion Search one replaces the nonhomogenous Markov Chains described by stochastic matrices P_i by the homogenous Markov Chains obtained by replacing the stochastic matrices P_i by their limit P. The steady state probability distribution of an ensemble of such Markov Chains will be a limit probability distribution attained by the ensemble of original processes.

Following the reasoning outlined in Chapter 4 one can deduce that the process obtained by lumping together states corresponding to the same numbers of agents active on the first and second pattern, will be a Markov Chain, (Whittle, 1986). Summing the probabilities in (5.16) over all N-tuples x corresponding to the same numbers of agents active and pointing to the first and second pattern one obtains a multinomial probability distribution

$$\pi\left(n_1, n_2\right) = N!\prod_{j=1}^{3} \frac{\pi_j^{n_j}}{n_j!} \tag{5.17}$$

with

$$n_j \geq 0, \quad \sum n_j = N \cdot$$

It is well known, (Johnson, Kotz, 1969), that the maxima of the multinomial distribution are located in the vicinity of the expected value point. More precisely, the following inequalities are satisfied, (Johnson, Kotz, 1969):

$$N\pi_i < E\left[n_i\right] \leq (N + k - 1)\pi_i, \quad i = 1, 2, 3$$

and they, together with the normalisation condition of (5.17) ensure that there are at most relatively few modes. In fact, as the present case is of small dimensionality, it is very likely that for most settings of parameters there will be only a unique maximum. In any case, it is known that there is only one region in which all maxima are placed, or in other words "(...) there is no local maximum remote from the set of equiprobable points, each giving the maximum value of $P(n_1, ..., n_k)$", (Johnson, Kotz, 1969).

5.5. Numerical examples and simulations.

Figure 5.9 - Figure 5.11 show joint probability distributions for various settings of parameters and numbers of agents. In the first two figures the parameter N, corresponding to the number of agents, was set to 100. In Figure 5.9 (a) one can see the joint probability distribution for the best fit pattern with high overlap with the target, $p^-=0.2$, and a distractor with relatively low overlap, $p^+=0.2$. Figure 5.9 (c) shows the joint probability distribution for $p^-=0.5$, $p^+=0.3$.

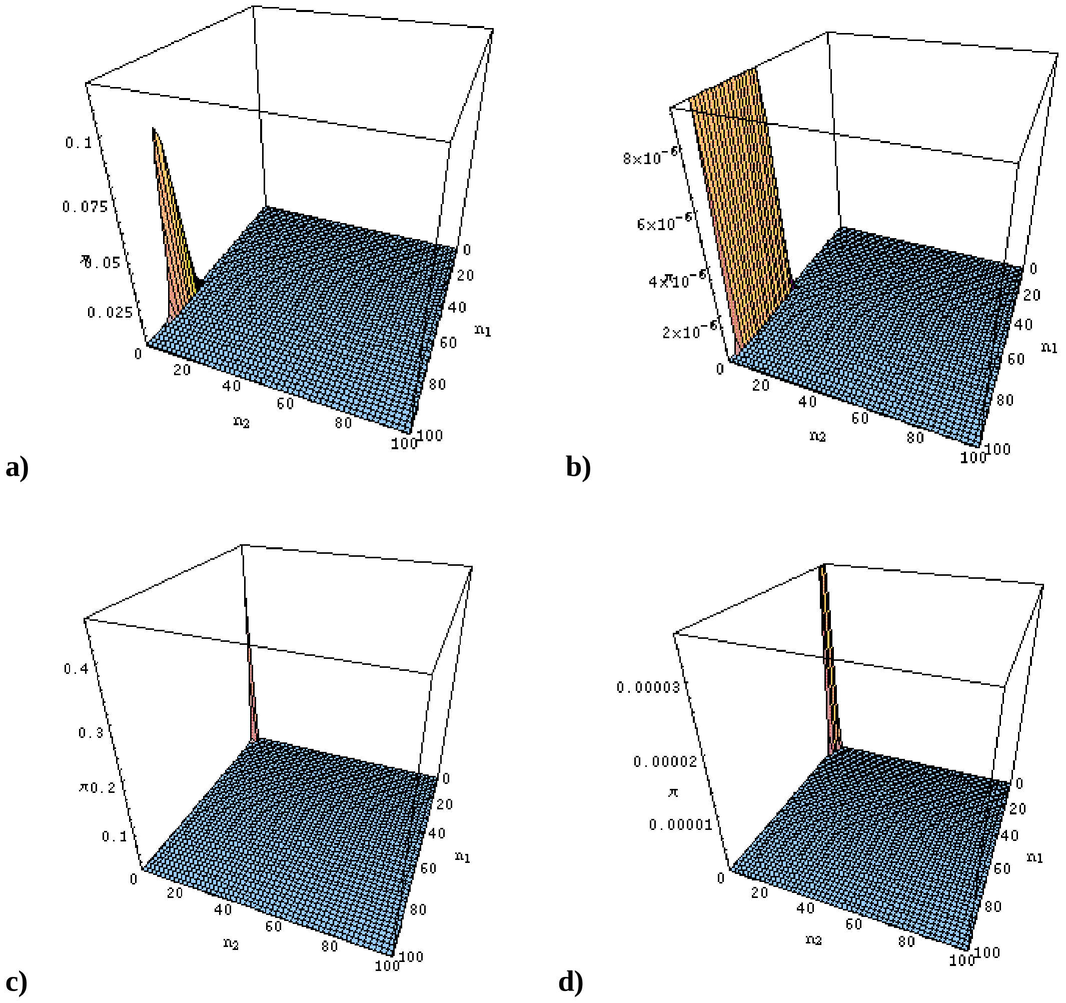


Figure 5.9 The joint probability distribution of a number of active agents allocated to two patterns as predicted by the model; $p_m=p_d=0.0001$, N=100. (a) $p^+=0.2$, $p^-=0.8$; (b) the same as in (a) but cut at $9x10^{-6}$; (c) $p^+=0.3$, $p^-=0.5$; (d) the same as in (c) but cut at $4x10^{-5}$.

Figure 5.9 (b) and (c) show the same joint probability distribution cut at the base.

From Figure 5.9 one can see that in both cases probability is well concentrated around relatively small region of n_1 x n_2 plane, probabilities of finding a configuration outside of these regions being practically negligible. Also it can be observed that, as expected, when the probability of activation on the best fit pattern decreases to 0.5, the joint probability distribution peaks, for this number of agents, around zero.

Also in both cases the modes of the joint probability distributions are at the borders of the plane, reflecting the big difference in magnitudes between equilibrium probabilities found from the extended Ehrenfest model.

Figure 5.10 shows analogous display of plots. Panels (a) and (b) show the joint probability distribution for both patterns equal; $1-p^{-}=p^{+}=0.9$. One can see that the mode is situated symmetrically on the line $n_1=n_2$ and its crossection extends perpendicularly to this diagonal. This reflects the fact that the number of agents active and pointing to one of the patterns can

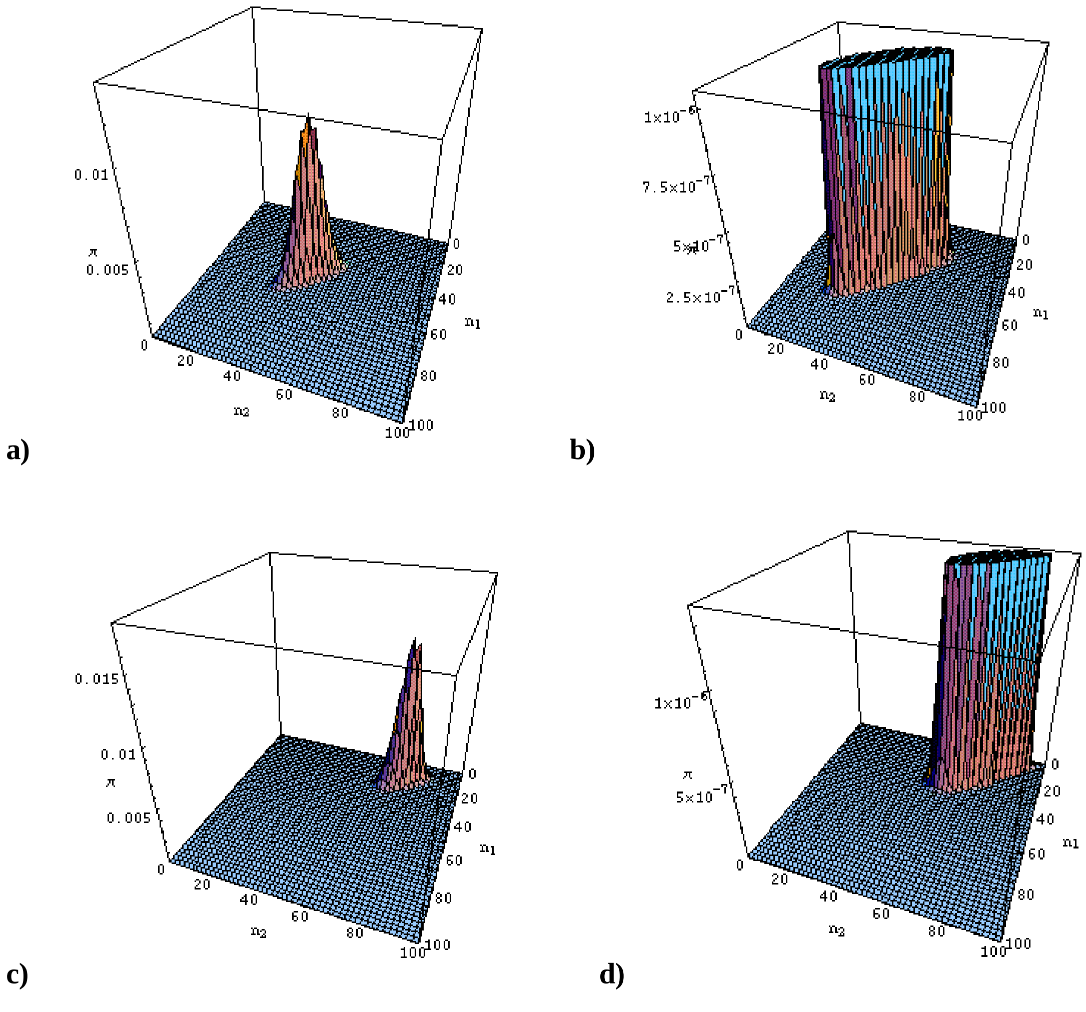


Figure 5.10 The joint probability distribution of a number of active agents allocated to two patterns as predicted by the model; $p_m=p_d=0.0001$, N=100. (a) $p^{+}=0.9$, $p^{-}=0.1$; (b) the same as in (a) but cut at $1x10^{-6}$; (c) $p^{+}=0.900005$, $p^{-}=0.1$; (d) the same as in (c) but cut at $1.5x10^{-6}$.

increase only at the expense of the decrease of the number of agents pointing to the other pattern. Panels (c) and (d) show analogous information, but for slightly ‘weaker’ first pattern - a distractor with p^-=0.900005. It follows, that although the difference between the patterns is very small, the mode of the joint probability distribution is shifted significantly towards the region of small n_1’s and high n_2’s, reflecting a very nonlinear dependence on the signal to noise ratio.

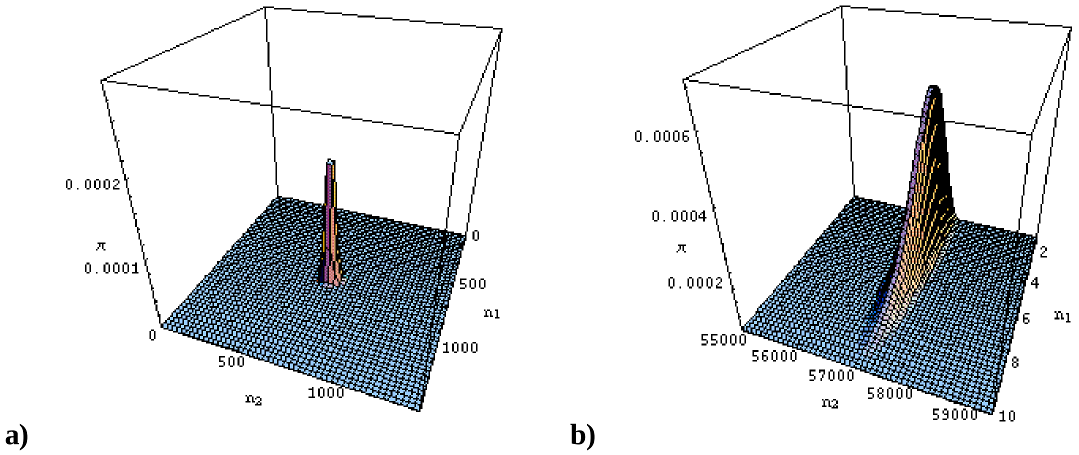


Figure 5.11 The joint probability distribution of a number of active agents allocated to two patterns as predicted by the model. Both panels show only a rectangular segment of the plane n_1xn_2, containing the significant information; p_m=p_d=0.0001, N=10^5. (a) p^+=0.5, p^-=0.5; (b) p^+=0.7, p^-=0.5.

Figure 5.11 shows the results obtained for a much higher number of agents, N=100 000. In Figure 5.11 (a) one can see that for equal patterns with 1-p^-=p^+=0.5 the mode of the joint probability distribution is again situated symmetrically, as in Figure 5.10 (a). However, the maximum value is two orders of magnitude lower than in the previous figure and the mode do not extend away from the diagonal. Moreover, the most probable numbers of active agents are very small in comparison to their total number.

Figure 5.11 (b) shows the joint probability distribution function for p^- as in panel (a) but with p^+ increased to 0.7. One can observe a marked asymmetry of the positioning of the region of maximal probability and of the mode itself. Namely, the mode is now located in the region of very low n_1 and n_2 fluctuating around 57000. Also, as implicated by the probability distribution, relative fluctuation of numbers of agents allocated to the first pattern is much higher than the corresponding fluctuation for the second, stronger pattern (however, this is not necessarily so for the absolute variations).

In order to assess the model several simulations were run, in which the parameters characterising the best fit pattern and a distractor were varied. Figure 5.12 - Figure 5.15 show the evolution of clusters associated with the patterns and lines corresponding to the expected value predicted by the model surrounded by +/- two standard deviations bands.

In some graphs of these figures and some figures hereafter the very low relative variation of the cluster sizes results, due to the limits of resolution, in Mathematica drawing only a single thick line. Also, if the expected cluster size decremented by two standard deviations were negative, the corresponding to this value stright line was not drawn.

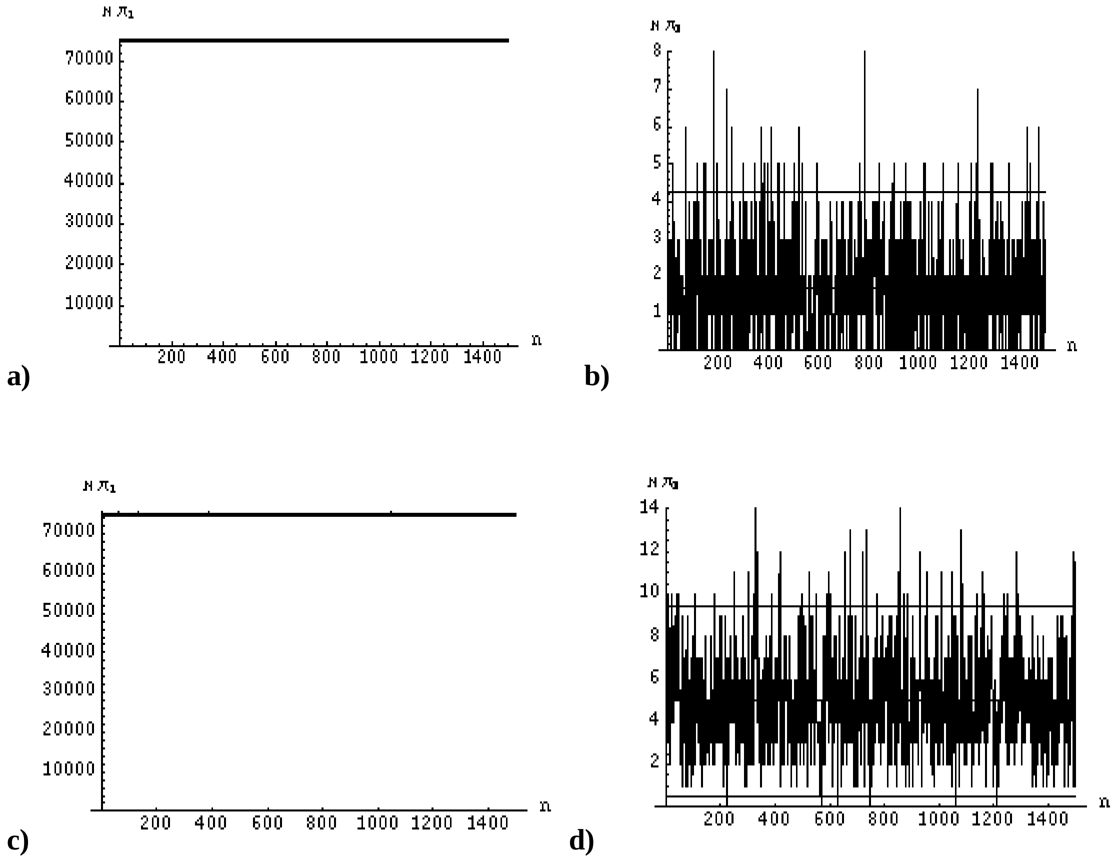


Figure 5.12 Evolution of SDS with two patterns in the search space; N=100000, p_m=p_d=0.001. In panels (a) and (b) p^-=0.2, p^+=0.2. (a) The evolution of the cluster associated with the best fit pattern (the expected value is 75006.7, the standard deviation is 136.9, a sample average is 75006.2 and a sample standard deviation is 142.292); (b) the evolution of the cluster corrsponding to the distractor (the expected value is 1.66, the standard deviation is 1.3, a sample average is 1.64 and a sample standard deviation is 1.34). In panels (c) and (d) p^-=0.2, p^+=0.4. (c) The evolution of the cluster associated with the best fit pattern (the expected value is 75003.3, the standard deviation is 136.9, a sample average is 75004.4 and a sample standard deviation is 141.38); (d) the evolution of the cluster corrsponding to the distractor (the expected value is 4.99, the standard deviation is 2.24, a sample average is 4.99 and a sample standard deviation is 2.32).

The extended Ehrenfest urn model predicts that, in the steady state, the distribution of agents in SDS between states "active and pointing to the best fit", "active and pointing to the distractor" and "inactive" will be multinomial. This makes it possible to calculate the first two moments connected with the active clusters. Due to properties of the multinomial distribution these moments - expected value and variation, will be given by formulas describing these quantities for binomial distribution, where the latter is the marginal distribution obtained from the multinomial distribution, (Johnston, Kotz, 1969).

Namely

$$\begin{aligned} &E[n_i] = N\pi_i, \\ &\mathrm{var}[n_i] = N\pi_i(1-\pi_i) \end{aligned}$$

where π_i is an equilibrium probability of an agent becoming active and locked to a corresponding cluster, as established from a solution to (5.3).

The expected values, standard deviations and their estimates from the displayed samples are quoted in the corresponding captions to figures. In all cases the estimates were calculated from the sample, excluding first 500 values.

This procedure is known in the Monte Carlo Markov Chain literature as an exclusion of burn-in period (Gilks et al, 1996) during which the stochastic process reaches its steady state. In general it improves the estimates, although, as will become clear from the further examples, the number of iterations necessary for the SDS to reach an equilibrium can vary to a great extent.

To illustrate this point, consider data from Figure 5.12 (a). The expected number of agents locked to the best fit pattern is 75006.7 and its standard deviation is 136.9. The estimates obtained from the sample of 2000 iterations are 74500 and 5400 respectively, whereas after exclusion of first 500 points, corresponding estimates from the remaining 1500 sample of iterations are 75006.2 and 142.3, which indeed are very close to the predicted values.

Figure 5.12 and Figure 5.13 show evolution of the Stochastic Diffusion Search with 100000 agents, the best fit pattern with p^{-}=0.2.

In these figures the probability of a successful test of the distractor is varied and its influence on the distribution of agents among both patterns is observed.

In Figure 5.12 distractor changes from p^{+}=0.2 in panels (a), (b) to p^{+}=0.4 in panels (c), (d). In Figure 5.13 the value of p^{+} is further increased from 0.5 in panels (a), (b) to 0.6 in panels (c), (d). From Figure 5.12 and Figure 5.13 it follows that increasing the strength of the distractor almost does not influence the cluster associated with the best fit pattern, however, it results in

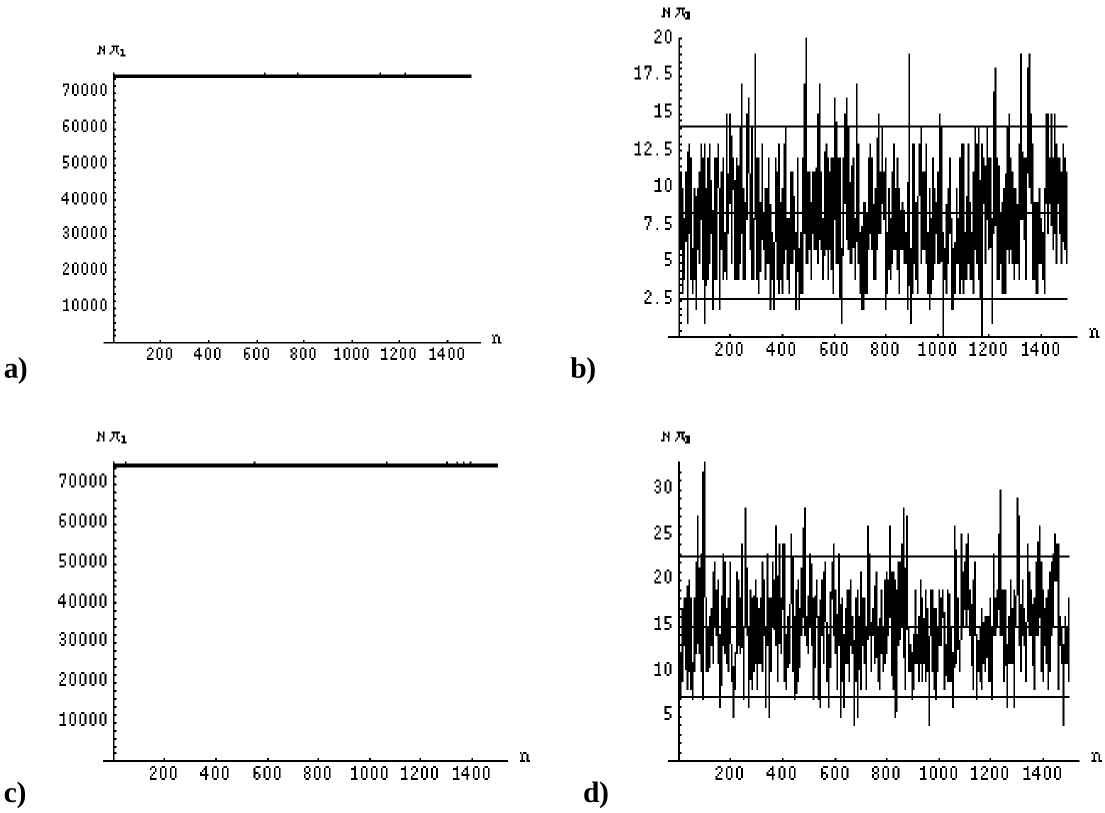


Figure 5.13 Evolution of SDS with two patterns in the search space; N=100000, p_m=p_d=0.001. In panels (a) and (b) p^{-}=0.2, p^{+}=0.5. (a) The evolution of the cluster associated with the best fit pattern (the expected value is 75000, the standard deviation is 136.9, a sample average is 74992.5 and a sample standard deviation is 144.5); (b) the evolution of the cluster corrsponding to the distractor (the expected value is 8.33, the standard deviation is 2.89, a sample average is 8.05 and a sample standard deviation is 3.13). In panels (c) and (d) p^{-}=0.2, p^{+}=0.6. (c) The evolution of the cluster associated with the best fit pattern (the expected value is 74993.3, the standard deviation is 136.9, a sample average is 74994.1 and a sample standard deviation is 146.49); (d) the evolution of the cluster corrsponding to the distractor (the expected value is 14.99, the standard deviation is 3.87, a sample average is 14.94 and a sample standard deviation is 4.37).

a systematic increase of the number of agents associated with the distractor.

Although the corresponding cluster's size decreases frequently to null in Figure 5.12 (b), it does so much less frequently in Figure 5.12 (d) and Figure 5.13 (b) and it does not vanish at all in Figure 5.13 (d).

It can also be noted that both clusters remain most of the iterations within bounds determined from the model.

Although the absolute value of the standard deviation is higher for the cluster associated with the best fit pattern than with the distractor, its value relative to the value of the expectation is very small, whereas the corresponding relative variation of the cluster associated with the distractors is significant.

Figure 5.12 and Figure 5.13 illustrate the prediction of the model, that for most of the values of p^+ such that $p^+<1-p^-$ there is a high difference in orders of magnitude between equilibrium probabilities of activation of an agent on a particular pattern (compare Figure 5.7).

It follows from the model that different pattern of the Stochastic Diffusion Search behaviour can be observed only when the parameter p^+ approaches the value of $1-p^-$. Therefore Figure 5.14 and Figure 5.15 illustrate the evolution of distribution of N=100000 agents of SDS for $p^-=0.5$ and p^+ increasing in steps of 0.02 from 0.42 in Figure 5.14 (a), (b) to 0.48 in Figure 5.15 (c), (d).

In these figures one can observe, with the increase of the parameter p^+, the decrease of the size of the best fit pattern cluster and a corresponding increase of the distractor's cluster. However, it is clear that the increase of the distractor's cluster is higher than decrease of the best fit match cluster. The relative changes in the size of the clusters are also much higher than corresponding changes observed in Figure 5.12 and Figure 5.13, although the parameter's p^+ range of change is only 0.1 in this case. Figure 5.14 and Figure 5.15 correspond to approaching the diagonal $p^+=1-p^-$ on the surface in Figure 5.7 along the line of a constant parameter p^-. It also follows from these figures, that the absolute values of the clusters' sizes do not differ to the extent observed in Figure 5.12 and Figure 5.13. Indeed, the minimal ratio of cluster sizes is around 0.076 as opposed to the maximal ratio of 0.0002 observed in Figure 5.12 and Figure 5.13.

All these graphs illustrate a highly non-linear dependence of the relative size of clusters on the relative value of parameters characterising the best fit pattern and a distractor.

Another feature of the graphs shown in Figure 5.14 and Figure 5.15 is a higher discrepancy between the expected value and the standard deviation predicted by the model and their finite sample estimates. This is much more pronounced for the graphs showing the evolution of the best fit pattern cluster size than the distractor's cluster size which apparently settles much better on the equilibrium. The changes in the latter are much more irregular but remain more often in the four standard deviation band around the predicted expected value. This is not the

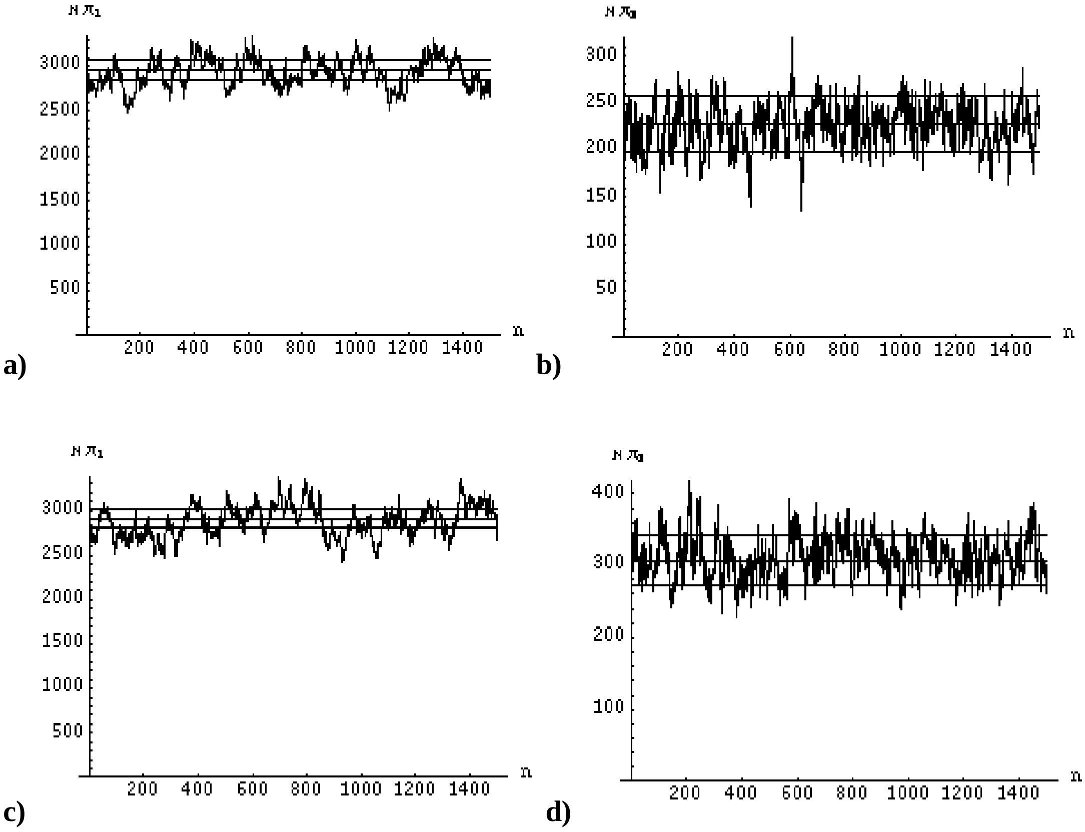


Figure 5.14 Evolution of SDS with two patterns in the search space; N=100000, p_m=p_d=0.001. In panels (a) and (b) p^-=0.5, p^+=0.42. (a) The evolution of the cluster associated with the best fit pattern (the expected value is 2950.33, the standard deviation is 53.51, a sample average is 2924.6 and a sample standard deviation is 159.59); (b) the evolution of the cluster corrsponding to the distractor (the expected value is 227.14, the standard deviation is 15.05, a sample average is 225.6 and a sample standard deviation is 24.27). In panels (c) and (d) p^-=0.5, p^+=0.44. (c) The evolution of the cluster associated with the best fit pattern (the expected value is 2910.77, the standard deviation is 53.16, a sample average is 2876.57 and a sample standard deviation is 174.88); (d) the evolution of the cluster corrsponding to the distractor (the expected value is 307.2, the standard deviation is 17.5, a sample average is 310.55 and a sample standard deviation is 30.34).

case for the best fit pattern cluster, in which changes seem to be more serially correlated, i.e. one can clearly notice interchanging periods when the cluster size is systematically higher or lower than the expected value. Also, during these periods the cluster size remains often more than two standard deviations above or below the expected value. These features suggest that the best fit pattern cluster size evolution did not reach stationarity. This may be caused by two factors.

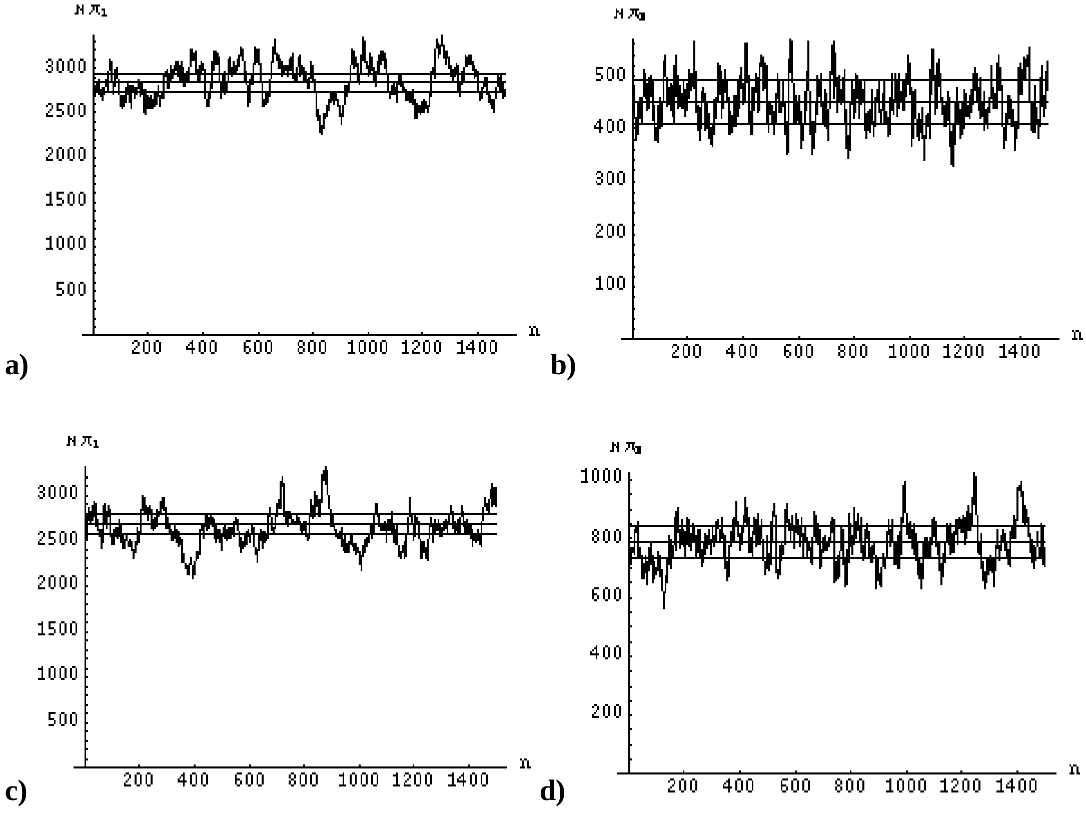


Figure 5.15 Evolution of SDS with two patterns in the search space; N=100000, $p_m=p_d=0.001$. In panels (a) and (b) $p^-=0.5$, $p^+=0.46$. (a) The evolution of the cluster associated with the best fit pattern (the expected value is 2840.42, the standard deviation is 52.53, a sample average is 2858.54 and a sample standard deviation is 202.69); (b) the evolution of the cluster corrsponding to the distractor (the expected value is 452.16 the standard deviation is 21.22, a sample average is 450.31 and a sample standard deviation is 42). In panels (c) and (d) $p^-=0.5$, $p^+=0.48$. (c) The evolution of the cluster associated with the best fit pattern (the expected value is 2683.25, the standard deviation is 51.1, a sample average is 2630.85 and a sample standard deviation is 191.73); (d) the evolution of the cluster corrsponding to the distractor (the expected value is 789.23, the standard deviation is 27.98, a sample average is 785.69 and a sample standard deviation is 68.43).

Firstly, although the condition (5.11) is fulfilled, nevertheless the expression on the left hand side of the inequality (5.11) is very close to one, so effectively the mapping S describing the evolution of SDS is a very weakly dissipative system.

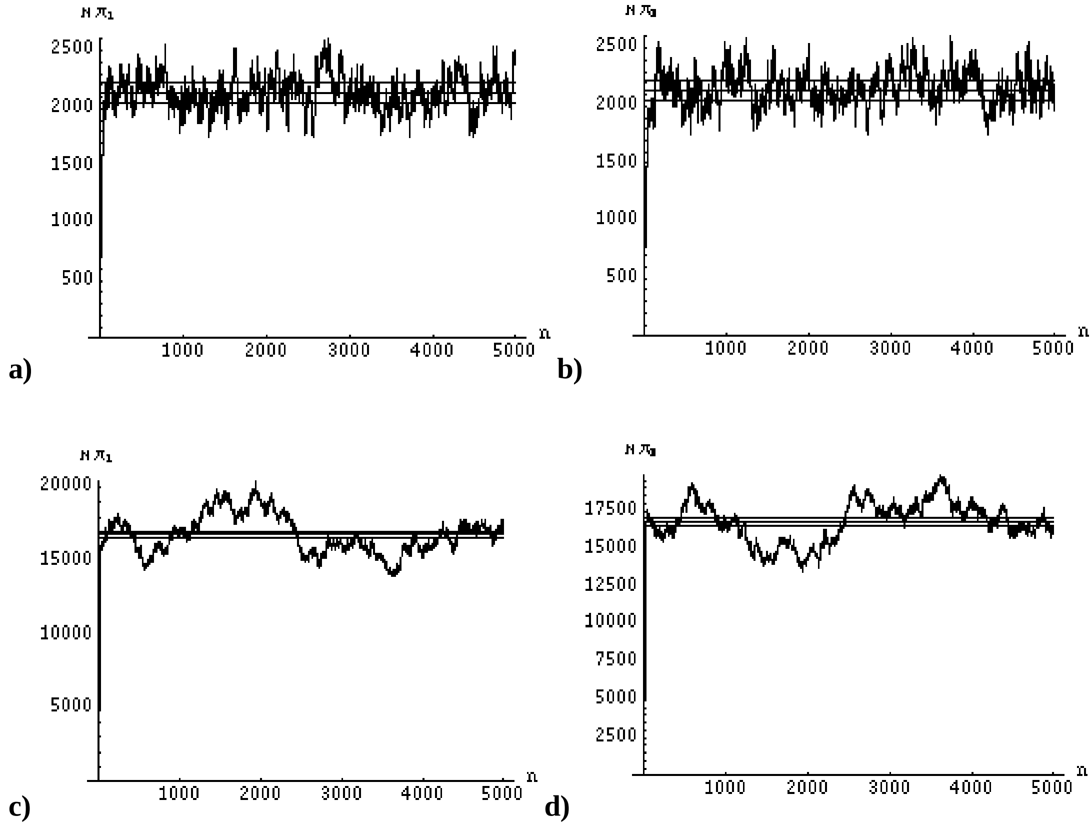


Figure 5.16 Evolution of SDS with two best fit matches in the search space; N=100000, $p_m=p_d=0.001$, $p^+=1-p^-=0.5$, the expected value predicted from the model is 2140.35, and the standard deviation is 45.77; (a) the evolution of the cluster associated with the first best fit pattern (the sample average is 2125.66 and a sample standard deviation is 192.67), and the second one (b) (the sample average is 2142.83 and the sample standard deviation is 194.7). The evolution of the clusters associated with the best fit patterns for $p^+=1-p^-=0.6$ is shown in panels (c), (a sample average is 16742.4 and a sample standard deviation is 1619.1), and (d) (the sample average is 16712.9 and a sample standard deviation is 1618.58); The expected value for the clusters displayed in panels (c) and (d) is 16797.9 and the standard deviation is 118.22.

On its own this does not necessarily cause slow convergence towards equilibrium, as seen in cases with significant difference in orders of magnitude between equilibrium probabilities π_1 and π_2.

However, secondly, the values of p^+ are relatively close to that of $1-p^-=0.5$ in Figure 5.14 and Figure 5.15, which makes these cases similar to the solution of system of equations (5.12), case 3 (compare discussion of Figure 5.6). The influence of the latter condition seems to increase with increase of the parameter p^+, as can be seen from the evolution of the distractor's cluster in Figure 5.15 (b), (d).

It seems instructive to analyse further the behaviour of SDS in the case when both patterns in the search space are equal, $p^+=1-p^-$, i.e. when there are two identical best fit patterns in the search space. Figure 5.16 - Figure 5.18 show the evolution of SDS in this special case.

Figure 5.16 (a), (b) shows the evolution of clusters in case of the targets with $1-p^-=p^+=0.5$. From this figure it follows that the model quite well predicts in this case the cluster´s size. However, the fluctuation around the average value are quite significant - in fact the sample standard deviations are almost 4 times as high as the value predicted from the model. These fluctuations have a very weakly correlated character. One can contrast this relatively stationary behaviour with fluctuations depicted in Figure 5.16 (c). (d), corresponding to $p^+=1-p^-=0.6$.

Although the average cluster sizes differ from predicted by the model by at most 0.5%, nevertheless there are clearly visible serial correlations in the series and the sample standard deviation is high (although it constitutes only about 10% of the sample average, nevertheless it is about 14-fold higher than estimated from the model).

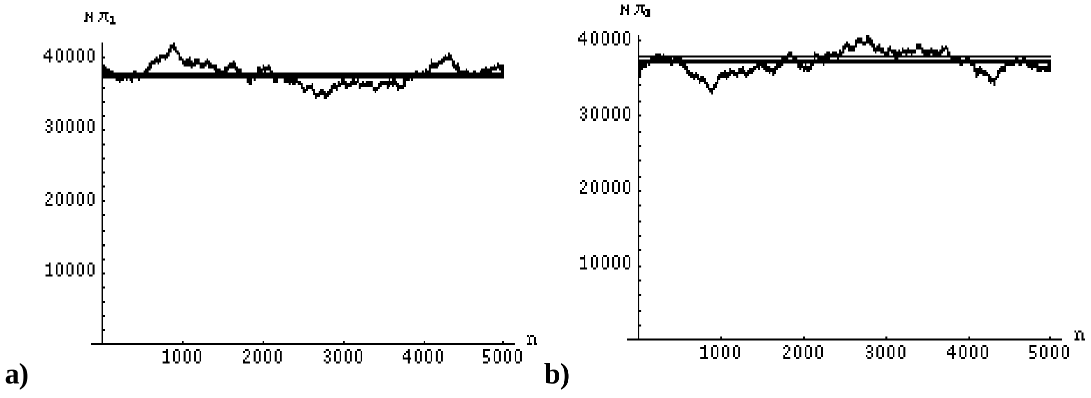


Figure 5.17 Evolution of SDS with two best fit matches in the search space; N=100000, $p_m=p_d=0.001$, $p^+=1-p^-=0.8$, the expected value predicted from the model is 37508.3, and the standard deviation is 153.1; (a) the evolution of the cluster associated with the first pattern (the sample average is 37701.7 and a sample standard deviation is 2140.08), and the second one (b) (the sample

In Figure 5.17 one can see evolution of clusters corresponding to patterns with $p^+=1-p^-=0.8$. The relative difference between the sample average and expected value is smaller than 1% in both cases and the relative variation, as estimated by the ratio of the sample standard deviation to the sample average, is less than 6%. Nevertheless, the nonstationarity is clearly pronounced and is accompanied again by a 14-fold difference between the sample and predicted standard deviations. One can conclude that indeed existence in the search space of two equal patterns with high overlap with the target results in slower convergence to an equilibrium distribution by the Stochastic Diffusion Search.

Figure 5.18 examines the cluster size behaviour when a longer burn-in period is discarded from the sample.

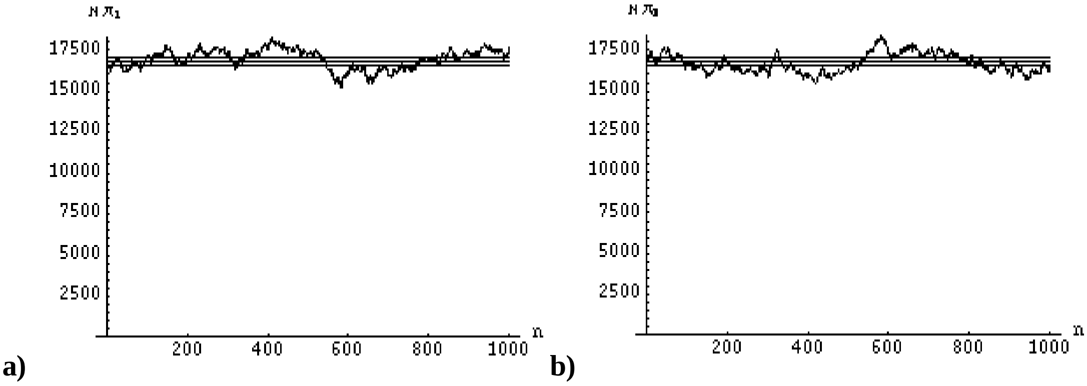


Figure 5.18 Evolution of SDS with two best fit matches in the search space; N=100000, $p_m=p_d=0.001$, $p^+=1-p^-=0.6$; (a) the evolution of the cluster associated with the first pattern (the sample average is 16941.7 and a sample standard deviation is 580.7), and the second one (b) (the sample average is 16679.3 and a sample standard deviation is 576.6). The expected value for the clusters is 16797.9 and the standard deviation is 118.22.

Figure 5.18 should be compared with Figure 5.16 (c), (d). In both cases the parameters characterising the search space and the patterns are the same, but Figure 5.18 displays the last 1000 points from 20 000 long sample (the same 1000 points were used in the estimation of moments).

The average cluster sizes differ from the predicted value by less than 1%, however the difference is slightly higher than in Figure 5.16 (c), (d). Nevertheless, there is a marked decrease in the sample standard deviation both relative to the sample average - 3.4%. Also the ratio of the sample standard deviation to the standard deviation obtained from the model decreases to nearly 5 in Figure 5.18, as compared with 14 from Figure 5.16.

5.6. Conclusions.

This chapter presented an extension of the generalised Ehrenfest urn model to describe the Stochastic Diffusion Search behaviour in the case of the search space with the best fit pattern and one distractor. The model does not require that the parameters characterising both patterns be different, in which case one is analysing SDS under relaxed condition of the unique best fit pattern in the search space. This is because it is assumed that the whole knowledge about the objects in the search space is contained in microfeatures and their relative positions. Thus there is no reason, for two objects of equal characteristics, to differentiate them into the best fit and a distractor.

It follows that it is possible to interpret the results of this chapter in two ways. First interpretation is related to the resource allocation by SDS in the case of distractors present in the search space. This interpretation is linked to the assumption of the uniqueness of the solution in the search space. A very nonlinear bias towards the best fit in allocating the resources by SDS shows that the algorithm has a very high signal from noise discriminability.

The second interpretation of the results follows from a relaxation of the assumption about uniqueness of the solution. From this perspective follows emergence of serial processing of the potential solutions, as for two solutions differing in their similarity to the template, only one largest cluster, corresponding to the best fit solution will be formed in practice. However, one can observe an onset of the parallel mode of processing for solutions with almost the same 'strength'.

Although it is possible in principle to solve the model in the case of two patterns in the search space, in practice the solutions are very complex. Nevertheless, the search still could be characterised by using Mathematica to solve and plot the equilibrium probabilities over ranges of parameters. It appeared that SDS is very sensitive to signal to noise ratio and allocates majority of resources to the pattern which has a slightly higher probability of successful test by agents. This suggests that SDS will be able to cope successfully with the noisy search space. From the presented simulations it follows that the model predicts distribution of resources with a very high accuracy, in cases when the system converges rapidly to the steady state. The cases, in which convergence to equilibrium might be slower were also characterised. In such a situation the model still accurately predicted the total amount of active agents, although their distribution could differ from the equilibrium distribution due to a prolonged transient phase. In the case of two equal patterns in the search space the solutions assumed a particularly simple form. In the next chapter this observation will constitute a basis for further generalisation of the model, which will characterise the steady state behaviour of SDS in the case of homogenous noise.

CHAPTER 6

STEADY STATE BEHAVIOUR OF SDS IN THE CASE OF A HOMOGENOUS NOISE SEARCH SPACE

6.1. *Introduction.*

Chapter 5 characterised the steady state distribution of agents among two patterns having a non-zero overlap with the template. As stated in that chapter it is possible to interpret this results in the context of the number of solutions sought. On the first interpretation, the underlying assumption is that the desired solution is unique and has the best fit to the template. Thus, any other, less optimal, pattern is treated as a disturbance. This leads to the question of how SDS performs its task in the case of a noisy search space.

This chapter will generalise further the model developed in Chapter 5 in order to investigate the resource management of SDS in the case of homogenous noise. The latter is defined in terms of uniformly randomly distributed partial matching patterns with an identical overlap with the template. The starting point of the extension of the generalised Ehrenfest model is the analysis, described in Chapter 5, of the equilibrium probability distribution of an agent in the case of two equal patterns in the search space. Starting from the above mentioned result, the next section will derive the model for the homogenous noise search space. The consecutive section will present a detailed discussion of the obtained solutions with respect to the parameters characterising the search space, the best fit and noise. The next section will illustrate the resource management in the homogenous noise case by means of simulations for chosen values of model parameters. The final section will present conclusions.

6.2. *Modelling the Stochastic Diffusion Search in the search space with homogenous noise.*

Chapter 5 extended the generalised Ehrenfest urn model to the case of two patterns in the search space. Although the general solutions could be, in principle, obtained analytically, their general form was very complex. However, it was possible to obtain simple analytic expressions for the solutions in the special case, when both patterns are equally similar to the template and have the same probability of random sampling from the search space. They will constitute a starting point for extending the model to cope with homogenous noise in the search space. The basis of this extension will be a particular property of the special case solutions, which will be derived next in this section.

The special case solutions will be repeated below for the most general case.

If parameters characterising SDS and the search space fulfil conditions

$$p \neq 1,\ p \neq 0,\ a \neq \frac{1}{2},\ a \neq 0\ ,$$

then

$$\begin{aligned} \pi_1 &= \frac{2(1-2a) - \frac{1}{p} + \sqrt{4(1-2a)\left(1-\frac{1}{p}\right) + \frac{1}{p^2}}}{4(1-2a)}, \\ \pi_2 &= \pi_1, \\ \pi_3 &= 1 - 2\pi_1. \end{aligned} \tag{6.1}$$

From (6.1) it follows that the probability of finding an agent activated by one of the patterns, $\pi_1+\pi_2$, is given by

$$\pi_1 + \pi_2 = \frac{2(1-2a) - \frac{1}{p} + \sqrt{4(1-2a)\left(1-\frac{1}{p}\right) + \frac{1}{p^2}}}{2(1-2a)}.$$

From the latter one obtains

$$\begin{aligned} \pi_1 + \pi_2 &= \frac{2(1-2a)p - 1 + \sqrt{4(1-2a)(p-1)p + 1}}{2(1-2a)p} = \\ &= \frac{2(1-2a)p - 1 + \sqrt{4(1-2a)p^2[1-2a+2a] - 4(1-2a)p + 1}}{2(1-2a)p} = \\ &= \frac{2(1-2a)p - 1 + \sqrt{4(1-2a)^2 p^2 - 4(1-2a)p + 1 + 8a(1-2a)p^2}}{2(1-2a)p}, \end{aligned}$$

and the above can be straightforwardly rearranged into the form

$$\pi_1 + \pi_2 = \frac{2(1-2a)p - 1 + \sqrt{[2(1-2a)p - 1]^2 + 4p^2(1-2a)2a}}{2(1-2a)p}.$$

After substitution 2a=p_m^a, and p=1-p_a^- one obtains

$$\pi_a = \frac{2\left(1-p_m^a\right)\left(1-p_a^-\right) - 1 + \sqrt{\left[2\left(1-p_m^a\right)\left(1-p_a^-\right)-1\right]^2 + 4\left(1-p_m^a\right)\left(1-p_a^-\right)^2 p_m^a}}{2\left(1-p_m^a\right)\left(1-p_a^-\right)}. \tag{6.2}$$

Comparing (6.2) with (4.7) from Chapter 4 one can see that the total steady state probability of an activation of an agent on two equally strong patterns is equivalent to the probability of activation on a single best fit pattern with the same overlap with the target as any of the two patterns and the probability of random draw being twice as high as a corresponding probability for the two patterns case.

From the above observation it follows immediately that, in the case of equally strong patterns in the search space, the equilibrium probability of an agent's activation is a linear function of the probability of random draw of the patterns from the search space, i.e.

$$\pi\left(p, p_{d_1}\right) + \pi\left(p, p_{d_2}\right) = \pi\left(p, p_{d_1} + p_{d_2}\right) \tag{6.3}$$

This property will make it possible to extend the analysis of SDS to the case of the best fit pattern and a homogenous noise. The extension will formalise an intuition, supported by the above observation, that in the case of the unique best pattern and a homogenous noise all distractors with equal strength may be replaced by a single distractor with appropriately chosen parameters. Of course, such a reduction will lead to the marginal probability distribution for the best fit pattern and all distractors clumped together, instead of the more general joint probability distribution. The latter could be recovered from the former again using the linearity property characterised by (6.3). However, one can obtain a full characterisation of the steady state behaviour of SDS from the marginal distribution.

Assume that there exists the unique best instantiation of the target in the search space characterised by usual parameters p^-, p_m and k distractors with parameters p^+, p_d. It should be noted, that a more general assumption would assign a distinct probability of random draw, p_d^i, to every distractor. However, due to the linearity property (6.3), it may be assumed without loss of generality, that all p_d^i are equal.

The full nonhomogenous Markov Chain formulation corresponding to a single agent evolution is then given by the following equation

$$(\pi_1, \ldots, \pi_{k+2}) \begin{bmatrix} 1-p^- & 0 & \cdots & 0 & p^- \\ 0 & p^+ & \cdots & 0 & 1-p^+ \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & & p^+ & 1-p^+ \\ p_1 & p_2 & \cdots & p_{k+1} & 1-\sum p_i \end{bmatrix} = (\pi_1, \ldots, \pi_{k+2}), \tag{6.4}$$

where

$$\begin{aligned}
&\sum_{i=1}^{k+2} \pi_i = 1, \\
&p_1 = \pi_1\left(1 - p^-\right) + \left(1 - \sum_{j=1}^{k+1} \pi_j\right) p_m\left(1 - p^-\right), \\
&p_i = \pi_i p^+ + \left(1 - \sum_{j=1}^{k+1} \pi_j\right) p_d p^+, \quad i = 2, \dots, k+1,
\end{aligned} \tag{6.5}$$

and the number of distractors, k, and p_d fulfil the condition

$$p_m + kp_d \le 1.$$

Noting, that from (6.4) one has that

$$p^+\left(\sum_{i=2}^{k+1} \pi_i\right) + \left(\sum_{i=2}^{k+1} p_i\right)\pi_{k+2} = \sum_{i=2}^{k+1} \pi_i,$$

and that from the last equation of (6.5) it follows that

$$\sum_{i=2}^{k+1} p_i = p^+\left(\sum_{i=2}^{k+1} \pi_i\right) + \left(1 - \sum_{i=1}^{k+1} \pi_i\right)\left(kp_d\right)p^+,$$

one can reduce the equation (6.4) to an equation

$$\left(\pi_1, \pi, \pi_{k+2}\right) = \begin{bmatrix} 1 - p^- & 0 & p^- \\ 0 & p^+ & 1 - p^+ \\ p_1 & p & 1 - p_1 - p \end{bmatrix} \begin{pmatrix} \pi_1 \\ \pi \\ \pi_{k+2} \end{pmatrix}, \tag{6.6}$$

where

$$\begin{aligned}
&p_1 = \pi_1\left(1 - p^-\right) + \left[1 - \left(\pi_1 + \pi\right)\right]p_m\left(1 - p^-\right) \\
&p = \pi\, p^+ + \left[1 - \left(\pi_1 + \pi\right)\right]\left(kp_d\right)p^+,
\end{aligned} \tag{6.7}$$

and the following abbreviations were used

$$\begin{aligned}
\pi &= \sum_{i=2}^{k+1} \pi_i, \\
p &= \sum_{i=2}^{k+1} p_i.
\end{aligned}$$

Thus, the model of the search space with a homogenous noise was reduced to the model for a search with the unique best fit and a single distractor described in Chapter 5. Therefore, in order to analyse the behaviour of SDS in the case of a homogenous noise search space one can solve the reduced model equations (6.6)-(6.7) which will determine parameters enabling construction of a marginal probability distribution. The full set of parameters necessary to construct full joint probability distribution of agents in the steady state can be recovered from the solutions to the system (6.6)-(6.7) using linearity property (6.3). Namely, one will have

$$\begin{aligned} \pi_1 &= \pi_1, \\ \pi_i &= \frac{\pi}{k}, \quad i = 2, \dots, k+1 \\ \pi_{k+2} &= \pi_{k+2} \end{aligned}$$

where π_1, π and π_{k+2} are solutions of (6.6)-(6.7).

6.3. *Characterisation of the solutions in terms of the parameters of the model.*

In order to perform qualitative analysis of SDS steady state distribution of agents in a homogenous noise case the Mathematica package was used in order to solve the system of equations corresponding to (6.6)-(6.7) for chosen values of parameters p^- and p_m and to plot the results in Figure 6.1 - Figure 6.1.

Figure 6.1 shows the prediction of the model of the average distribution of the agents between the best fit and distractors. In this case the best fit pattern has a high chance of activating an agent pointing to it, 80% (i.e. p^-=0.2). From Figure 6.1 it follows that even in the case of a homogenous noise search space the distribution of agents is very biased in favour of a strong solution.

This is particularly clear from Figure 6.1 (a), which shows that the average total amount of agents activated by all distractors taken together will be orders of magnitude smaller than the average cluster of agents concentrated on the best fit pattern. One can see that the slope of the surface, as a function of p_d alone, is steep for relatively small values of parameter and becomes fairly flat for remaining values. It follows from the figure that it makes a more significant difference for SDS to switch from noise-free search space to the search space with a small amount of noise, then increasing existing amount of noise by the same amount. The probability of successful activation on a distractor, p^+, seems to have a greater influence on the ability of SDS to discriminate signal from noise. Indeed, one can observe that over a range of values of parameters p^+ and p_d, the slope of the surface decreases faster along lines of constant p^+ than along the lines of constant p_d. This decrease is almost linear until values of p^+

approach p^-, when the slope starts to decrease very steeply.

Figure 6.1 (b) shows the equilibrium probability of activation of an agent on the best fit pattern. As previously, it is equivalent to the average number of agents clustered on the desired solution. Clearly, the fact that the surface remains almost flat for the majority of values of p^+ and p_d indicates a relatively small dependence of the average size of the cluster of agents concentrated on the best fit pattern.

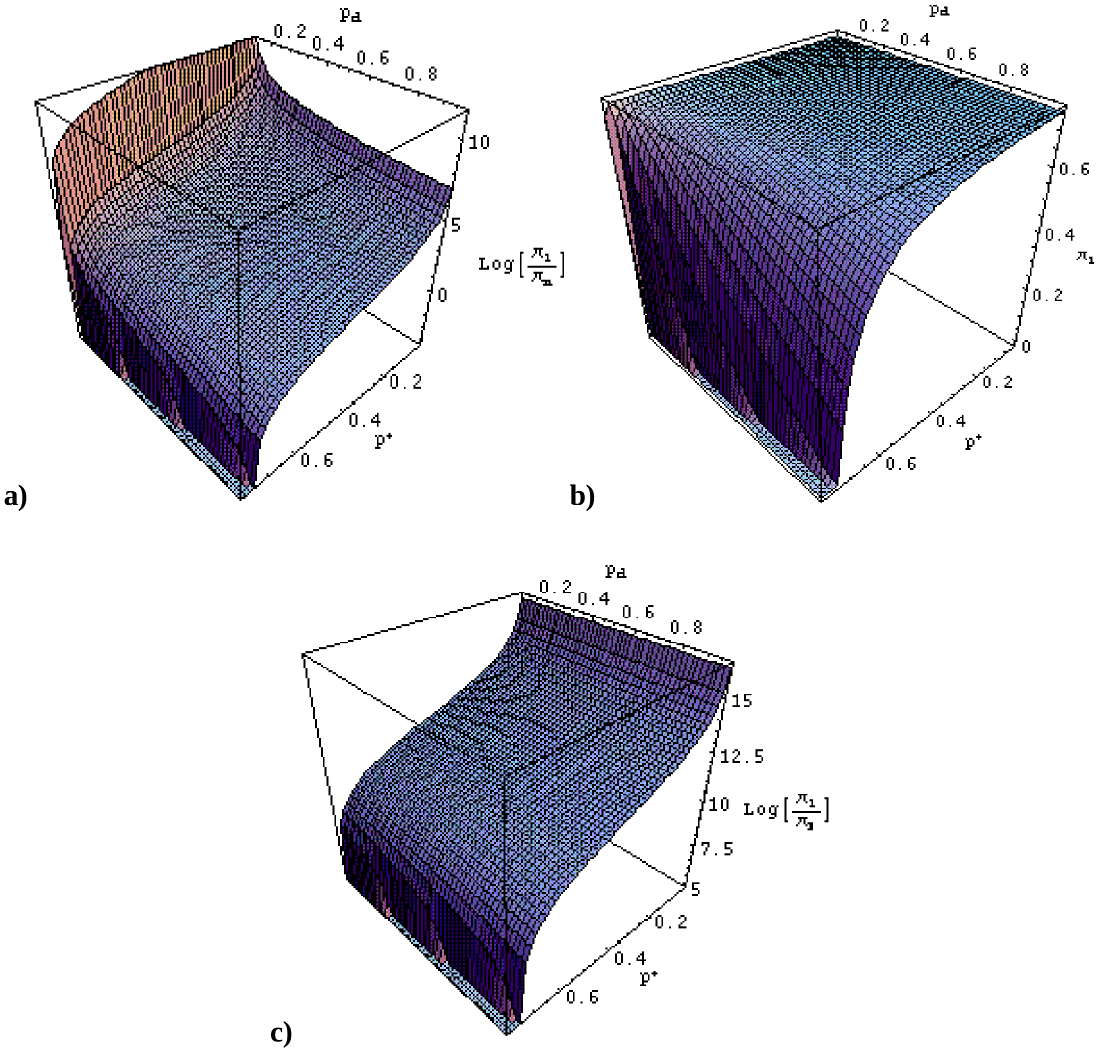


Figure 6.1 (a) A surface describing average orders of magnitude of a difference between the cluster of agents pointing to the correct solution and the total number of agents activated by distractors as a function of the successful test on a distractor, p^+, and a probability of random draw of a distractor from the search space, p_d; (b) an average cluster of active agents pointing to the correct solution; (c) a surface corresponding to average orders of magnitude of a difference between the cluster of agents pointing to the correct solution and a randomly chosen distractor. All cases describe the distribution of agents in the steady state for p^-=0.2 and p_m=0.0001 and p^+<1-p^-.

Comparing both Figure 6.1 (a) and (b) suggests that for the majority of values of p^+ and p_d the relative change of amounts of agents assigned to the best fit pattern and all distractors results from change of the latter, the former being relatively constant. All this implies that the increase of the amount of active agents pointing to distractors follows from the decrease of the amount of inactive agents, rather than from the decrease of the best fit pattern cluster size.

The fast decrease of the size of that cluster in a region of values of p^+ approaching $1-p^-$ agrees with the change of the surface in Figure 6.1 (a) for corresponding values of parameters and suggests increase, in this region, of the coupling between the clusters of agents pointing towards the best fit pattern and towards distractors. Figure 6.1 (c) shows that the average orders of magnitude of a difference between clusters concentrated on the best fit pattern and on any single distractor is very insensitive to p_d - the parameter characterising the amount of noise in the search space. Again for majority of the values of parameters p^+ and p_d this quantity decreases steadily in an almost linear fashion with an increase of p^+ and drops quickly as p^+ approaches $1-p^-$.

It follows from the analysis in Chapter 5 that the value 0.5 of the parameter p^- plays an important role separating in practice regions of the variability of p^- in which SDS shows different behaviour, with the separation of the best fit pattern from the distractor being much more pronounced and clear above 0.5 than below. It is then interesting to see the distribution of agents in SDS for the best fit pattern having this boundary value of p^-. This is illustrated in Figure 6.2.

Figure 6.2 (a) corresponds to Figure 6.1 (a). One can see that for $p^-=0.5$ it is no longer the case that majority of activation of agents in the Stochastic Diffusion Search is due to the best fit pattern alone. In fact, most of the area of the surface in Figure 6.2 (a) lies below plane z=0, i.e. majority of agents are activated by distractors. Also although, the dependence on p_d of the average order of magnitude of difference between the number of agents activated by the best fit pattern and by all of the distractors qualitatively resembles that of Figure 6.1 (a), nevertheless one can notice the higher sensitivity to p_d for high values of the parameter p^+. Also decrease of the surface slope as a function of p^+ alone is faster than in Figure 6.1 (a) and seems to be approximately linear also for the values of p^+ approaching $1-p^-$, contrary to the behaviour observed in Figure 6.1 (a).

From Figure 6.2 (b) it follows that the equilibrium probability of activation of an agent on the best fit pattern is very small for most of the values of parameters p^+ and p_d and shows the relatively small variation in the corresponding region of the parameter space. The dependence of this quantity on the values of p^+ and p_d is most significant for small values of the parameters. In summary, the pattern of a change of an average cluster of agents activated by the best fit pattern as a function of p^+ and p_d is opposite to the one corresponding to the strong solution, as can be observed in Figure 6.1 (b).

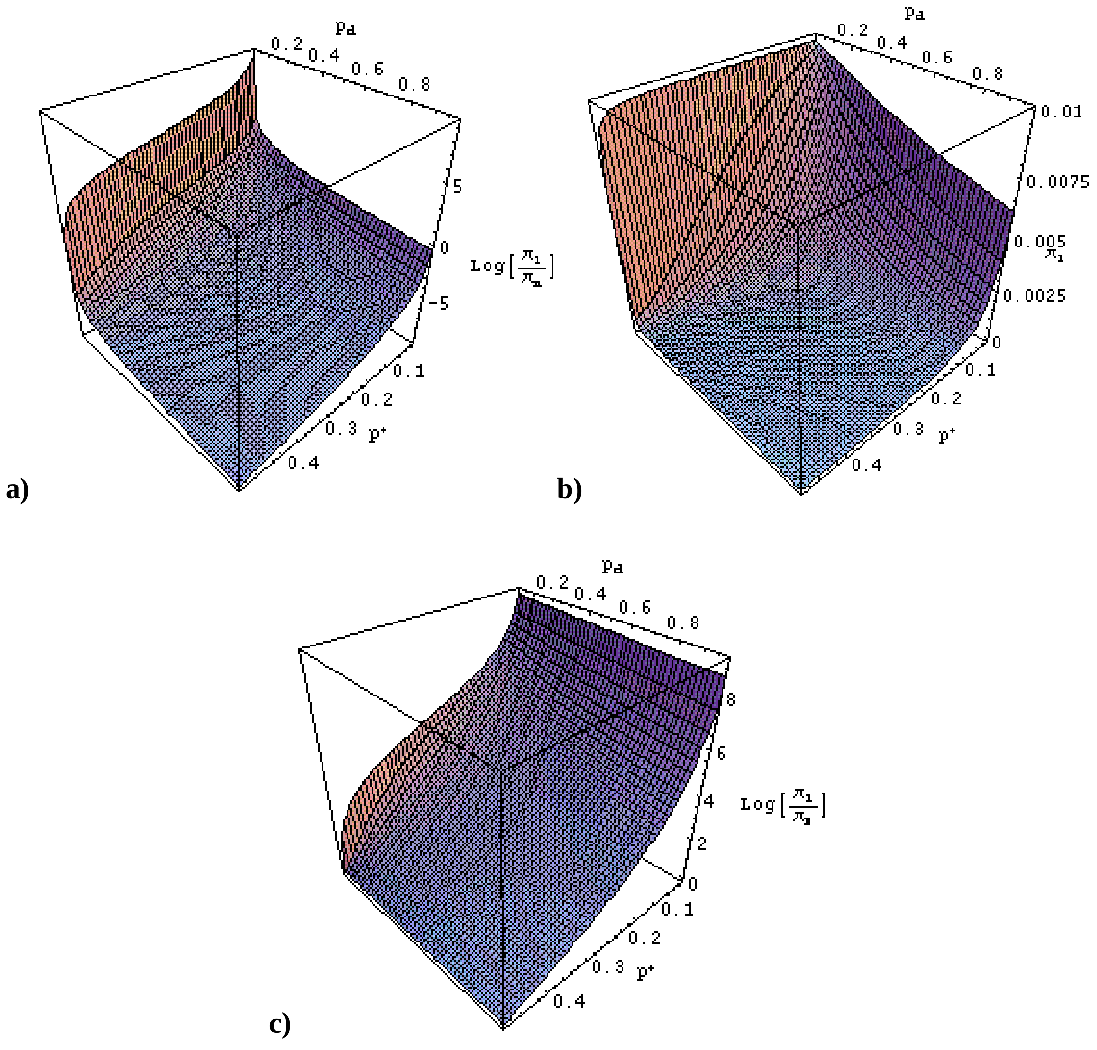


Figure 6.2 (a) A surface describing average orders of magnitude of a difference between the cluster of agents pointing to the correct solution and the total number of agents activated by distractors as a function of the successful test on a distractor, p^+, and a probability of random draw of a distractor from the search space, p_d; (b) an average cluster of active agents pointing to the correct solution; (c) a surface corresponding to average orders of magnitude of a difference between the cluster of agents pointing to the correct solution and a randomly chosen distractor. All cases describe the distribution of agents in the steady state for p^-=0.5 and p_m=0.0001 and p^+<1-p^-.

Although it follows from Figure 6.2 (a) that for $p^-=0.5$ the activity of agents is mostly due to distractors and not due to the best fit pattern alone, Figure 6.2 (c) clearly indicates that the model predicts clear dominance of the cluster of activity on the best fit pattern over the corresponding cluster on an arbitrary distractor. However, it can also be seen that the difference between average orders of magnitude of clusters pointing to the best fit pattern and to any distractor decreases approximately linearly as p^+ approaches $1-p^-$, which should be contrasted with a rapid nonlinear decrease in Figure 6.1 (c).

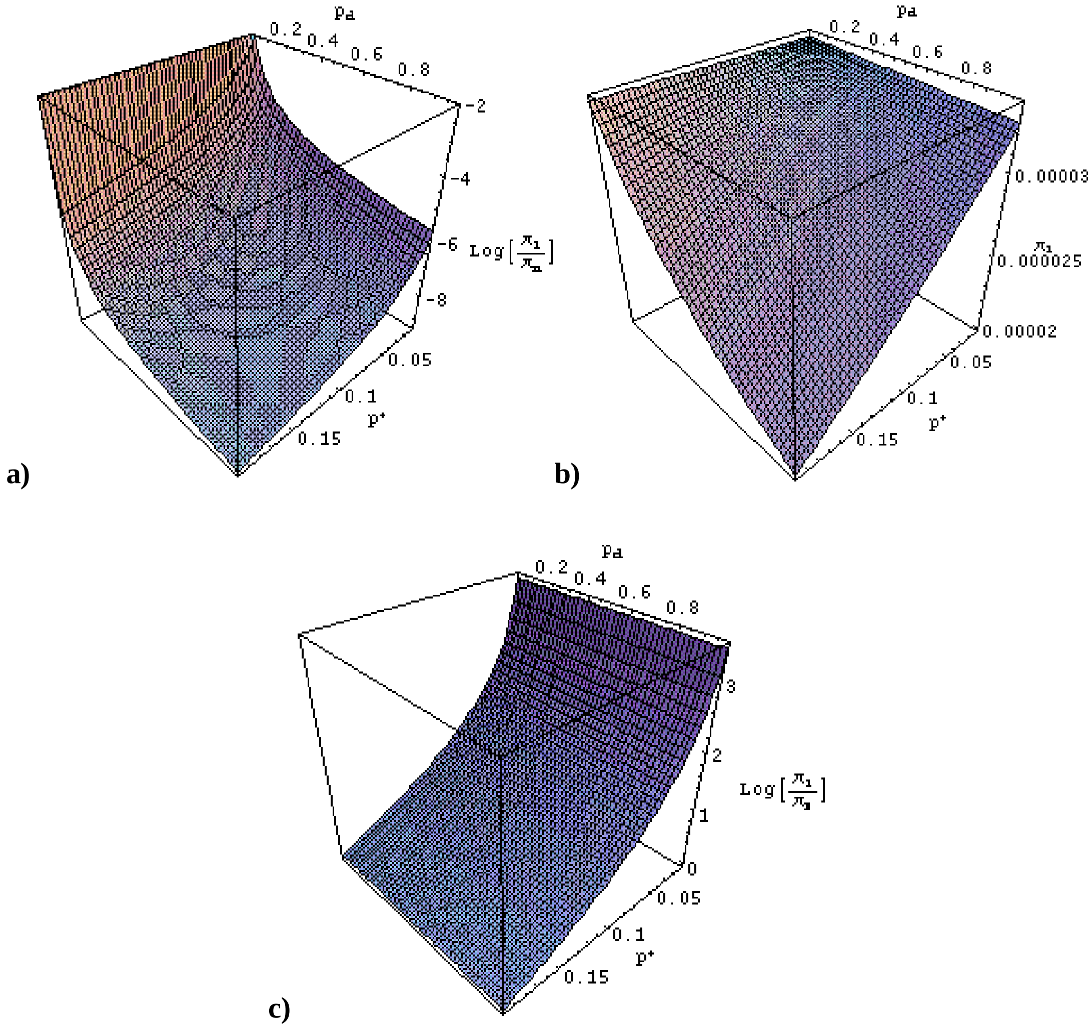


Figure 6.3 (a) A surface describing average orders of magnitude of a difference between the cluster of agents pointing to the correct solution and the total number of agents activated by distractors as a function of the successful test on a distractor, p^+, and a probability of random draw of a distractor from the search space, p_d; (b) an average cluster of active agents pointing to the correct solution; (c) a surface corresponding to average orders of magnitude of a difference between the cluster of agents pointing to the correct solution and a randomly chosen distractor. All cases describe the distribution of agents in the steady state for $p^-=0.8$ and $p_m=0.0001$ and $p^+<1-p^-$.

Finally Figure 6.3 shows an equilibrium distribution of agents in the case of a weak solution, p^-=0.8. From Figure 6.3 (a) it follows that the average difference of orders of magnitude between the best fit cluster and the total activity caused by the noise follows qualitatively similar pattern of dependence on parameters p^+ and p_d as observed in Figure 6.2 (a). The main difference is that in the case of a weak best solution, the number of agents activated on the distractors is greater than the size of the cluster pointing to the best fit pattern for practically all values of parameters p^+ and p_d.

Figure 6.3 (b) confirms that in this case the average amount of agents pointing to the best fit pattern constitutes a very small ratio of the total amount of agents. It reveals also that the surface describing this amount as a function of two parameters decreases practically linearly along lines of constant p_d, i.e. it is approximately linearly dependent on the probability of successful activation of an agent on a distractor, p^+.

The decrease resulting from the increase of p_d alone is also quite well approximating linear function. The very small range of variability of π_1 across all values of p^+ and p_d suggests that the distribution of agents depicted in Figure 6.3 (b) is qualitatively similar to the one corresponding to the flat part of the surface from Figure 6.2 (b).

One can also note the similarity of Figure 6.3 (c) to Figure 6.2 (c), except an expected change of a range of variability of the average relative orders of magnitude and slightly higher sensitivity to changes in p_d for small values of this parameter observed in Figure 6.2 (c).

The extension of the model from Chapter 5 will allow us also to address an issue of dissipativity of SDS. The condition (5.11) can be used and, in the case of a homogenous noise in the search space, it reads

$$4p^+\left(1-p^-\right)\left[1-\left(p_m+kp_d\right)\right]\leq 1\,. \tag{6.8}$$

In the inequality (6.8) the absolute value present in (5.11) could be abandoned due to the condition spelled under the equation (6.5).

It is interesting to note that although in Chapter 5 one could neglect the last term of the left hand side of the inequality (5.11) and obtain equivalent of the condition discussed after the proposition 4.3 in Chapter 4, however this step cannot be repeated in the current case. This is because, in general, one can expect that the number of distractors is significant, so the last term of the right hand of (6.8) cannot be approximated by unity. On the contrary, one should expect this term to be significantly smaller than 1. This means that (6.8) could be fulfilled for much higher values of the parameters p^- and p^+ than the corresponding inequality (5.11). This leads to a very interesting conclusion. It follows from (6.8) that noise is, surprisingly, useful for the operation of SDS in that it forces the system to behave in a dissipative manner.

6.4. *Simulations.*

In this section several simulations of the Stochastic Diffusion Search will be presented illustrating some of the points discussed in the preceeding sections.

In all the simulations 1000 agents of the Stochastic Diffusion Search performed searches for the unique best fit pattern. In all cases there was a number of weaker patterns, distractors, inserted in random positions in the search space. The amount of distractors, characterised by the parameter p_d, their overlap with the target pattern, p^+, and the best fit pattern, p^- were varied. The evolution of SDS was recorded for 5000 iterations. All the figures below present data from last 4500 iterations, the first 500 being discarded to diminish bias in estimates. In all the simulations the quantities of interest are the steady state behaviour of clusters of agents supporting the best fit solution, a randomly chosen distractor and a whole population of distractors. Thus, all the figures contain graphs reflecting these quantities.

The assumption of a homogenous noise imposes constraints on variability of parameters p^+ and p_d. Namely, increasing the overlap between a distractor and the target template, i.e. increasing p^+, enforces decrease of the parameter p_d characterising the amount of noise. Without this precaution one would not be able to maintain homogeneity of the noise. However, in all the simulations the value of p_d was kept close to the maximal allowable for a given value of p^+.

Figure 6.4 - Figure 6.6 show a typical behaviour of SDS searching for the best fit pattern with a high overlap with the template and varying overlaps of distractors constituting a homogenous noise. In all these figures p^- is constant and set to 0.2 while the strength and the amount of distractors are varied.

Figure 6.4 shows the steady state distribution of agents in the case of a relatively weak noise (p^+=0.2, p_d=0.993). One can see that the cluster of agents concentrated on the best fit solution is very stable - Figure 6.4 (a), and more than 40 times bigger than the total population of agents activated by distractors, as shown in Figure 6.4 (c). Also the size of the latter cluster varies to a much higher extent than the former (relative variability of about 26% as contrasted

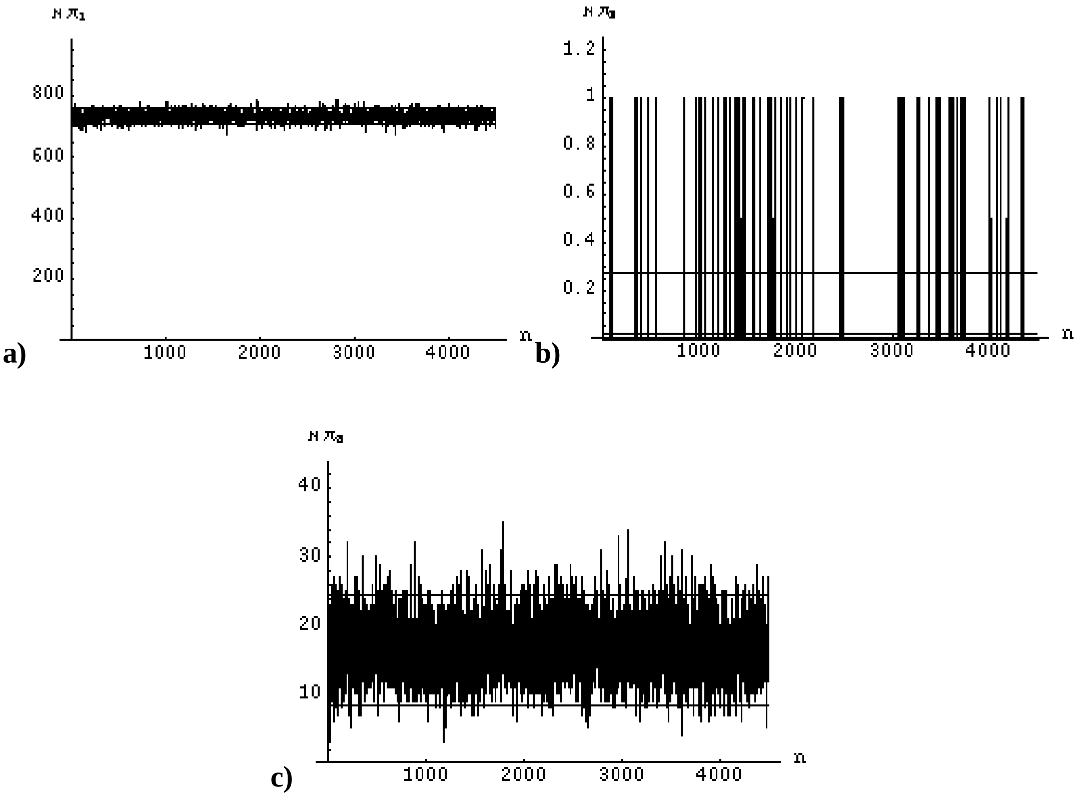


Figure 6.4 Evolution of SDS in the homogeneously noisy search space over 4500 iterations. The best fit pattern has a high overlap with the target, $p^-=0.2$; $p_m=0.001$, N=1000, $p^+=0.2$, $p_d=0.993$; (a) the evolution of the cluster associated with the best fit pattern (its expected size is 733.547 and a standard deviation is 13.98 and a sample mean is 733.158 and a sample standard deviation 15.447); (b) the evolution of the cluster associated with a distractor (its expected size is 0.0167 and a standard deviation is 0.129 and a sample mean is 0.019 and a sample standard deviation 0.137); (c) the evolution of the cluster associated with all distractors (its expected size is 16.54 and a standard deviation is 4.03 and a sample mean is 16.68 and a sample standard deviation 4.31).

with corresponding 2% for the best fit pattern cluster). However, in spite of the higher variability it is clear that the total population of agents activated by noise is quite stable. This can be contrasted with Figure 6.4 (b) showing a cluster of agents activated by a distractor. It is clear that, for the given parameters, the cluster of agents activated by a distractor is very small and highly unstable. In fact, most of the time there are no active agents associated with a given distractor. One can also note a very good agreement between predictions of the model and sample estimates of the distribution of active agents.

From a comparison of Figure 6.4 -Figure 6.6 one can conclude that in all of them a similar characteristic, to the one described for Figure 6.4, can be observed. In particular, the simulations are in a good agreement with the predictions of the model.

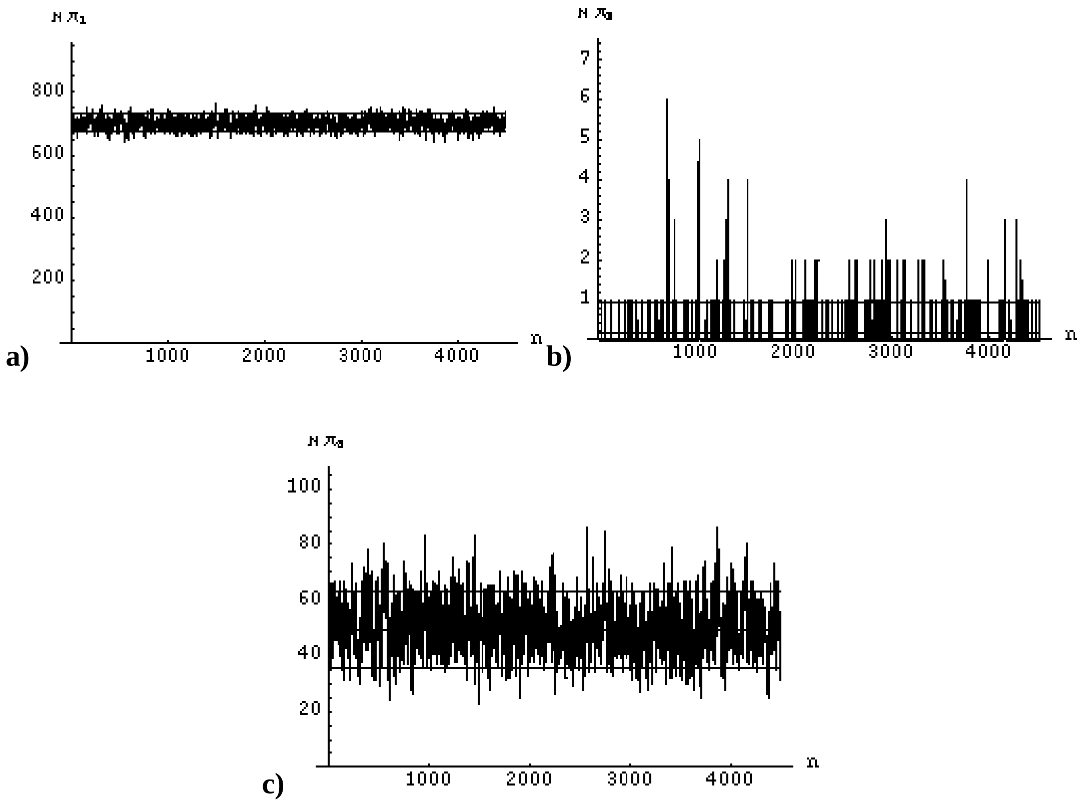


Figure 6.5 Evolution of SDS in the homogeneously noisy search space over 4500 iterations. The best fit pattern has a high overlap with the target, p^{-}=0.2; p_m=0.001, N=1000, p^{+}=0.6, p_d=0.331; (a) the evolution of the cluster associated with the best fit pattern (its expected size is 700.485 and a standard deviation is 14.48 and a sample mean is 699.534 and a sample standard deviation 17.33); (b) the evolution of the cluster associated with a distractor (its expected size is 0.15 and a standard deviation is 0.387 and a sample mean is 0.149 and a sample standard deviation 0.485); (c) the evolution of the cluster associated with all distractors (its expected size is 49.604 and a standard deviation is 6.87 and a sample mean is 50.13 and a sample standard deviation 9.13).

From these figures it also follows that increasing the overlap of distractors with the target template results in very slow decrease of the best pattern cluster size, even though in Figure 6.6 distractors differ from the best fit pattern only by 12.5%.

Although, from comparison of the distractor cluster sizes across all three figures one could conclude their significant increase from the case depicted in Figure 6.4 (b) to that in Figure 6.6 (b), in practise in all these cases most of the time there is at most one active agent attending to a distractor.

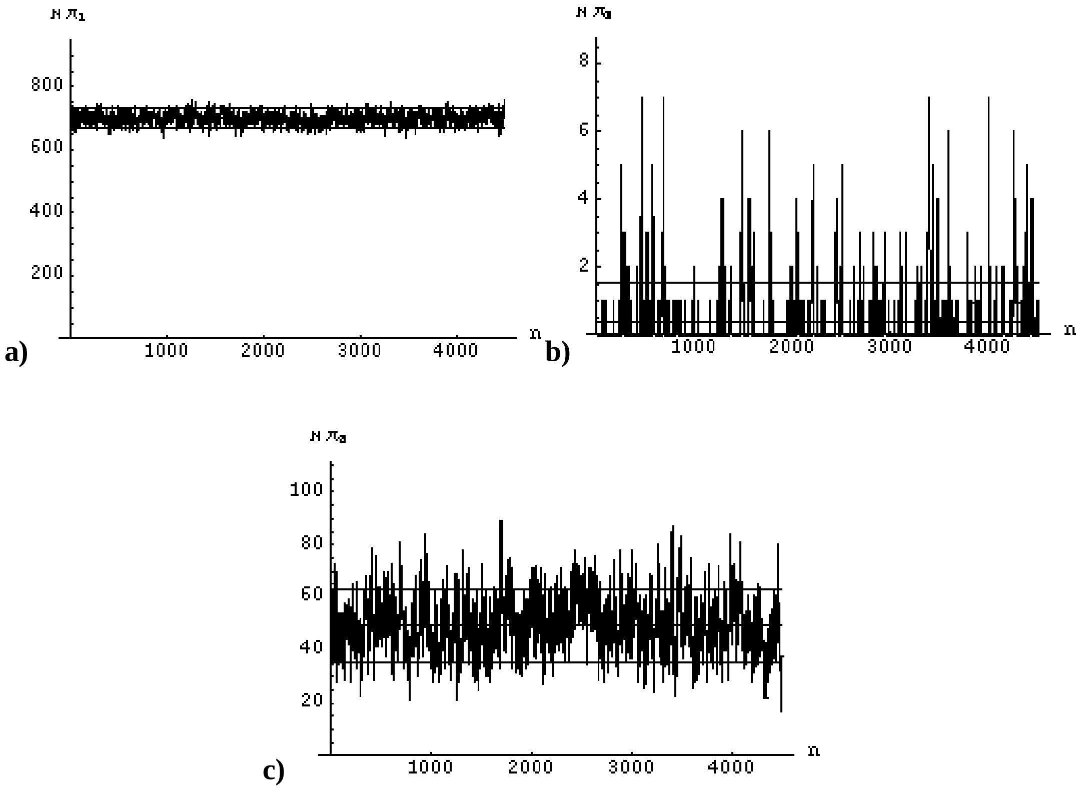


Figure 6.6 Evolution of SDS in the homogeneously noisy search space over 4500 iterations. The best fit pattern has a high overlap with the target, p^-=0.2; p_m=0.001, N=1000, p^+=0.7, p_d=0.141; (a) the evolution of the cluster associated with the best fit pattern (its expected size is 700.799 and a standard deviation is 14.48 and a sample mean is 699.722 and a sample standard deviation 17.69); (b) the evolution of the cluster associated with a distractor (its expected size is 0.35 and a standard deviation is 0.59 and a sample mean is 0.4 and a sample standard deviation 0.92); (c) the evolution of the cluster associated with all distractors (its expected size is 49.29 and a standard deviation is 6.845 and a sample mean is 49.77 and a sample standard deviation 10.78).

The increase of the parameter p^+, seems rather to be reflected in an increased probability of encountering more than one active agent pointing to a given distractor at the same time (no such events were observed in Figure 6.4 (b), whereas they can be observed quite frequently in Figure 6.6 (b)). However, it seems that the increased overlap of the distractors with the template have a noticeable influence on the total amount of the agents activated by noise. Although, in the case when the total amount of agents in this cluster is highest, it still constitutes only a small fraction of the best fit pattern cluster size (about 7%), nevertheless it increased about 3-fold from Figure 6.4 (c) to Figure 6.6 (c), even in spite of the fact that the total amount of distractors actually decreased considerably (as characterised by a decrease of p_d from 0.993 to 0.141). In summary Figure 6.4 - Figure 6.6 illustrate the fact that SDS has a very non-linear gain amplification property - it transforms small signal to noise (best fit to a distractor) ratio into a high ratio of the corresponding cluster sizes.

Figure 6.7 and Figure 6.8 show the distributions of agents in the case of a relatively weak best

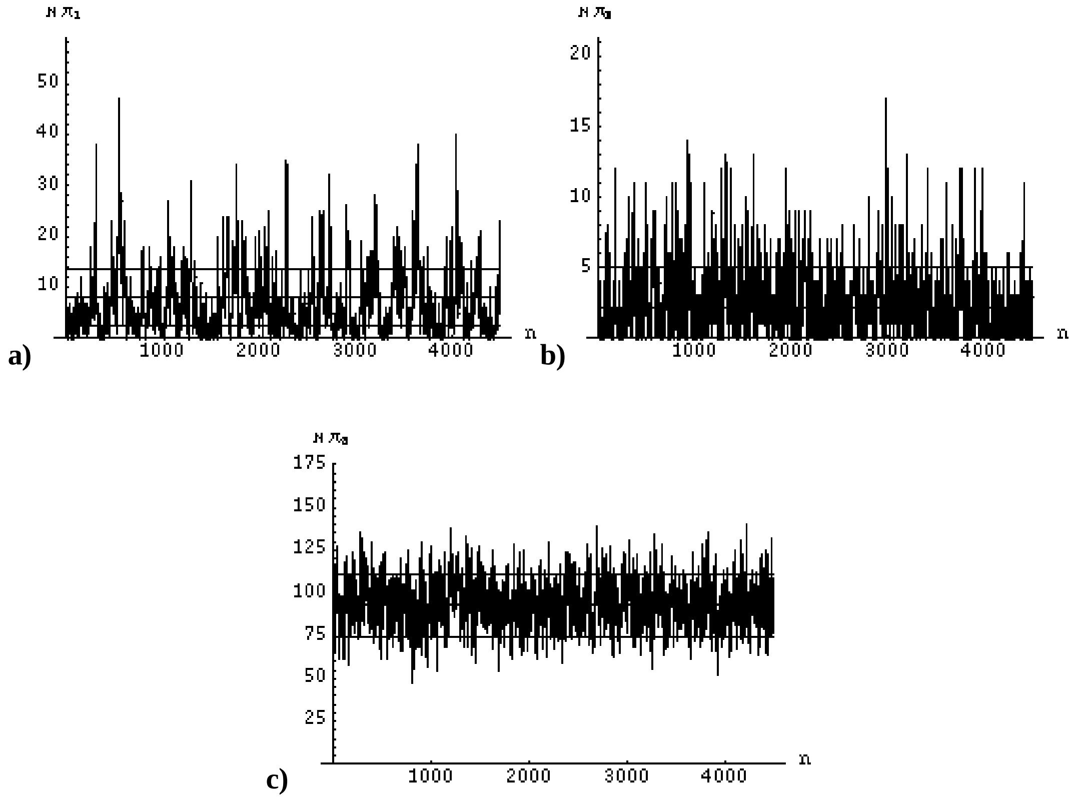


Figure 6.7 Evolution of SDS in the homogeneously noisy search space over 4500 iterations. The best fit pattern has a relatively weak overlap with the target, p^{-}=0.5; p_m=0.0001, N=1000, p^{+}=0.44, p_d=0.043; (a) the evolution of the cluster associated with the best fit pattern (its expected size is 8.0 and a standard deviation is 2.82 and a sample mean is 6.96 and a sample standard deviation 6.51); (b) the evolution of the cluster associated with a distractor (its expected size is 2.16 and a standard deviation is 1.47 and a sample mean is 2.4 and a sample standard deviation 2.43); (c) the evolution of the cluster associated with all distractors (its expected size is 92.99 and a standard deviation is 9.18 and a sample mean is 93.46 and a sample standard deviation 13.3).

fit pattern, p^{-}=0.5.

In Figure 6.7 one can observe a steady state distribution of SDS agents for distractors characterised by p^{+}=0.44 and p_d=0.043. In panel (a) the best fit pattern cluster is small and is no longer stable - in some iterations it disappears and its variation is quite high. One can also note that although the expected value of a size of this cluster and its sample mean are relatively close, however the sample standard deviation is still more than two times higher than the one predicted by the model. In spite of this, the best fit pattern cluster is still almost three times as big as the cluster corresponding to a distractor - Figure 6.7 (b). In panel (b) one can see that the cluster activated by a distractor disappears even more frequently than the one corresponding to the best fit pattern.

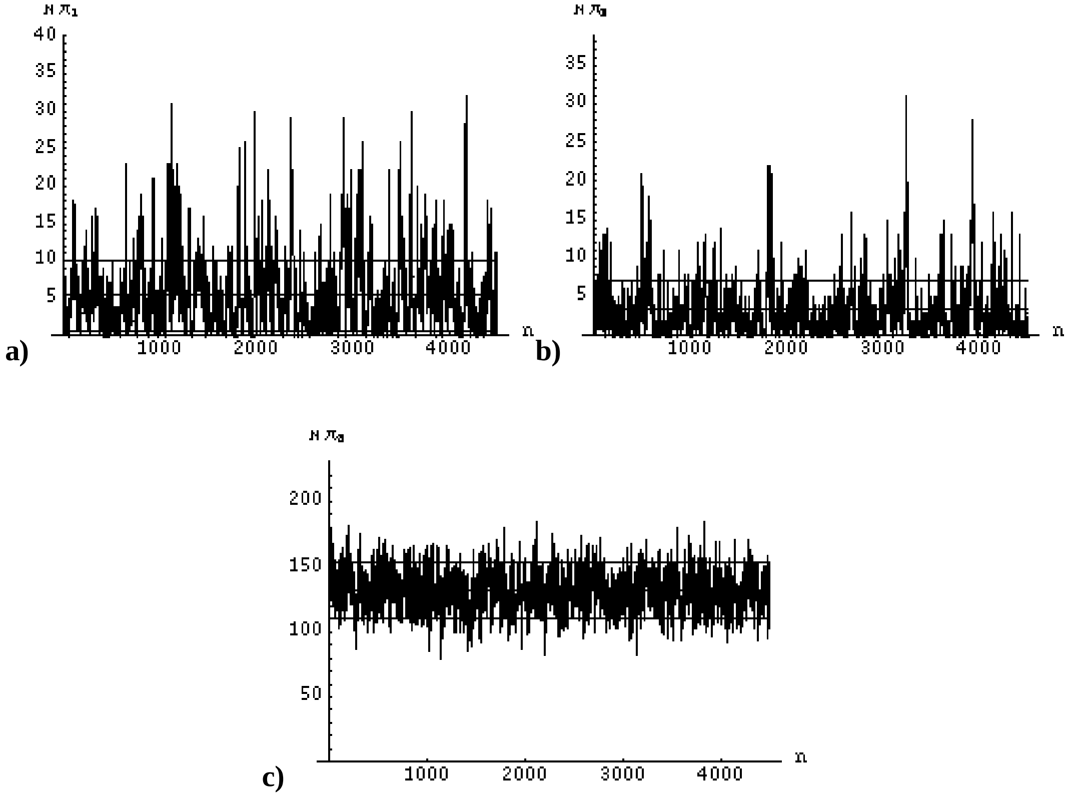


Figure 6.8 Evolution of SDS in the homogeneously noisy search space over 4500 iterations. The best fit pattern has a relatively weak overlap with the target, p^{-}=0.5; p_m=0.0001, N=1000, p^{+}=0.48, p_d=0.039; (a) the evolution of the cluster associated with the best fit pattern (its expected size is 5.43 and a standard deviation is 2.32 and a sample mean is 5.68 and a sample standard deviation 5.24); (b) the evolution of the cluster associated with a distractor (its expected size is 3.38 and a standard deviation is 1.83 and a sample mean is 3.18 and a sample standard deviation 3.64); (c) the evolution of the cluster associated with all distractors (its expected size is 131.71 and a standard deviation is 10.69 and a sample mean is 130.97 and a sample standard deviation 15.4).

From Figure 6.7 (c) it follows that the population of agents activated by all of the distractors is stable and quite well fits to the predictions of the model. Clearly, it is the biggest of all three clusters.

From a comparison of Figure 6.8 with Figure 6.7 one can see that increasing the overlap of distractors with the template in practise has the biggest impact on the total population of agents activated by noise. Overall the distribution of agents shown in the last figure shares characteristics of the distribution depicted in Figure 6.7.

Contrasting Figure 6.4 - Figure 6.6 with Figure 6.7 and Figure 6.8 one can note that in the latter the clusters corresponding to the best fit pattern and a distractor are qualitatively similar, whereas their characteristic differs considerably in Figure 6.4 - Figure 6.6. Also, although the former cluster is still bigger than the latter, the difference is not so significant as in Figure 6.4 - Figure 6.6 and is much smaller than the total number of agent activated by noise.

It is worth noting that in all cases, simulations fit well to the predictions of the model.

6.5. *Conclusions.*

This chapter generalised the results from Chapter 5 by observing first that, in the case of two best fit patterns, the steady state probability of finding an agent activated by any of the patterns is equal to the equilibrium probability of activation on a single best fit solution whose probability of random sampling from the search space is equal to the sum of the corresponding probabilities for the two patterns. This made possible a straightforward generalisation of the model, taking into account homogenous noise. The model led to a qualitative analysis of the equilibrium resource allocation management of SDS, which showed that it possessed an excellent signal detection characteristics due to a non-linear gain amplification mechanism, the latter being highly sensitive to the signal-to-noise ratio. The subsequent simulations illustrated above mentioned characteristics of SDS and confirmed that the model predicted very well the steady state behaviour of the Stochastic Diffusion Search.

CHAPTER 7

APPROXIMATION OF THE SOLUTION TO THE EXTENDED EHRENFEST URN MODEL

7.1. *Introduction.*

Chapter 4 defined and solved analytically the extended Ehrenfest Urn model of the Stochastic Diffusion Search. Chapter 5 contained an extension of the model to the case of the search space with the best fit pattern and one distractor. However, although analytically solvable in principle, the model of Chapter 5 produced solutions which were very complex functions of the parameters characterising the search conditions and therefore inconvenient for inspection. This chapter will introduce an approximation of the model from Chapter 5 based on the results of Chapter 4.

In Section 7.2 the analysis of the steady state distribution of agents in the case of a unique solution in the search space will be re-examined. The results of this section will be used in Section 7.3 where the approximate solution will be derived analytically and consequently, in Section 7.4, compared against the exact model. The next section will illustrate the predictions of the exact model and approximation by comparing them with simulations of SDS for chosen values of parameters.

It will appear that, for a wide range of parameters characterising the search conditions, the approximate scheme agrees well with the theoretical model. In regions where the quantitative predictions of the approximation are poor, it nevertheless describes well the qualitative trend of changes of SDS under variation of parameters.

7.2. *Analysis of the distribution of agents in case of the unique solution in the search space.*

The basis for the extension is the observation that in the steady state the most common state in which SDS will reside is the quasi equilibrium. This implies that the solution will attract, practically a stable population of agents, which therefore cannot participate in exploration of the search space. On the other hand, the quasi equilibrium state is stable over long periods of time, which implies that the population supporting the solution cannot either decrease or increase. From all of these considerations it follows that in the steady state one would expect the existence of the dynamic population of inactive agents, which will in effect act as if the solution was not present in the search space. This is because this population constitutes a complement of the quasi stable cluster supporting the best solution.

However, if one had removed the solution from the search space then the single distractor would have automatically became the best current solution. Thus one can replicate the reasoning from the proceeding paragraph and conclude that, in the presence of the best solution, the distractor will attract some proportion of agents pointing to it as a potential solution (in fact false positive). The size of this cluster will be determined by the size of the population of “free” agents and by the “strength” of the distractor.

Thus in order to assess the proportion of agents that will be concentrated on the distractor, it is necessary to estimate how many agents will be capable of exploring freely the search space. From the results reported in Chapter 4 one would be tempted to use the difference between the total number of agents and the estimated number of agents pointing to the best solution as an estimate of the “free” agent’s population size. However, the effective number of agents exploring other positions will be smaller than that implied by this quantity. This is because even though the quasi equilibrium is a good estimation of the average steady state of the system, nevertheless the dynamical character of the cluster supporting the best solution implies that there will be a number of agents which will be inactive but nevertheless pointing to the position of the best solution (e.g. agents which were active but lost activity as a result of failed test). These agents will not participate in creation of a cluster supporting the false positive solution. This pool of inactive agents locked to the best solution will have qualitatively the same dynamical characteristics as that of the cluster of active agents. Namely, the identities of individual agents contributing to the pool will change constantly with agents in perpetual movement in and out of the pool. However, the pool as a whole will be quite stable, subject to inevitable fluctuations.

In order to formalise the above ideas it is necessary to consider an extension of the Markov Chain model of the evolution of a single agent. In order to do so one needs to extend the notation used in the previous chapters. The state space used in Chapter 4 will be extended by splitting the inactive state into two states: ‘inactive pointing to the best solution’ state, denoted (n,m), and ‘inactive pointing elsewhere’ state, denoted (n,*). Thus the evolution of an agent is described by the three dimensional stochastic matrix P

$$P = \begin{array}{c|ccc} & (a,m) & (n,m) & (n,*) \\ \hline (a,m) & 1-p^- & p^- & 0 \\ (n,m) & p_1 & p_2 & 1-p_1-p_2 \\ (n,*) & p_1 & p_2 & 1-p_1-p_2 \end{array} \tag{7.1}$$

where

$$\begin{aligned} p_1 &= \tfrac{m}{N}\left(1-p^-\right)+\left(1-\tfrac{m}{N}\right)p_m\left(1-p^-\right), \\ p_2 &= \tfrac{m}{N}p^- + \left(1-\tfrac{m}{N}\right)p_m p^-, \end{aligned} \tag{7.2}$$

and m is the number of active agents, and N - the total number of agents used in the Stochastic Diffusion Search.

Note, that from (7.2) it follows that

$$1-p_1-p_2 = \left(1-\tfrac{m}{N}\right)\left(1-p_m\right) \tag{7.3}$$

The following equation for the equilibrium probability distribution will be solved

$$\begin{aligned} &\pi P = \pi, \\ &\pi = \left(\pi_1,\pi_2,\pi_3\right) \end{aligned}$$

This can be written as a system of three linear equations

$$\begin{cases} \pi_1\left(1-p^-\right)+\pi_2 p_1+\pi_3 p_1 = \pi_1 \\ \pi_1 p^- + \pi_2 p_2 + \pi_3 p_2 = \pi_2 \\ \pi_2\left(1-p_1-p_2\right)+\pi_3\left(1-p_1-p_2\right) = \pi_3 \end{cases} \tag{7.4}$$

with additional constraints $\sum \pi_i = 1, \pi_i \geq 0, \quad i = 1, 2, 3$.

The first two equations of (7.4) yield

$$\pi_2\left(p_1+p_2\right)+\pi_3\left(p_1+p_2\right) = \pi_2,$$

which leads to

$$\pi_2 = \frac{p_1+p_2}{1-\left(p_1+p_2\right)}\pi_3.$$

Thus one obtains two equations

$$\begin{cases} \pi_1 p^- + \pi_2 p_2 + \pi_3 p_2 = \pi_2 \\ \pi_2 = \dfrac{p_1 + p_2}{1 - (p_1 + p_2)} \pi_3 \end{cases}$$

From the last set of equations one gets

$$\pi_1 = \frac{p_1}{p^- [1 - (p_1 + p_2)]} \pi_3$$

So the first two components of the equilibrium probability vector were expressed in tems of the last component. This, together with the normalisation $\sum \pi_i = 1$, enables one to calculate explicitly all the components of the stationary probability distribution vector

$$\begin{aligned} \pi_1 &= \frac{p_1}{p_1 + p^-} \\ \pi_2 &= \frac{p^-(p_1 + p_2)}{p_1 + p^-} \\ \pi_3 &= \frac{p^-[1 - (p_1 + p_2)]}{p_1 + p^-} \end{aligned} \tag{7.5}$$

The extended model is, as previously, an inhomogenous Markov Chain. The expressions in (7.5) still contain p$_1$ and p$_2$ which change during evolution of SDS. In order to obtain final steady state solutions it is necessary to use equation (7.5) in a manner analogous to calculation of the equilibrium probability distribution from Chapter 4. However, one can observe that both the equation for π_1 in (7.5) and the equation for p$_1$ in (7.2) are exactly the same as in that chapter. This is a result of the fact that the procedure of splitting of the inactive state into two distinct ones did not affect the active state, and so did not change the probabilities of moving into this state. From this one obtains exactly the same expression for π_1 as calculated in Chapter 4 and given in equation (4.7). Thus, all the equilibrium probabilities can be expressed using the equation (4.7) for π_1 and (7.5).

Summing up the second and third components of the equilibrium probability vector (7.5) leads to the solution corresponding to the stationary probability of the inactive state from Chapter 4. Thus, the previous equilibrium probability of inactive state is divided into probabilities of ‘inactive pointing to the best solution’ and ‘inactive pointing elsewhere’ states in proportion

$$(p_1 + p_2, 1 - p_1 + p_2)$$

which, using (7.3), reads

$$\left(1-(1-\pi_1)(1-p_m),(1-\pi_1)(1-p_m)\right).$$

Taking into account that the proportion of all the inactive agents in the entire population is $1-\pi_1$, one gets finally that the distribution of the inactive agents between 'inactive and pointing to the best solution' and 'inactive pointing elsewhere' states is given by

$$\left(-(1-p_m)(1-\pi_1)^2+(1-\pi_1),(1-p_m)(1-\pi_1)^2\right) \tag{7.6}$$

This means that the proportion of $(1-\pi_1)-(1-p_m)(1-\pi_1)^2$ of all the agents is, on average, permanently "locked" onto the best solution's position. In intuitive terms this amount of agents provides a buffer which lowers the rate of an outflow, resulting from random fluctuations, of the active agents from the cluster. From this point of view this pool of inactive agents is desirable because it contributes to the stability of the cluster pointing to the best solution. However, existence of this pool implies that the actual proportion of the inactive agents free to explore the rest of the search space is equal on average $(1-p_m)(1-\pi_1)^2$ and not $(1-\pi_1)$, as one might conclude from Chapter 4. Figure 7.1 illustrates the dependence of the pool's size and the size of the "free" agents population on the "strength" of the best solution p^-.

From the panel (a) of Figure 7.1 one can see that the pool of inactive agents increases rapidly with p^- until it reaches a maximal value and then decreases rapidly with p^- increasing further to around 0.5. The marked difference between the dependence on p^- between both halves of the

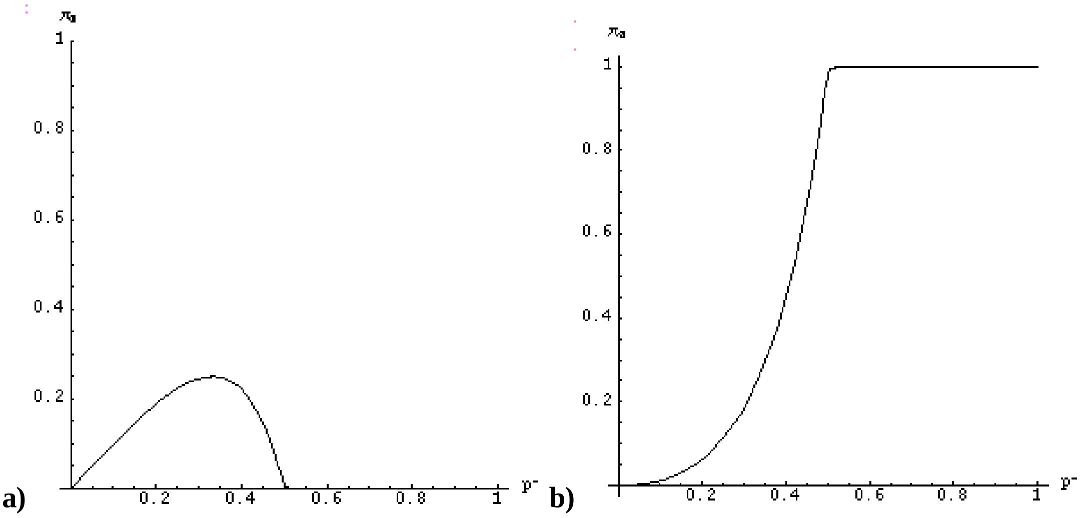


Figure 7.1 (a) The normalised population of the inactive agents 'locked' to the best solution, π_2; (b) the population of agents free to explore the search space, π_3, as functions of the best solution "strength", p^-; $p_m=0.0001$.

unit interval is the range of values attained, with the values in the second half being orders of magnitude smaller than those in the first half (this is the reason for which the graph looks as if it were identically zero for $p^{-}>0.5$). Another difference is the monotonous decrease of the graph in the panel (a) to 0 for p^{-} increasing from 0.5 to 1. The population of "free" agents increases nonlinearly to values very close to 1 when p^{-} increases from 0 to around 0.5 and then levels off with the very small increase to 1 on the second half of the interval.

It will be illustrative to see an example of how big is the pool size in the case of N=1000 agents for the search space with $p_m=0.0001$ and $p^{-}=0.2$. In this case the amount of active agents is the same as in the example from Chapter 4, i.e. E[n]=750. The amount of inactive agents locked to the best solution is on average 187.5 and the pool of "free" agents is about 62.5.

One can see from this example that if one used the number of inactive agents to assess the pool of agents ready to explore the search space, one would overestimate its size about 4 times. However, by changing the "strength" of the solution to $p^{-}=0.4$, one gets, all other parameters constant, E[n]=333.5 of active agents, an average amount of 222.3 inactive agents pointing to the correct position and 444.2 "free" agents. Thus, in this case using the total number of inactive agents would result in an overestimation by a factor of 1.5. Figure 7.2 shows the general tendency of the ratio of the total number of inactive agents to the number of

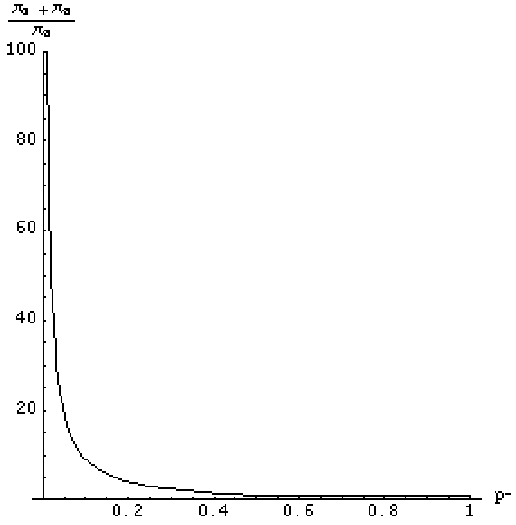


Figure 7.2 The ratio of the total number of inactive agents, $\pi_2+\pi_3$, to "free" agents, π_3 as a function of p^{-}; $p_m=0.0001$.

“free” inactive agents.

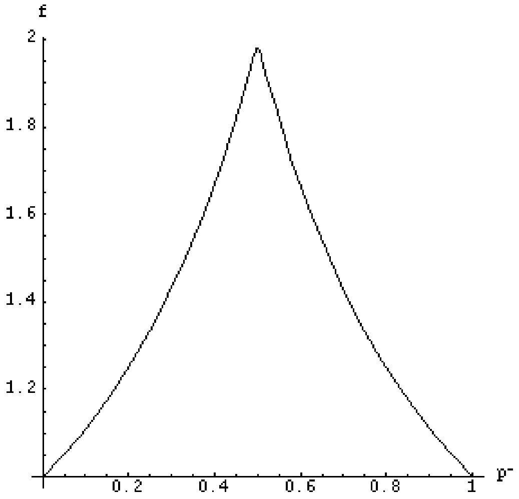


Figure 7.3 The curve describing ratio between inverse proportionality and the ratio of the total number of inactive agents to the number of “free” inactive agents as a function of p^-; p_m=0.0001.

This ratio seems to be inversely proportional to p^-. However, there is a systematic difference between inverse proportionality and the ratio of the total number of inactive agents to the number of “free” inactive agents, as shown in Figure 7.3. From this figure it is clear that the relative difference between the ratio of the total number of inactive agents to the number of “free” inactive agents as a function of p^- and the inverse function is symmetrical in the unit interval around the value of 0.5. It increases in the first half and then decreases in the second half of the unit interval.

From the above one can see that in order for the Stochastic Diffusion Search to be able to explore effectively the search space, the total number of agents has to fulfil the inequality

$$N\pi_3 \geq k \ ,$$

which, taking into account (7.6), gives a lower bound for N

$$N \geq N_{\min} = \frac{k}{(1-\pi_1)^2 (1-p_m)} \tag{7.7}$$

where π_1 is given in (4.7) and k is an arbitrary number of agents supposed to search for alternative solutions.

The condition (7.7) ensures that there will be on average at least k agents able to explore other positions in the search space than the best solution's position.

If the number of agents falls below the expression on the right hand side of (7.7) for k=1, then effectively there will be very long periods of time when no agents will be involved in exploration of the search space.

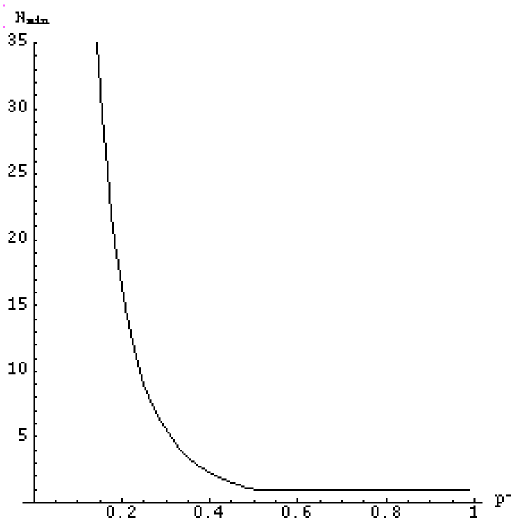


Figure 7.4 The minimal number of agents in SDS, N_{min}, necessary to support active exploration of the search space by "free" agents as a function of p^-; N=1000, p_m=0.0001, k=1.

Figure 7.4 shows the curve describing the minimal total number of agents used in the Stochastic Diffusion Search necessary for an exploration to take place.

The observations concerning the distribution of agents made in this section will be used in the next one in order to calculate an approximate allocation of resources in the case of two solutions in the search space.

7.3. Approximation of the steady state probabilities in the extended Ehrenfest Urn Model

The characterisation of the "free" agents population in Stochastic Diffusion Search provided in Section 7.2 will make it possible to extend the model from Chapter 4 to cope with additional distractor in the search space. The notation will be extended in order to take into account the population of inactive agents locked to the best solution. Let the population of "free" agents will be denoted by N_f. Thus only N_f out of N agents will be able to locate the distractor and form a cluster supporting it. This population will be distributed according to the probability vector

$$\left(\overline{\pi}_1, \overline{\pi}_2, \overline{\pi}_3\right)$$

where

$$\begin{aligned} \overline{\pi}_1 &= \frac{2p^+(1-p_d)-1+\sqrt{\left[2p^+(1-p_d)-1\right]^2+4\left(p^+\right)^2(1-p_d)p_d}}{2p^+(1-p_d)}, \\ \overline{\pi}_2 &= -(1-p_d)(1-\overline{\pi}_1)^2+(1-\overline{\pi}_1), \\ \overline{\pi}_3 &= (1-p_d)(1-\overline{\pi}_1)^2 \end{aligned} \tag{7.8}$$

In equation (7.8) one used the first equation of (4.7) for the cluster supporting the best solution and the equation (7.6). This can be explained by the fact that once the supporting cluster has been formed for the best solution, the latter one, in a sense, becomes invisible for a number of agents equal exactly to the size of a population of "free" agents. It is now possible to derive the formulas reflecting explicitly the dependence of the cluster supporting the distractor on the total number of agents and parameters characterising the search space. Taking into account (7.6) one obtains the following expression for the steady state probability that an agent will be a part of a cluster supporting the distractor

$$(1-p_m)\left[\frac{1-\sqrt{\left[2(1-p^-)(1-p_m)-1\right]^2+4(1-p^-)^2(1-p_m)p_m}}{2(1-p^-)(1-p_m)}\right]^2\frac{2p^+(1-p_d)-1+\sqrt{\left[2p^+(1-p_d)-1\right]^2+4\left(p^+\right)^2(1-p_d)p_d}}{2p^+(1-p_d)} \tag{7.9}$$

In fact, the formula (7.9) has an interpretation limited to values of p^- and p^+ such that 1-$p^->p^+$. This is because otherwise the distractor would have became the best solution and due to the fluctuations agents would flow from the cluster corresponding to the weaker solution to the cluster supporting a new, stronger, solution. This property is the basis for Stochastic Diffusion Search's ability of escaping from local minima. During searches in big and noisy search spaces it may often occur that SDS would locate at first a relatively strong false positive solution.

Thus, a cluster supporting it will be formed as, temporarily, for SDS this is the best solution found so far. However, as the "free" agents still explore the search space, sooner or later they will locate the proper solution. This will quickly cause the minimum of the search landscape corresponding to the false positive disappear and the flow of agents will enable SDS to support the correct solution.

However, the extended Ehrenfest Urn model finds a distribution of agents for any pair of parameters p^-, p^+. From the above considerations it is clear than that a full approximate steady state probability distribution is given by the probability vector

$$\left(\pi_1^{'}, \pi_2^{'}, 1-\pi_1^{'}-\pi_2^{'}\right),$$

and the probabilities π_1‘, π_2‘ associated with the patterns characterised, respectively, by the pairs of parameters (p^-,p_m) and (p^+,p_d) are:

$$\pi_1^{'} = \begin{cases} \pi_1\left(p^-, p_m\right), & 1-p^- \geq p^+ \\ \pi_3\left(1-p^+, p_d\right)\overline{\pi}_1\left(p^-, p_m\right), & 1-p^- < p^+ \end{cases}$$
$$\pi_2^{'} = \begin{cases} \pi_3\left(p^-, p_m\right)\overline{\pi}_1\left(1-p^+, p_d\right), & 1-p^- \geq p^+ \\ \pi_1\left(1-p^+, p_d\right), & 1-p^- < p^+ \end{cases} \qquad (7.10)$$

where

$$\pi_3(x,y) = (1-y)\left[\frac{1-\sqrt{\left[2(1-x)(1-y)-1\right]^2 + 4(1-x)^2(1-y)y}}{2(1-x)(1-y)}\right]^2$$
$$\overline{\pi}_1(x,y) = \frac{2(1-x)(1-y)-1+\sqrt{\left[2(1-x)(1-y)-1\right]^2 + 4(1-x)^2(1-y)y}}{2(1-x)(1-y)}$$

and the used notation stresses the fact that the above formulas were drived from equations (7.6) and (7.8).

Figure 7.5 illustrates the dependence of π_1' in the equation (7.10) on the parameters p^-, p^+ characterising both patterns.

One can observe an overall similarity of graph in Figure 7.5 to that corresponding to it in Figure 5.8 of Chapter 5. The main difference between the amounts of agents allocated to the first pattern as predicted by extended Ehrenfest Urn model and the approximate solution seems to be confined mainly in the region of the parameter space, where both patterns are strong. Clearly, when the second pattern is the best fit solution, the approximate model overestimates, with respect to the Ehrenfest Urn solution, the amount of resources allocated to the first pattern in the region of a diagonal $1-p^-=p^+$.

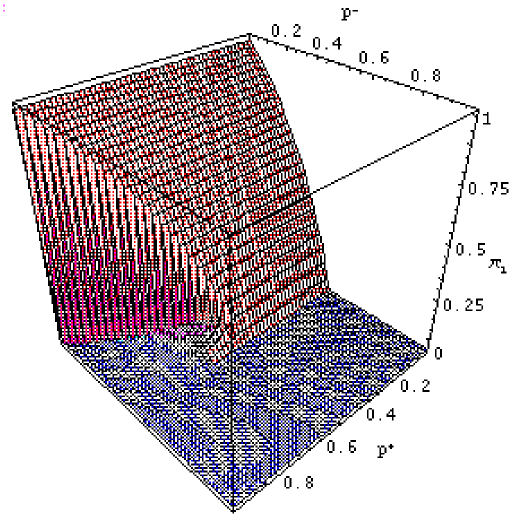


Figure 7.5 The normalised average number of active agents supporting the cluster concentrated on the first solution; $p_m=p_d=0.0001$.

The next section will present a more detailed comparison of proposed approximation and the extended Ehrenfest Urn model.

7.4. Qualitative and numerical comparison of extended Ehrenfest Urn model and approximate estimates of resource allocation in SDS.

In this section the approximate solution will be analysed in order to determine if it approximates well the exact model of Chapter 5.

In Figure 7.6 (a) one can see the graph of the logarithm of a difference of the orders of magnitude between average number of agents assigned to both patterns, as determined by the approximate probability distribution of finding an agent activated on either of the two.

The whole surface can be divided into four areas separated from each other by the zones of a very steep change of the difference of the orders of magnitude of allocated resources. Three of these are relatively flat with small to mild slopes. They correspond to both patterns bearing a weak resemblance to the template ($p^{-}>0.5$, $p^{+}<0.5$) and to patterns, of which one is strongly and the other - weakly similar to the template ($p^{-}<0.5$, $p^{+}<0.5$ or $p^{-}>0.5$, $p^{+}>0.5$).

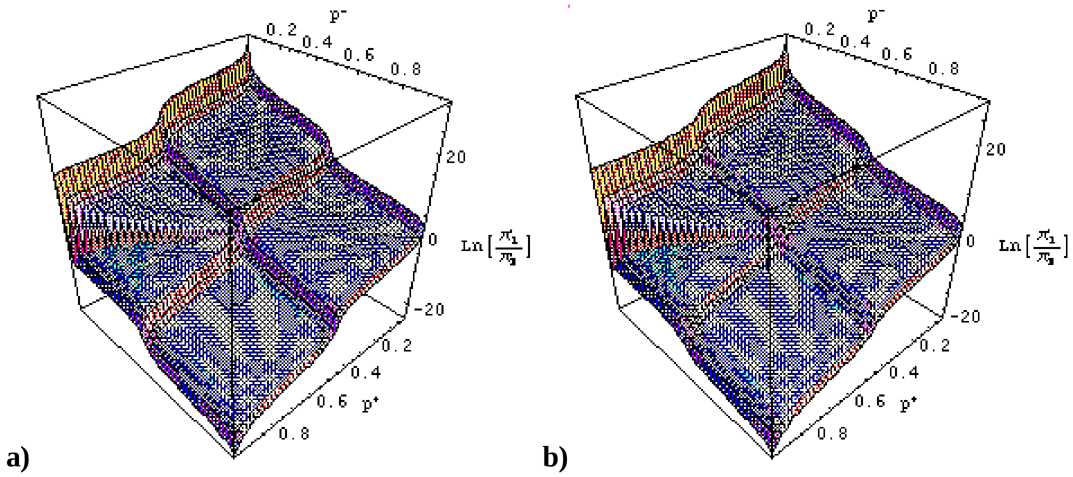


Figure 7.6 Each point of the surface describes the difference between equilibrium probabilities of finding an agent activated by one of the two patterns in the search space as predicted by the approximate model; (a) $p_m=p_d=0.0001$; (b) $p_m=p_d=0.001$.

In the first area, where both signals are weak, the distribution of agents between them is of similar order of magnitudes and changes smoothly when one changes the relative strength of the signals. The height of the areas of strong dominance of one of the signals over another reflects the highly biased allocation of resources towards the stronger signal. The fourth area, where both signals are strong ($p^{-}<0.5$, $p^{+}>0.5$), can be subdivided into two smooth subareas separated by a discontinuous transition along the diagonal. This transition corresponds to the abrupt reversal of the allocation of resources associated with a reversal of the relative strength of two signals. One can see, that away from the reversal region the allocation is similar to the one in areas with one strong and one weak pattern, but the difference in amounts of allocated resources is smaller.

Figure 7.6 corresponds to Figure 5.7 of Chapter 5. Overall, one can see that qualitatively both surfaces describe a similar distribution of agents. Both graphs are symmetric along the diagonal line corresponding to the equal amounts of agents allocated to both patterns, for which the logarithm of the ratio of both quantities vanishes. In fact, the allocation of resources seems to differ only in the region where both signals are strong. In Figure 5.7 this region is not separated from the region of dominance of one of the two signals, as is the case in Figure 7.6, and the difference of the heights of the surface on the both sides of the steep slope along the diagonal is much more pronounced than in Figure 7.6. It is worth noting that even in this region the surface of Figure 7.6 qualitatively describes well the character of a change of the surface from Figure 5.7.

In fact, the only regions where the qualitative predictions seem poor are the steep slopes separating this region from a region of a dominance of one signal over the other - they are present only in Figure 7.6, transition between corresponding regions in Figure 5.7 being smooth.

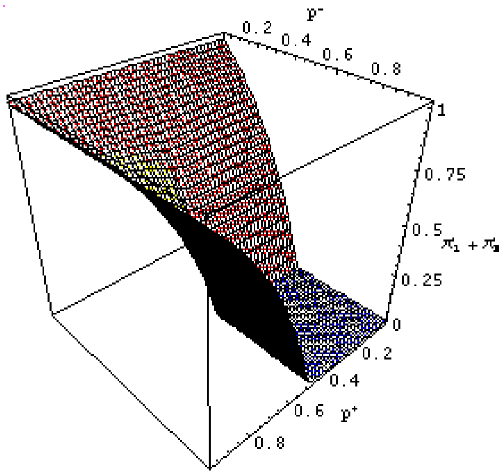


Figure 7.7 Total equilibrium probability of finding an agent activated by either of the patterns; p_m=p_d=0.0001.

Figure 7.7 shows the sum of probabilities of an agent being activated on either of the patterns. Again the similarity of this figure to the corresponding to it Figure 5.8 (b) from Chapter 5 is striking. The overall characteristics seems to be preserved very well with an exception of a slightly greater curvature of the surface in Figure 5.8.

From Figure 7.6 and Figure 7.7 it follows that the approximate solution describes quite well the resources distribution and their change with respect to changing the parameters, as predicted by the extended Ehrenfest urn model. However, these two graphs do not show where the quantitative prediction actually is not so good. In order to illustate this issue Figure 7.8 shows the norm of the difference between the equilibrium probability vectors predicted by the model and by the approximation.

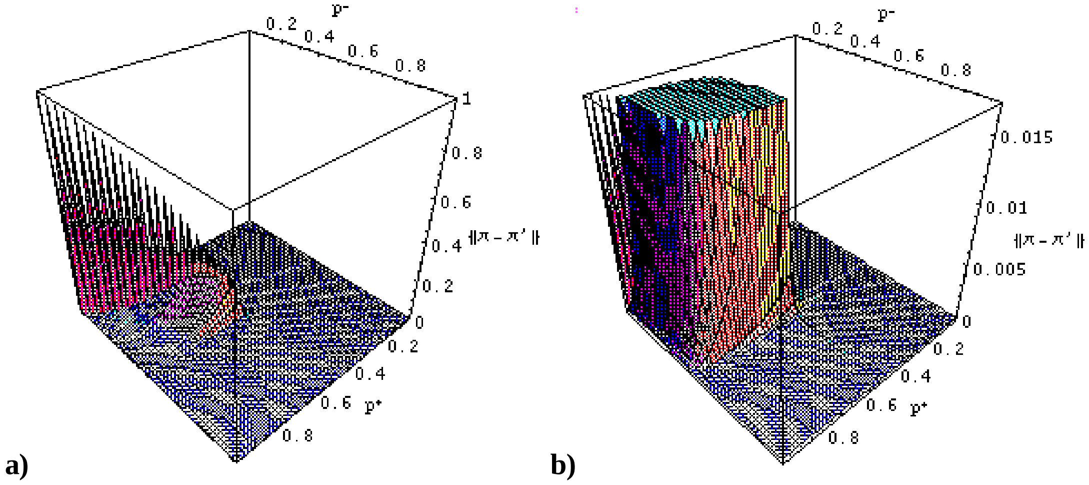


Figure 7.8 (a) The norm of a difference between the equilibrium probability distribution vectors predicted by the Ehrenfest urn model and by an approximation. (b) the same as in (a) but cut at the height 0.017; p_m=p_d= 0.0001.

From Figure 7.8 it follows that, as expected, the poor qualitative prediction is mainly confined to the values of parameters p^-, p^+ for which both signals are strong. In panel (a) one can see that the highest divergence between the two vectors is confined to a narrow area around a diagonal. The reason for this high discrepancy is that the construction of the approximate solution breaks down in that region.

Namely, it was assumed that SDS stabilises first on a stronger pattern and subsequently, with the remaining 'free' resources, it achieves an equilibrium on the second one. In other words, it was assumed that one can factorise the joint probability distribution into probability distribution associated with the stronger pattern alone and a conditional probability distribution given that the resources allocated to the first pattern correspond to the expected level of activity. This assumption implies that there is no interaction between the populations of agents concentrated on both patterns. This is a reasonable assumption when both patterns are weak, because they do not have a strong influence on each other, and also in the case when one of them is weak and one is strong, as in this case the interaction is limited, because the bigger cluster is of the size very close to the size it would have if the corresponding pattern was indeed the only pattern similar to the template in the search space (in a sense the cluster is nearly saturated). However, the stronger both patterns are, the stronger is the interaction between the corresponding clusters, resulting in an increased discrepancy between an approximate solution and the prediction of the Ehrenfest urn model. This is particularly true in the case when both patterns are equal, i.e. when the mutual interaction between the clusters is the strongest.

Panel (b) of Figure 7.8 shows the norm distance surface cut at the height 0.017. From the very steep increase of the surface on the border of the area where both patterns are strong one can infer that outside that area the approximation is predicting the resource allocation very well.

It is also interesting to note two small ridges outside of the area where each of the patterns is strong, both along the lines corresponding to the value of p_m=0.5 or p_d=0.5.

These are exactly the areas where the qualitative prediction of the change of the resource allocation, given by the approximate solution, shows too high sensitivity to the change between relative strength of the two patterns, as compared with the theoretical model.

In Chapter 5 one could observe that in some cases the theoretical model predicted very well the total allocation of resources to both patterns, while the distribution of agents between them was converging very slowly to its steady state limit.

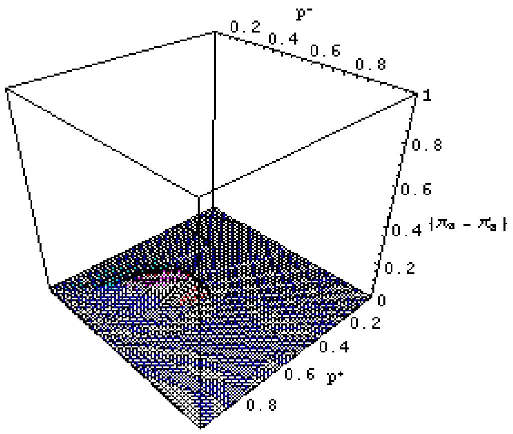


Figure 7.9 The difference between the total allocation of resources, as predicted by the Ehrenfest urn model and the approximation, as a function of p^- and p^+; p_m=p_d=0.0001.

p^+	$N\pi_1$	σ_1	$N\pi_2$	σ_2
0.32	197.728	14.0477	88.1574	9.38508
0.34	197.594	14.0429	105.269	10.2546
0.36	197.422	14.0368	127.207	11.2714
0.38	197.194	14.0287	156.341	12.4938

Table 7.1 Theoretical prediction of the average cluster sizes and their variation for p^-=0.6 and varying p^+. π_1 and π_2 are equilibrium probabilities of agent's activation on the best fit and the second pattern respectively and σ_1 and σ_2 are corresponding standard deviations; N=100000, p_m=p_d=0.001.

It is thus worth checking if the approximate method yields a reasonable approximation for the total allocation of resources. This will be illustrated in Figure 7.9 showing difference between third co-ordinates of the corresponding probability distribution vectors.

From Figure 7.9 one can see that, as expected for both patterns being weak or one weak and one strong pattern, both - the Ehrenfest urn model and the approximate solution, agree very well on the total number of agents allocated to both patterns. Also, according to one's expectations, the approximation yields worse quantitative prediction in the region of parameters where both patterns are strong. However, from comparison of Figure 7.9 and Figure 7.8 it follows that, in this, case the prediction of the total allocation of resources by approximate equilibrium probability distribution is much better than the detailed distribution of agents. Thus the approximation yields a reasonable estimate of the order of magnitude of the total allocation of resources by SDS also for values of parameters for which detailed information is lost.

p^+	$N\pi'_1$	σ'_1	$N\pi'_2$	σ'_2
0.32	198.42	14.0722	88.2217	9.38849
0.34	198.42	14.0722	105.381	10.2601
0.36	198.42	14.0722	127.397	11.2798
0.38	198.42	14.0722	156.662	12.5066

Table 7.2 Approximate prediction of the average cluster sizes and their variation for p^-=0.6 and varying p^+. π'_1 and π'_2 are equilibrium probabilities of agent's activation on the best fit and the second pattern respectively and σ'_1 and σ'_2 are corresponding standard deviations; N=100000, p_m=p_d=0.001.

Finally, the last part of this section will illustrate the approximate solution for the two patterns case by comparing its predictions concerning the equilibrium resource distribution with the predictions of the Ehrenfest urn model and with corresponding simulations of SDS. Table 7.1-Table 7.3 compare data corresponding to the case of weak patterns. Appendix 3 shows tables for two other settings of parameters.

As expected from previous analysis for this region of parameters, the agreement between predictions of the Ehrenfest urn model and of the approximate solution is very good. This is true for all the statistics presented in the tables. The estimates of these statistics from 1500 data points presented in Table 7.3 also agree with the theoretical and approximate results.

However, one can note that the sample means are closer to the theoretical predictions than the sample standard deviations. This is probably an effect of the finite sample size and of the fact that SDS converges to the theoretical equilibrium distribution in the limit.

p^+	n_1	σ_1	n_2	σ_2
0.32	196.3	21.3992	87.97	10.9659
0.34	197.511	22.4052	104.63	13.1627
0.36	200.639	20.6769	127.825	15.445
0.38	198.159	21.2858	155.021	17.4814

Table 7.3 Statistics from SDS simulation for p^-=0.6 and varying p^+. n_1 and n_2 are equilibrium probabilities of agent's activation on the best fit and the second pattern respectively and σ_1 and σ_2 are corresponding standard deviations; N=100000, p_m=p_d=0.001.

The tables presented in the Appendix 3 present data for values of parameters, where one would expect a poor prediction quality of the approximate solution. They show that although the approximate solution cannot be used for a reliable evaluation of the resource distribution, when both patterns are (almost) equally similar to the template, or when both are strong, however it can still give an impression of the qualitative behaviour of SDS under change of parameters.

7.5. Conclusions.

This chapter presented an approximate solution for the steady state distribution of SDS's resources in the case of two patterns in the search space similar to the template. The approximation relied on the factorisation of the joint probability distribution of the allocation of resources to both patterns into probability of the allocation of resources to the best fit pattern conditional on no influence from the second pattern and a probability distribution of the allocation of resources to the second pattern conditional on the stable cluster of agents allocated to the best fit pattern. These conditions hold approximately if the interaction between both populations of agents is weak, i.e. when both signals are weak, or when one of them is weak and the other is strong. In such cases the approximate solution allows one to obtain simple expressions for equilibrium probabilities and thus enables one to analyse the behaviour of SDS under changing of the parameters.

It is clear from the work presented so far in this thesis that the distribution of resources plays a very important role in SDS, allowing it to locate the best fit pattern and maintain the cluster supporting this solution in spite of presence of distractors. However, it is possible to construct algorithms similar in spirit to the Stochastic Diffusion Search which allocate resources according to different schemes. The next chapter will define two such algorithms and will describe how a change of the resource allocation influences their behaviour with respect to the corresponding behaviour of standard SDS.

CHAPTER 8
ALTERNATIVE RESOURCE ALLOCATION SCHEMES

8.1. *Introduction.*

The analysis of the behaviour of SDS, leading to the approximation scheme in Chapter 7, enabled one to understand better how the Stochastic Diffusion Search arrives at balancing of the exploitation and exploration of the search space. This balance is a crucial property as it affects the ability of SDS to separate signal from noise, to process other sub-optimal solutions or the time to find the best solution. One striking property of this balance, in the case of SDS, is a huge amplification of the small differences between the best fit solution and the distractors. This is definitely desired if one is interested in finding only the best solution, or alternatively if one has to process only a single task.

However, in many cases this type of resource allocation seems to form an unnecessarily big cluster of agents on the best fit pattern. In other words, SDS seems to put a higher weight to exploitation over an exploration of the search space.

This conclusion follows from the fact that, even if one is interested in a single solution, a smaller amount of resources might suffice to process it or distinguish it from the distractors. The above observation, together with the analysis of resource allocation discussed in Chapter 7, suggest that it is an interesting issue to derive other search procedures from the basic Stochastic Diffusion Search which strike a different balance between the exploitation and exploration - the two contradictory modes of processing the search space.

The observation made in remark 6 under the Proposition 4.3. of Chapter 4 will be used to propose a modified allocation of resources. Section 8.2 will present a discussion of how the new allocation scheme affects the number of agents forming the best-fit pattern's cluster. Subsequently two searches fitting in a new general scheme will be defined. It will be demonstrated that they retain some characteristic behaviour of SDS, however, in different search conditions, they will distribute the resources in a different manner. This will affect both the nonlinear discriminability of the best fit and convergence time of these methods. The selected simulations will make it possible to draw careful conclusions about the usefulness of these algorithms for performing tasks under different search objectives.

8.2. Modified resource allocation scheme.

The starting point towards a modified allocation scheme will be an observation that formation and size of the clusters supporting solutions is directly influenced by the “strength” of the corresponding solution and only indirectly by other clusters; interaction being the strongest in the case of patterns equally similar to the template. This is a consequence of the fact noted in the remark 6 under the Proposition 4.3 of Chapter 4, where it was mentioned that the evolution of an active agent is independent of the state of other agents in the population. The modification will in fact relax this requirement. This will allow one to introduce a greater dependence of the cluster of active agents on its environment, i.e. on the state of all the agents. As the Stochastic Diffusion Search places more emphasis on exploitation rather than on exploration of the search space, the objective will be to change this balance; more specifically, to maintain a balance more favorable for exploration of the search space without however compromising too much on the Stochastic Diffusion Search ability to maintain a stable population of agents on the best solution. To this end the evolution of an active agent will be made dependent on the state of other agents in the population. In order to shift the balance between exploitation and exploration in the desired direction it will be assumed that the activity of agents will have an inhibitory effect on the activity of each other. However, in order to spare the characteristics of signal from noise separability it will assume a form of a probabilistic rule analogous to the dependence of inactive agents on the state of the whole population.

Namely, in the new scheme active agents will be included into activity diffusion phase. The inactive agents will proceed as in the standard SDS. Each of active agents will perform a uniformly random draw from the whole population of agents. If an active agent chooses another active agent, it will become immediately inactive, otherwise it will retain its activity and will proceed as in SDS. All other phases will remain unchanged.

Thus, following the usual notation, the evolution of a single agent under the new resource allocation scheme in the Stochastic Diffusion Search is described by the following transition probability matrix:

$$P_n = \begin{array}{c|cc} & (a,m) & (n,*) \\ (a,m) & p_1^n & 1-p_1^n \\ (n,*) & p_2^n & 1-p_2^n \end{array}, \tag{8.1}$$

where

$$p_1{}^n = (1-\tfrac{m}{N})\left(1-p^-\right) + \tfrac{m}{N}\,p_m(1-p^-)\,,$$

$$p_2{}^n = \tfrac{m}{N}(1-p^-) + (1-\tfrac{m}{N})\,p_m(1-p^-)$$

and (a,m) denotes an active agent pointing to the correct solution and (n,*) denotes the inactive agent. It is necessary to note that, the notation for entries of the transition matrix used in this section is not consistent with the corresponding one from Chapter 4. The sub-indices of the entries of the stochastic matrix in (8.1) were reverted in order to agree with the natural numbering of rows. This should not, however cause any confusion.

Obviously, as in Chapter 4, the model corresponding to the above transition probability matrix is a non-homogenous Markov Chain. However, because the above stochastic matrix differs from the one describing original extended Ehrenfest urn model in first row only and the dependence of this row on parameters is analogous to the corresponding relationship for the second row, therefore the proof of the strong ergodicity of the model follows along the same lines as in Chapter 4. More specifically it is necessary to show first the weak ergodicity, and subsequently use the fixed point theorem in order to infer an asymptotic stationarity and so a strong ergodicity. The proofs of corresponding propositions are direct extensions of proofs from Chapter 4. and therefore are omitted.

The steady state probability distribution of the modified Stochastic Diffusion Search will be derived. In order to obtain equilibrium probabilities one has to solve a system of equations:

$$\left(\pi_1,\pi_2\right)\begin{bmatrix} p_1 & 1-p_1 \\ p_2 & 1-p_2^n \end{bmatrix} = \left(\pi_1,\pi_2\right), \tag{8.2}$$

where

$$\begin{aligned} p_1 &= (1-\pi_1)\left(1-p^-\right) + \pi_1 p_m(1-p^-), \\ p_2 &= \pi_1(1-p^-) + (1-\pi_1)p_m(1-p^-), \end{aligned}$$

and $\pi_1+\pi_2=1$.

After elimination of π_2 from the above constraint one obtains a quadratic equation in π_1, for which it has two solutions

$$\pi_1^i = \frac{1-2p_m\left(1-p^-\right)-2p^- \pm \sqrt{\left[1-2p_m\left(1-p^-\right)-2p^-\right]^2+8p_m\left(1-p^-\right)\left(1-p_m\right)\left(1-p^-\right)}}{4\left(1-p^-\right)\left(1-p_m\right)}, \qquad i=1,2$$

A straightforward analysis of the possible regions of the parameter space leads to the conclusion that the only possible solution of the steady state set of equations is given by the equation with a square root being added in a nominator. This equation can be rearranged into the form

$$\pi_1 = \frac{2\left(1-p^-\right)\left(1-p_m\right)-1+\sqrt{\left[2\left(1-p^-\right)\left(1-p_m\right)-1\right]^2+8\left(1-p^-\right)^2\left(1-p_m\right)p_m}}{4\left(1-p^-\right)\left(1-p_m\right)},$$

Thus the equilibrium probability vector is given by

$$\begin{gathered}\pi = \left(\pi_1, \pi_2\right) = \\ \left(\frac{2\left(1-p^-\right)\left(1-p_m\right)-1+\sqrt{\left[2\left(1-p^-\right)\left(1-p_m\right)-1\right]^2+8\left(1-p^-\right)^2\left(1-p_m\right)p_m}}{4\left(1-p^-\right)\left(1-p_m\right)},\right. \\ \left.\frac{2\left(1-p^-\right)\left(1-p_m\right)+1-\sqrt{\left[2\left(1-p^-\right)\left(1-p_m\right)-1\right]^2+8\left(1-p^-\right)^2\left(1-p_m\right)p_m}}{4\left(1-p^-\right)\left(1-p_m\right)}\right)\end{gathered} \tag{8.3}$$

It is interesting to compare the above equilibrium probability vector with the corresponding one from Chapter 4. One can see that these vectors have almost identical dependence on the parameters, the only differences are the constant in the second component of the sum under the square root sign and the multiplication constant in the denominator, both twice as big for the current probability vector than for the one from Chapter 4. This suggests that one should expect overall similar qualitative behaviour of the modified and standard SDS, with differences of quantitative nature. In fact, for values of the parameters for which the second term under the square root in (8.3) is negligibly small the steady state probability of activity of an agent is approximately half of the corresponding quantity for the standard Stochastic Diffusion Search.

As one could see the new resource allocation scheme reflects the change in the diffusion phase and therefore could be taken into account in the model by changing the transition probability matrix corresponding to the one step evolution of a single agent. However, the evolution of the whole population is analogous to that of SDS, so the discussion from Chapter 4, leading to the formulation of the probability distribution of resources in the Stochastic Diffusion Search, is valid also for the modified SDS presented here. It follows that the probability distribution of the number of agents clustered on the solution is binomial also in the present case. As before one expects the modified SDS to spend most of the iterations fluctuating around the quasi equilibrium. From standard formulas for statistics of the binomial distribution one can fully characterise the expected number of active agents and its variation around this state.

The next section will provide a thorough discussion of the distribution of agents in the modified SDS.

8.2.1. *Characterisation of the resource allocation of the modified Stochastic Diffusion Search.*

The section will start with a numerical example. Thus, one will consider the Stochastic Diffusion Search with N=1000 agents, a probability of a false negative $p^-=0.2$ and a probability of hitting the best fit instantiation $p_m=0.001$. From (8.3) it follows that $\pi_1=0.3757$, $\pi_2=0.6243$, $E(n)=375.7$, $\sigma=15.3$. One can see that indeed the expected value is close to a half of the expected value from the example in Chapter 4, although the standard deviation has not changed very much and still constitutes about 1.5% of the total number of agents. Of course, the variation relative to the average cluster size approximately doubled.

The mean return time to the 'all inactive' state is equal

$$m_0 = \frac{1}{\pi(0)} \approx \frac{1}{(0.6243)^{1000}} \propto 10^{204} \cdot$$

One can see then that, although it is much smaller than for the standard SDS, nevertheless it is still a very large value. Thus, in practice it indicates that in this case the modified SDS is very stable.

Consider a behaviour of the modified algorithm with N=10 agents. Here, $E(n)=3.8$ and $\sigma=1.5$, so the variability of the number of active agents raises to about 15% of the total (the relative variability raises to 39%). The mean return time to the 'no active' state decreased to

$$m_0 = \frac{1}{\pi(0)} \approx \frac{1}{(0.6243)^{10}} \approx 111.2 \propto 10^{2.04}$$

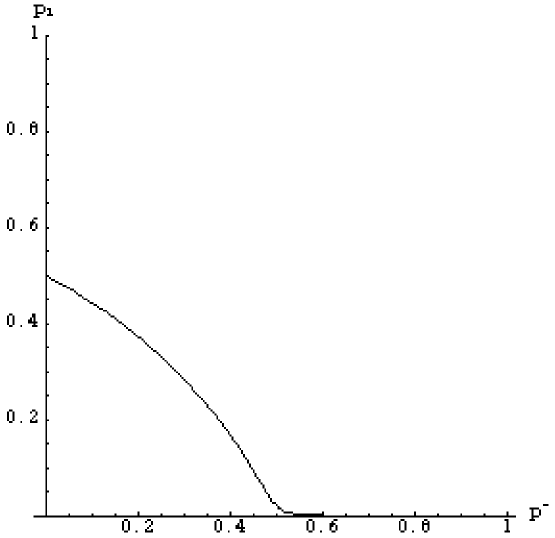


Figure 8.1 The normalised average number of active agents in the modified SDS as function of the parameter p^-; p_m=0.001.

so it became significantly low. One can thus expect that it indicates a poor stability of the quasi equilibrium state. In practice it means that a modified SDS would not, for this setting of parameters, support a stable cluster of active agents corresponding to the solution. This can be contrasted with the corresponding mean return time for the classical SDS, which was a four orders of magnitude greater than in the present case, and characterised still quite stable performance. One can see that the change of resource allocation resulted for a smaller number of agents in a qualitative change of the behaviour of the modified algorithm as compared with the original.

Obviously, also in the case of the modified SDS the ratio of the orders of magnitude of mean return times to ‘all inactive’ state is equal to the ratio of agents employed in the search.

Overall, one can see that the modification retained certain properties of the original algorithm. The qualitative changes are a result of the quantitative change in the amount of allocated resources, which is quite systematic. This means that the new algorithm does not introduce a new type of behaviour, but rather shows behaviour similar to that of SDS but in different areas of the parameter space.

In order to characterise further the equilibrium behaviour of the modified SDS Figure 8.1 will show the normalised average size of an active cluster as a function of the parameter p^- for p_m=0.001.

From Figure 8.1 one can see that the dependence of the resource allocation to the best fit pattern is qualitatively very similar to the one observed in the standard SDS. Namely the amount of resources decreases non-linearly and monotonically to very small values as p^- increases from 0 to about 0.5. On the interval from 0.5 to 1 one can observe again that the amounts of resources allocated to the best fit solution are orders of magnitude smaller than for stronger signal. However, as in Chapter 4, it follows from equations (8.3) that even for weak signals the average amount of active agents is strictly positive and vanishes only for the limiting value $p^-=1$. As was the case for the original algorithm, this has to be interpreted in the context of finite resources. Namely, it is always possible to find such a value of p^-, that above it the actual number of active agents is almost always null. However, increasing the total number of resources sufficiently high would stabilise a cluster of active agents supporting such a solution.

The main difference between Figure 4.1 and Figure 8.1 is that the latter decreases from around 0.5 for $p^-=0$, while the former decreases from 1. This means that, while in the case of the ideal instantiation of the template in the search space ($p^-=0$) the standard Stochastic Diffusion Search allocates to the solution all the resources which form a deterministic cluster, the modified SDS even in this case stabilises only in the statistical sense, with, on average, only about 50% of all agents allocated to the best fit pattern. This means that about 50% of

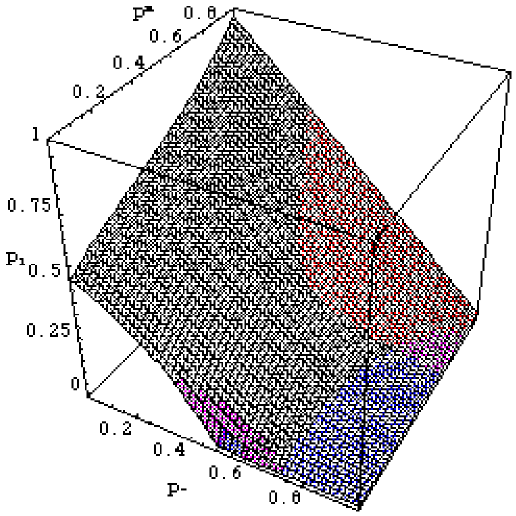


Figure 8.2 The normalised mean number of active agents in a modified resource allocation SDS as a function of both the false negative parameter p^- and the probability of a random hit on the best instantiation p_m.

agents remain inactive and search for alternative solutions.

Figure 8.2 shows the two dimensional plot of the normalised average number of agents allocated to the best fit solution as a function of p^- and p_m.

In Figure 8.2 one can see that the sharp transition between two distinct regions characterising different resource allocation by the modified SDS as a function of the parameter p^-, which are clearly distinguishable in Figure 8.1, gradually shifts towards greater values of p^- and disappears with increase of p_m. One can also observe a steady increase from around 0.5 to around 1 of a normalised ratio of resources allocated to the ideal instantiation of the template ($p^-=0$) with an increase of the probability of a random sampling of its position in the search space, p_m.

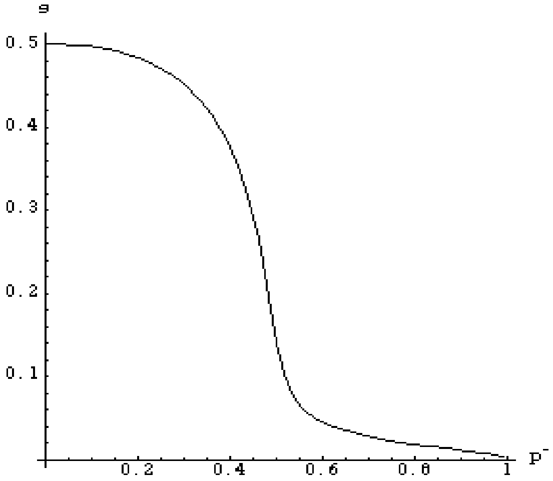


Figure 8.3 The rescaled standard deviation of the number of active agents calculated from the model for N=1000 agents and p_m=0.001; the scaling factor is $\alpha = N^{-0.5} \approx 31.62$.

Figure 8.3 and Figure 8.4 show the dependence of the standard deviation of the active cluster s a function of parameters characterising the best solution.

Figure 8.3 illustrates the behaviour of the standard deviation of the number of active agents concentrated on the solution as a function of p^- for p_m=0.001. One can observe that the fluctuation of the resource allocation in the modified SDS show slightly different dependence on the strength of the signal than the standard Stochastic Diffusion Search. The main difference is for the values of the parameter p^- corresponding to the strong signal ($0<p^-<0.5$),

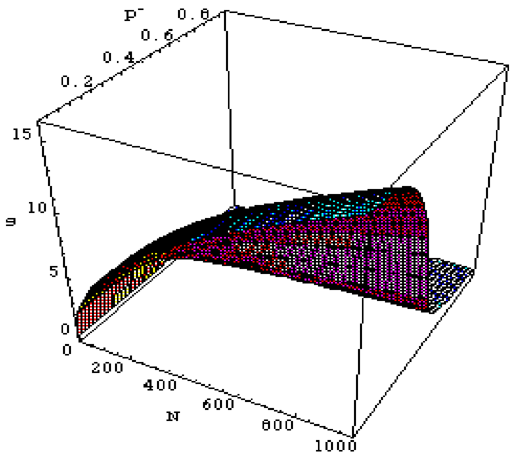


Figure 8.4 The standard deviation of the number of active agents in the modified SDS as a function of the total number of agents, N, and the probability of false negative, p^-; p_m=0.001.

where the maximum fluctuations of the active agents cluster size in the modified SDS are attained for the ideal instantiation of the pattern in the search space and the fluctuations decrease monotonously and non-linearly with an increase of p^-. This can be contrasted with the corresponding part of Figure 4.3 in Chapter 4 where fluctuations are null for p^-=0 and are maximal for the value of p^- about 0.4.

The amount of fluctuations in the remaining range of the parameter p^- changes analogously to that of Figure 4.3 - it steadily decreases to zero as p^- progresses from 0.5 to 1.

Figure 8.4 shows the 3D plot of the standard deviation of the amount of active agents as a function of p^- and N. One can see that Figure 8.3 corresponds to a cross-section of Figure 8.4 for a fixed number of agents used in the search.

8.2.2. *Simulations.*

This sub-section will present results of simulations of the modified Stochastic Diffusion Search, which are analogous to the simulations form Chapter 4. It will be interesting to compare between theoretical predictions concerning statistics of the steady state active cluster and their estimates from a finite sample. One will also be able to contrast these results with the data describing the standard SDS. To make the comparison possible the simulations will be run with N=1000 agents, p_m=0.001 and p^- assuming values 0.1, 0.2, 0.5 and 0.7 respectively. The simulations will run for 2000 iterations and first 500 data will be discarded as a burn-in period.

The results are summarised in Table 8.1 and illustrated in Figure 8.5 and Figure 8.6.

p^-	E[n]	σ	$\overline{n}$	$\overline{\sigma}$
0.1	445	15.7	431.66	16.7
0.2	375.7	15.32	350.35	16.93
0.5	21.9	4.63	17.22	9.92
0.7	0.75	0.86	0.73	1.1

Table 8.1 Statistics from the simulation of the modified SDS for p_m=0.001 and varying p^-. E[n] and $\overline{n}$ are the expected value and the sample mean estimate, respectively, of the best fit cluster size and σ and $\overline{\sigma}$ are the corresponding standard deviations; N=1000.

From Table 8.1 one can see that the finite sample estimates are, in general, close to their theoretical predictions from the model. The relatively higher discrepancies for the mean cluster size for the values of p^- equal to 0.2 and 0.5 may follow from the slower convergence towards the equilibrium. This conclusion seems to be supported by accompanying higher than expected

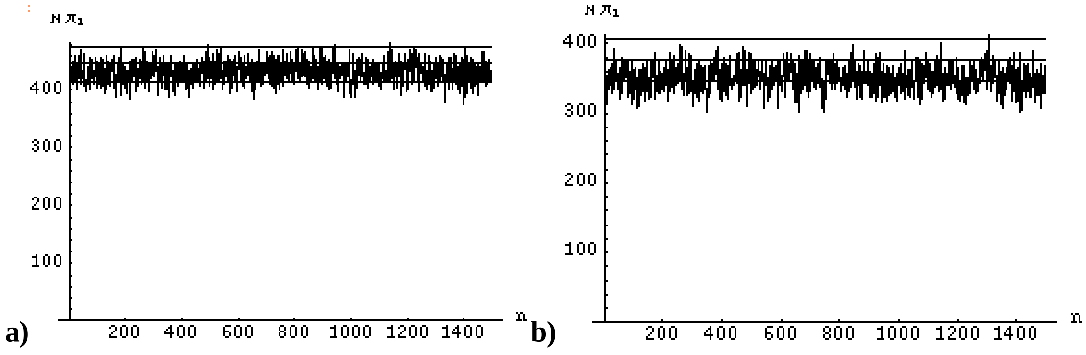


Figure 8.5 Evolution of the number of active agents in the modified SDS with N=1000 agents and p_m=0.001. The false negative parameter, p^-, is 0.1 (a), and 0.2 (b). The straight lines correspond to the expected average activity predicted by the model plus/minus two standard deviations.

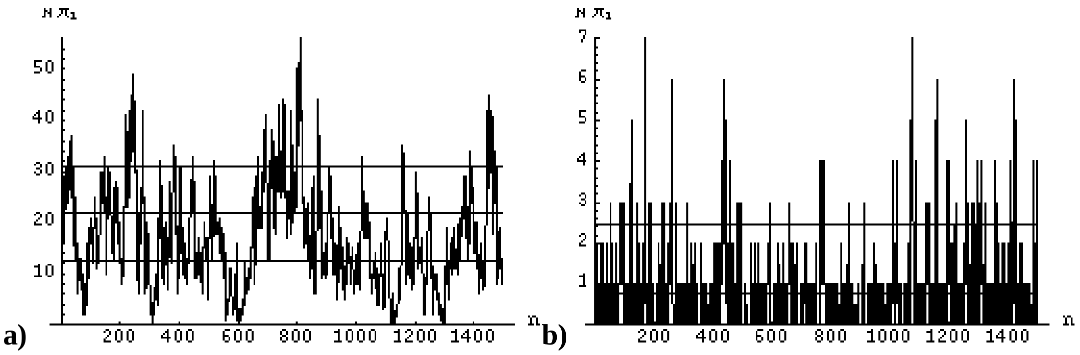


Figure 8.6 Evolution of the number of active agents in the modified SDS with N=1000 agents and p_m=0.001. The false negative parameter, p^-, is 0.5 (a), and 0.7 (b). The straight lines correspond to the expected average activity predicted by the model plus-minus two standard deviations.

values of the standard deviations.

In Figure 8.6 (a) one can see that for p^-=0.5, similarly to the corresponding case from Figure 4.6, in addition to the higher discrepancy between the expected value and the sample mean the cluster size varies considerably. The visible serial correlations are a clear sign of the non-stationarity of the system. This behaviour may be related to the fact that 0.5 is a value separating two regions in which resource allocation and active cluster fluctuation differ considerably as depicted in Figure 8.1 and Figure 8.3.

In Figure 8.6 (b) one can again see that, for the chosen value of N, the pattern characterised by the parameter p^-=0.7 cannot form a stable quasi-equilibrium state in the modified SDS.

Above, one looked at the amount of agents allocated to the solution by the modified SDS. It will be interesting to analyse next how the modification influenced the management of resources. One can observe that in order to compare the modified resource allocation with the one used in the standard SDS the actual amount of resources allocated to the solution should be rescaled by the inverse of the maximal amount of resources that the algorithm may concentrate on a pattern, for given value of p_m. This is because then the amounts of resources will be normalised to a unit interval, with 1 corresponding to the maximal possible cluster. This was not necessary in the case of the standard SDS because, in that case, the normalised maximal cluster size is 1 so rescaling does not change the range. However, in the case of the modified SDS this is no longer the case and one has to take rescaling into account before comparing the two algorithms with respect to the resource management.

Thus, Table 8.2 compares the rescaled amounts of resources allocated to the best fit pattern, for the same fixed values of p_m and N.

p^-	$E_{modified}[n]$	$E_{standard}[n]$
0.1	0.889	0.888
0.2	0.7506	0.75
0.5	0.0437	0.0307
0.7	0.0015	0.00075

Table 8.2 Relative amounts of the resource allocation to the best fit pattern as a function of the parameter p^- (the first column), for p_m=0.001. The second column contains the normalised best fit cluster size for the modified SDS, rescaled by $\alpha = (0.5005)^{-1}$ - the inverse of the normalised maximal cluster size for a given p_m. The third column contains corresponding data for the standard SDS (the scaling factor is 1 in this case).

It follows that the relative amounts of resources allocated to the best fit pattern are very close for both algorithms. Thus, the conclusion follows that the discrimination of the best fit pattern from the background is not affected by the modification of the resource allocation.

8.2.3. Contrastive discussion of the operation mode of standard and modified SDS.

One can also consider the size of the population of inactive agents locked to the best fit solution and the 'free' agents population, as defined in Chapter 7. The extension of the model necessary to quantify them parallels the discussion of the corresponding quantities in that chapter. However, although it leads to the analytic solution for the equilibrium distribution of resources between the three populations, the form of this solution is very complicated and thus not very informative. The symbolic package Mathematica was used in order to derive the closed form solutions. For the reasons mentioned above derivation of the model and solutions were omitted here. Instead, graphs obtained in Mathematica, of 'inactive locked to the best fit solution' and 'free' population sizes as functions of the false negative parameter p^- will be presented.

Figure 8.7 presents the relevant graphs. Again one can note similarities with the corresponding graphs of Figure 7.1. The graph of the 'inactive' population in panel (a) of Figure 8.7 shows similar dependence on the parameter p^- as the corresponding graph from Figure 7.1, the main difference being the (approximately twice) lower maximum in the former graph. Also panels (b) of these Figures show a very big similarity, differing mainly in the value of the minimal size of the ‘free’ population. From the comparison of both figures it follows that the modified resource allocation scheme reduces the size of the population of inactive agents which are not directly used neither for the support of the best fit solution nor for the exploration of the search space. Moreover, the minimal amount of agents exploring the search space equals, on average, half of the whole population, and increases quickly with increase of p$^-$. This means that the modified SDS will very actively explore the search space with at least half of its resources, while still retaining the capability of the standard SDS of creating the stable support of the best fit solution in the quasi equilibrium state.

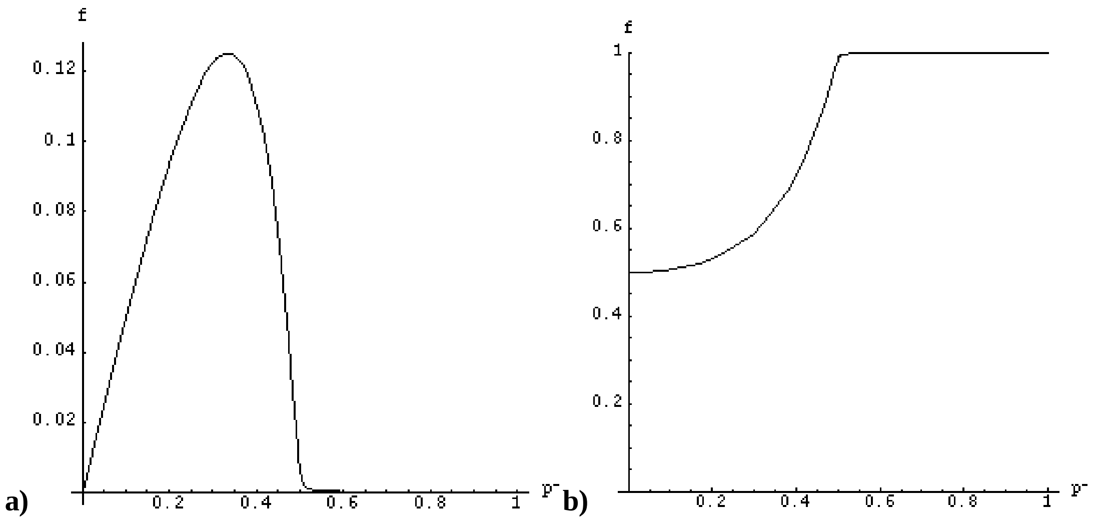


Figure 8.7 (a) The normalised population of the inactive agents ‘locked’ to the best solution, π_2; (b) and the population of agents free to explore the search space, π_3, as functions of p$^-$ in the modified SDS, p_m=0.0001.

One has to note here, that it was not necessary to rescale the graphs of Figure 8.7 in order to perform comparisons with Figure 7.1 or draw the conclusions. This is because rescaling was important only for the discrimination of the solution from the background. In other words, in this case the important factor for pattern discrimination was not the absolute amount of allocated resources, but their size relative to the maximal allowable cluster size (only in the particular case when the maximal allowable cluster size equals the whole population both quantities are equivalent). This is not so for the resource management concerning the exploration of the search space, because it is the absolute amount of resources deployed that matters for the efficient exploration.

Although one could observe that the modified allocation scheme does not change very much the form of dependence of the resource allocation on the parameters, nevertheless it is interesting to check if it influences the time complexity of the system. In order to investigate this question the convergence time to the ideal instantiation of the template while varying the search space size was analysed for the modified SDS. Figure 8.8 shows the mean convergence time and the standard deviation, averaged over 1000 runs, as a function of the search space size.

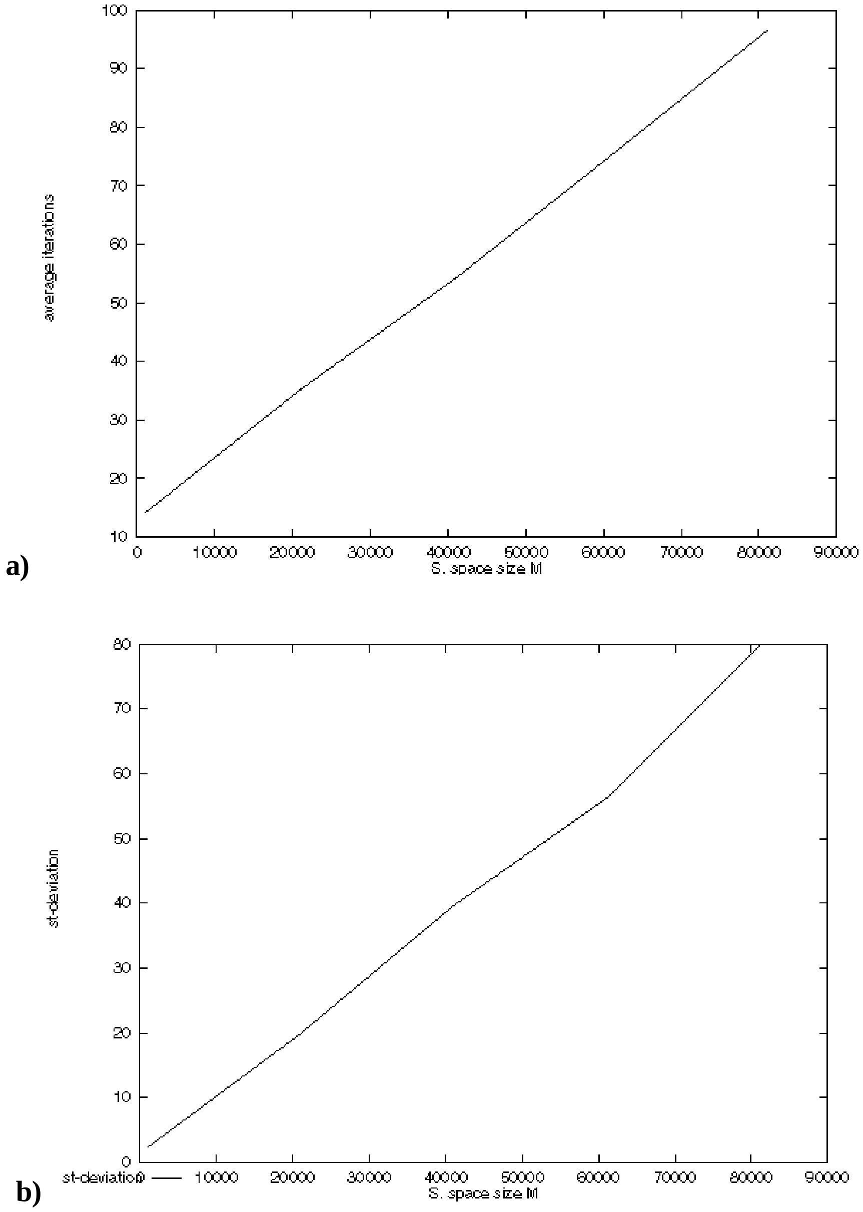


Figure 8.8 The time complexity of the modified SDS, for N=1000 agents and a noisless search space. Each point corresponds to the average of 1000 runs for a particular search space size, which assumes the values 1000, 21000, 41000, 61000 and 81000. (a) mean convergence time, (b) standard deviation.

The p_m - the probability of a random hit on the best fit pattern varies accordingly to the search space size. One can see that the modified resource allocation scheme did not change the linear time complexity of the algorithm. In particular, both the mean convergence time, panel (a), and the standard deviation increase linearly with the search space size. Moreover, comparing Figure 8.8 and Figure 3.4 from Chapter 3 it can be concluded that both quantities remained practically unaffected. Thus the change of the resource allocation strategy did not alter the convergence time characteristics of SDS. This is not surprising, if one considers that the modified SDS retained the pattern discrimination capabilities of the original SDS but moved the balance between the exploitation and exploration of the search space towards the latter. However, in the noise free search space only the discrimination capability is relevant for the convergence time since, until the pattern is hit, both algorithms allocate 100% of resources to the exploration.

The above observations together with the ones made in relation to Figure 8.1 and Figure 8.2 and concerning the resource allocation to the ideal instantiation of the template suggest the following problem. Why should the modified SDS still search for alternatives if it has already found the ideal instantiation of the template ? However, the same question can be reformulated and asked in the case of the standard SDS - why concentrate all the resources on the ideal solution when it would be sufficient to allocate only as much as is necessary to differentiate it from anything else ?

The former question implicitly implies that any algorithm, so the modified SDS in particular, continuing search after finding the best fit solution is less optimal than the standard SDS in managing the computational resources. The latter, suggests, in a sense, the least action principle - “allocate only as much resources as is necessary to discriminate the best solution from the rest of the search space”. This renders any algorithm, allocating to the ideal template instantiation more resources than is necessary for a succesful discrimination (in particular the standard SDS), less optimal with respect to the computational resources management than the modified SDS. This claim is based on the fact that the cluster size equal to the half of the resources, corresponding to the ideal instantiation of the template, constitutes the lower bound of the cluster sizes allowing for the unambiguous discrimination of the ideal solution.

It needs to be stressed that the claims made here have to be understood in a weak sense, i.e. it is suggested only that both scenarios implied by the above questions correspond to *particular* orderings of search algorithms with respect to the *particular* interpretation of the optimality of resource management. Thus these statements do not form strong claims pertaining to the global optimality of any of the two algorithms with respect to the above optimality criteria.

It thus seems natural that any interpretation of the search performance depends on the objectives of the search, or in other words on the particular context of the search.

Thus, if the objective of the search is to find a best solution (i.e. any of the patterns that are equally and maximally similar to the template), then the standard SDS is going to provide one with the satisfactory solution for a broader range of the parameter p^- than the modified algorithm.

Below three scenarios are proposed, in which SDS with the modified resource allocation scheme might be the more appropriate choice:

- the assumption of the uniqueness of the best instantiation of the template in the search space is relaxed and the search algorithm is to find all best fit solutions,
- the search algorithm is to find not only the best fit pattern(s) but also the most promising sub-optimal solutions,
- the search space or/and the template are changing dynamically so the best solution at one time instant may not be so some time later.

From the analysis performed in Chapter 4 and Chapter 5 one can infer the behaviour of the standard SDS in these cases. In the first scenario the classical Stochastic Diffusion Search will converge to a deterministic steady state which will be characterised by all agents being active and a distribution of resources dependent on the initial conditions (however a typical situation is when SDS allocates all resources to one of the best solutions only, or fixed amounts of agents to few of them). In the second case, SDS would in principle distribute resources among all potentially promising solutions but as one could observe in Chapter 5 the distribution would be very strongly biased in favour of the best fit pattern(s), rendering clusters of active agents corresponding to all the other potential solutions practically unstable. Finally, in the dynamic scenario case, it is intuitively necessary to employ more resources for the exploration of the search space, in order to react immediately to changes in the search space or to updates of the template. In this case, one could expect SDS reacting slower to the changes than an algorithm with a more even balance between exploitation and exploration.

In fact, the last two cases are closely related, because they can be reduced to one another. Namely, the search in the dynamic environment can be treated as a search in a large noisy search space in which the probability of the first found solution being a false positive is high. This is clear if one assumes the “third person” perspective and observes that all the patterns have to be automatically reassessed once the environment or the template change. The “third person perspective has to be understood as opposed to that of the modified SDS or the “first person” perspective, in which the system at any time has a limited access to the search space and the only sign of the changes in the environment and/or template is a change of the relative strength of the best fit pattern found so far. Similarly the search in the dynamic environment can be treated as performed in a large, noisy search space and any changes of the best fit solution can be interpreted as the case of convergence on the false positive. Thus, one can discuss both these cases together without making a reference to their distinctive features.

In the remainder of this chapter some of the features of the modified resource allocation scheme will be analysed, which bear direct implication for the search characteristics relevant to the above mentioned cases.

8.3. *Context-free extension of the modified resource allocation.*

In order to proceed one has to define first the modified resource allocation scheme in the case of two patterns similar to the template. The first definition of the modified resource allocation included the possibility of a change of the status of active agents in the diffusion phase. The total amount of active agents was self regulating because the decision about the status of an active agent was dependent on the other active agents - the more numerous the active agents were, the more any of them was likely to chose another one during a diffusion phase and hence to become inactive.

In the following extension of the modified resource allocation scheme in the two patterns case this self regulation property of the total number of active agents will be preserved. Thus, it will be ascertained that if an active agent chooses for communication another active agent, it becomes inactive. The Extended Ehrenfest urn model of the standard SDS defined in Chapter 5 for the two patterns case will be used. The modification of the resource allocation will be reflected in the appropriate change of the entries of the stochastic matrix defining the one step evolution of the standard SDS.

The model of the extension of the modified SDS assumes the following form

$$P = \begin{array}{c} \\ (a_m, m) \\ (a_d, m) \\ (n, *) \end{array} \begin{array}{c} \begin{array}{ccc} (a_m, m) & (a_d, m) & (n, *) \end{array} \\ \begin{bmatrix} p_{11} & p_{12} & 1 - p_{11} - p_{12} \\ p_{21} & p_{22} & 1 - p_{21} - p_{22} \\ p_{31} & p_{32} & 1 - p_{31} - p_{32} \end{bmatrix} \end{array} \qquad (8.1)$$

where

$$
\begin{aligned}
p_{11} &= \left(1-\tfrac{m_1}{N}-\tfrac{m_2}{N}\right)\left(1-p^-\right)+\left(\tfrac{m_1}{N}+\tfrac{m_2}{N}\right)p_m\left(1-p^-\right),\\
p_{12} &= \left(\tfrac{m_1}{N}+\tfrac{m_2}{N}\right)p_d\,p^+,\\
p_{21} &= \left(\tfrac{m_1}{N}+\tfrac{m_2}{N}\right)p_m\left(1-p^-\right),\\
p_{22} &= \left(1-\tfrac{m_1}{N}-\tfrac{m_2}{N}\right)p^+ +\left(\tfrac{m_1}{N}+\tfrac{m_2}{N}\right)p_d\,p^+,\\
p_{31} &= \tfrac{m_1}{N}\left(1-p^-\right)+\left(1-\left(\tfrac{m_1}{N}+\tfrac{m_2}{N}\right)\right)p_m\left(1-p^-\right),\\
p_{32} &= \tfrac{m_2}{N}p^+ +\left(1-\left(\tfrac{m_1}{N}+\tfrac{m_2}{N}\right)\right)p_d\,p^+,
\end{aligned}
\tag{8.2}
$$

and m_1, m_2 denote, respectively, the size of populations of active agents concentrated on corresponding patterns similar to the template.

Notes.

1. One can see in the equations (8.1)-(8.2) that all the entries of the one step probability transition matrix are non-zero in general. Also, in contrast to the standard SDS, all of them are now dependent on the state of the search and therefore change in time.

2. It is interesting to observe that the extension of the modified resource allocation scheme defined in such a way effectively uses only general information about the activity of the system, in comparison with the standard SDS, in order to modify the active cluster sizes. In other words, the self maintenance of the overall activity depends on the activity level and not on the information that caused this activity. Thus the modification of activity of agents is context independent. This is because the activity of an agent is affected by the overall number of active agents regardless of causes of their activity. One can compare this scenario to the more traditional connectionist models in which the activity of a single model neuron is dependent on the gross incoming activity from other neurons to which it is connected and not on the information content of this activity. These modes of operation are reminiscent of current computational models of the information processing in the brain, in which it is (often implicitly) assumed that the information is carried in the activity level (mean firing rate) of neurons and that the particular type of the encoded information does not bear any influence on the processing. In other words, the same activity levels corresponding to encoding of different features would be processed in the same way regardless of the fact that each time they may correspond to a qualitatively completely different type of information.

3. For the reasons discussed in the note 2 the above algorithm will be referred to as a 'context-free' extension of SDS.

4. As it will become clear from the second part of this chapter it is possible to define the extension of the modified resource allocation scheme, which will take into account the context dependence.

In order to discuss the above extension of the modified resource allocation scheme one will have to find the equilibrium probability distribution vector from the usual fixed point equations

$$\begin{aligned} \pi P &= \pi, \\ \pi &= (\pi_1, \pi_2, \pi_3) \end{aligned} \tag{8.3}$$

where

$$\sum \pi_i = 1, \pi_i \geq 0, \quad i = 1, 2, 3 \,.$$

This equation (8.3) can be transformed into a system of two nonlinear equations with two unknowns

$$\begin{aligned} \pi_1 &= (1 - p^-)\left[-2(1 - p_m)\pi_1^2 + 2p_m\pi_2^2 + 2(2p_m - 1)\pi_1\pi_2 - 2p_m\pi_2 + 2(1 - p_m)\pi_1 + p_m\right], \\ \pi_2 &= p^+\left[2p_d\pi_1^2 - 2(1 - p_d)\pi_2^2 + 2(2p_d - 1)\pi_1\pi_2 + 2(1 - p_d)\pi_2 - 2p_d\pi_1 + p_d\right] \end{aligned} \tag{8.4}$$

where π_1 and π_2 are the appropriate equilibrium probabilities of finding an agent activated on one of the two patterns.

Notes.

1. It is possible to define the evolution of the probability distribution π in terms of an orbit of a non-linear mapping S, analogously to the mapping introduced in Chapter 5. Thus, one can write

$$\begin{aligned} \pi_1^{n+1} &= f_1(\pi_1, \pi_2), \\ \pi_2^{n+1} &= f_2(\pi_1, \pi_2), \end{aligned}$$

where f_1 and f_2 are defined by the right hand sides of (8.4).

One can now observe that the modified SDS is related to the standard Stochastic Diffusion Search in the following way

$$f_1(\pi_1,\pi_2) = f_1^{SDS}(\pi_1,\pi_2) + (1-p^-)\left[-(1-p_m)\pi_1^2 + p_m\pi_2^2 + (2p_m-1)\pi_1\pi_2\right],$$
$$f_2(\pi_1,\pi_2) = f_2^{SDS}(\pi_1,\pi_2) + p^+\left[p_d\pi_1^2 - (1-p_d)\pi_2^2 + (2p_d-1)\pi_1\pi_2\right]$$

where f_1^{SDS} and f_2^{SDS} denote the corresponding mappings of the standard SDS.

2. From the above it immediately follows that, for $p_m=p_d$, the extended algorithm enjoys symmetry analogous to that of the standard SDS. Namely, in that case

$$f_1(\pi_1,\pi_2)\frac{1}{1-p^-} = f_2(\pi_2,\pi_1)\frac{1}{p^+},$$
$$f_2(\pi_1,\pi_2)\frac{1}{p^+} = f_1(\pi_2,\pi_1)\frac{1}{1-p^-}$$

8.3.1. *Characterisation of the properties of the context-free SDS.*

The symbolic package Mathematica was used in order to obtain solutions of the system (8.4). From the solution of this system one can plot the surface describing the dependence of the allocation of agents to the two patterns. This graph is shown in Figure 8.9.

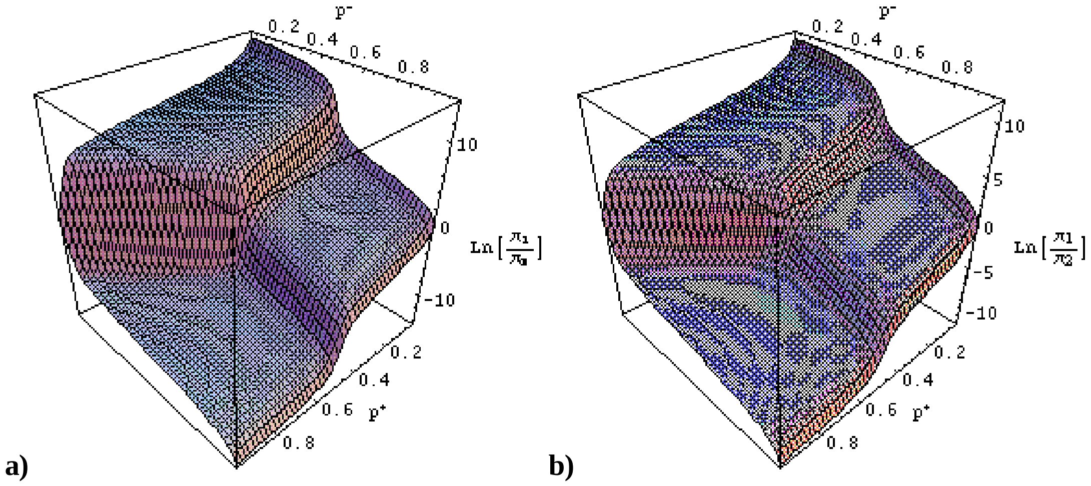


Figure 8.9 The surface describing an average difference of orders of magnitude of resources allocated to both patterns by the modified SDS, as a function of parameters p^- and p^+, (a) $p_m=p_d=0.0001$, (b) $p_m=p_d=0.001$.

One can notice immediately that the graphs in Figure 8.9 are very similar to the corresponding graph from Figure 5.7 of Chapter 5. There are analogous three flat areas corresponding to similar regions of the parameter space. The transitions between them are abrupt especially in the region of the parameters space corresponding to both patterns being strong along the diagonal $p^+=1-p^-$. This means that the resource allocation of the modified Stochastic Diffusion Search follows a similar overall dependence on the parameters as the standard SDS. In particular, for two patterns of equal strength it implies that the modified SDS will allocate the same order of magnitude of resources. In fact it follows, from the solution of equations (8.4), that average amounts of agents allocated to the best fit patterns of equal strength should be the same. It will be also the case for two ideal instantiations of the template in the search space. It is worth stressing that the modified SDS will maintain the equal distribution of resources between two ideal instantiations independently of initial conditions. The amounts of agents assigned to particular solution will fluctuate around its quasi equilibrium value.

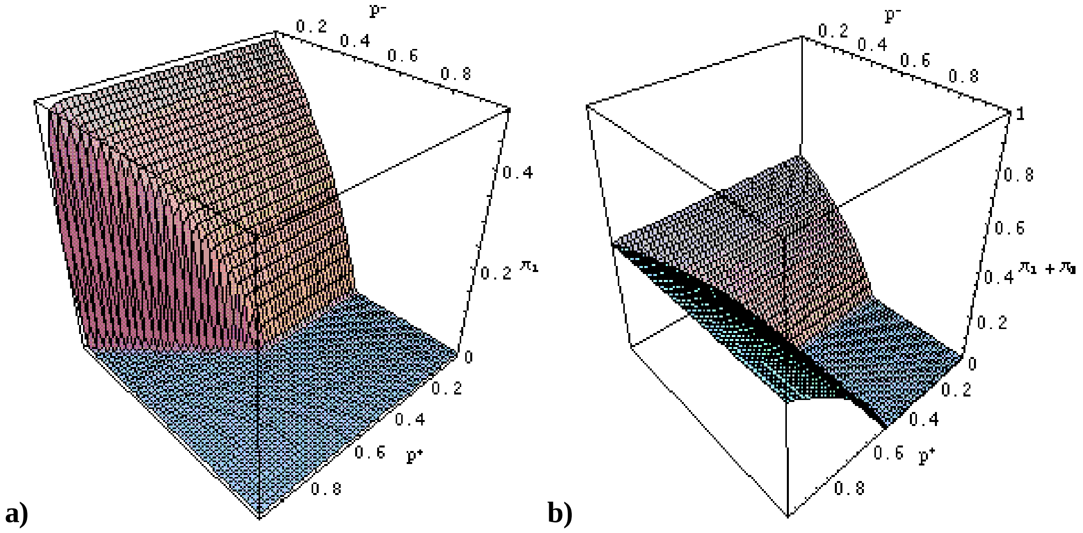


Figure 8.10 The equilibrium probability of the modified SDS of finding an agent activated by the first pattern, the panel (a), and the total probability of finding an agent activated, i.e. a sum of π_1 and π_2, the panel (b); $p_m=p_d=0.0001$.

This is different from the corresponding behaviour of the standard SDS. In the latter case one could expect an equal amount of resources distributed among two ideal instantiations of the template only if one averaged them across many runs. In any single run the distribution of resources would be fixed, uneven and, to a high extent, dependent on the initial conditions.

If, however, the two patterns are not equally similar to the template, the modified SDS allocates the majority of resources to the stronger one and the dependence of the allocation on the parameters p^+ and p^- is analogous to that of the standard SDS. Comparing the graphs in panels (a) and (b) shows that the change in the allocation of resources associated with the change of the parameter $p_m=p_d$ also shows similar pattern to the corresponding change in the case of the standard SDS. Graphs in Figure 8.10, showing the average normalised amount of resources allocated to the best fit pattern - the panel (a) - and the total amount of resources allocated to both patterns - the panel (b) - confirm the qualitative similarity of the dependence of these quantities on the parameters p^- and p^+ in both algorithms.

However, it is possible to note also some clear differences in the allocation of resources visible in Figure 8.9 and Figure 8.10. The surfaces of Figure 5.7 span approximately twice the range of values attained by the corresponding graphs of Figure 8.9. This means that, although the resource allocation changes analogously in both algorithms with the same change of parameters, however there is a systematic difference between amounts of allocated resources. The difference between amounts of agents allocated to both patterns for fixed values of the parameters p^-, p^+ is in general smaller for the modified than for the standard SDS. Also the transition between areas in the search space, in which the difference in the average orders of magnitude of allocated resources does not change significantly, is less abrupt than in the case of the graphs corresponding to the standard SDS. This means that, for example in the region of a diagonal $p^+=1-p^-$ for $p^+>0.5$ the modified algorithm is less sensitive to the difference between the best fit pattern and the second best solution. From the above observation it follows that, for very small differences between two patterns, the numbers of agents allocated to them will differ to a much lesser extent than would be the case for a standard SDS, but nevertheless will increase very rapidly with the increase of the difference between the two patterns. The same observation is valid for the transition areas around the values of parameters p_m and p_d being constant and equal to 0.5. From Figure 8.10 one can see that the average total amount of resources allocated to both ideal instantiations of the template in the search space is equal to around half of the number of all agents. Therefore the modified SDS will keep a considerable amount of the resources for exploration of the search space also in the case of many ideal instantiations of the template.

The above discussion utilised implicitly the fact that the equilibrium probabilities obtained from the solution of the system of equations (8.4) are used as parameters of the multinomial joint probability distribution of the allocation of resources analogous to the one derived for the standard SDS in Chapter 5.

Next, the properties of the modified resource allocation scheme will be illustrated by means of the joint probability distribution. Figure 8.11 shows the joint probability distribution of the allocation of resources to the two patterns of equal strength.

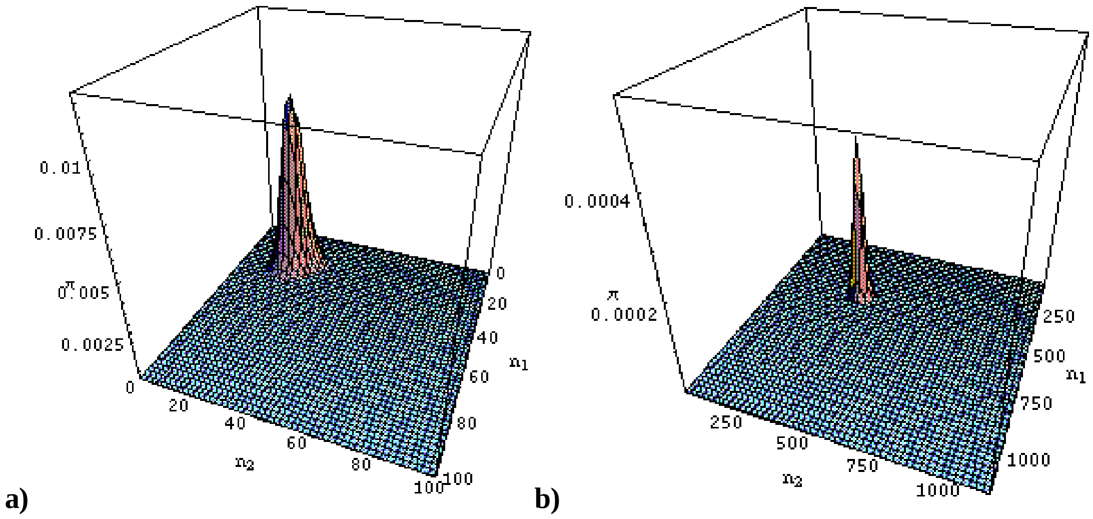


Figure 8.11 The joint probability distribution of the number of agents allocated by the modified SDS to two patterns of equal strength; (a) N=100, $p^{+}=1-p^{-}=0.9$; (b) N=100000, $p^{+}=1-p^{-}=0.5$; $p_m=p_d=0.0001$.

One can observe in Figure 8.11 (a) that, in the case of both patterns equally strong and very similar to the template, the modified SDS forms a joint probability distribution peaking very strongly around the most probable distribution of agents. The difference between this distribution and the distribution of the standard SDS for corresponding values of parameters is in the location of the peak; in the latter case the peak removed further away from the origin along the diagonal. For the graph in panel (b) in both cases one had to increase the total number of agents to N= 100000 in order to observe a clear peak of the probability distribution. These graphs illustrate the fact that the modified SDS allocates equal amounts of resources to the patterns equally similar to the template but less than it would be the case for the standard algorithm. Figure 8.12 and Figure 8.13 address the case of the patterns of unequal strength.

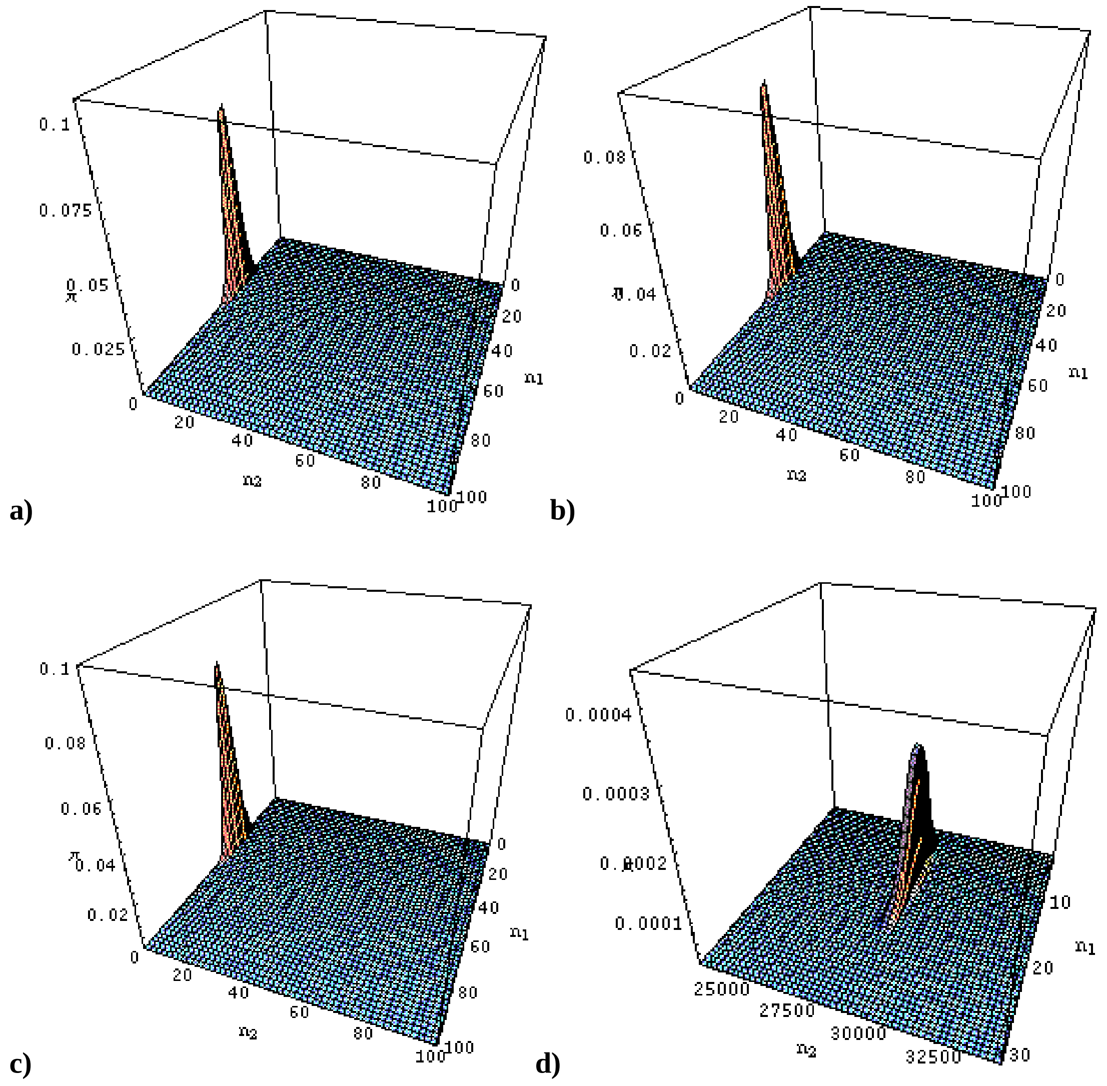


Figure 8.12 The joint probability distribution of the number of agents allocated by the modified SDS to two unequal patterns; (a) N=100, p^+=0.2, p^-=0.2; (b) N=100, p^+=0.6, p^-=0.2; (c) N=100, p^+=0.7, p^-=0.2; (d) N=100000, p^+=0.7, p^-=0.5; p_m=p_d=0.0001.

It is clear from Figure 8.12 that the modified SDS allocates most of the resources to the stronger pattern. In fact comparing Figure 8.12 (a), (d) with Figures 5.9 (a) and 5.11 (b) one can see that both modified and standard SDS treat similarly patterns differing significantly from each other. In particular it is worth noting in Figure 8.12 (a) - (c) that for a fixed p^- and a wide range of values of the parameter p^+ its increase does not shift significantly the peak of the joint probability distribution. This feature, common to both algorithms, is characteristic for the regions of the parameter space corresponding to relatively flat areas of the surfaces depicted in Figure 8.9 and 5.7.

In the light of this similarity it is interesting to compare Figure 8.11 (a), Figure 8.13 and Figure 5.10 (a), (c).

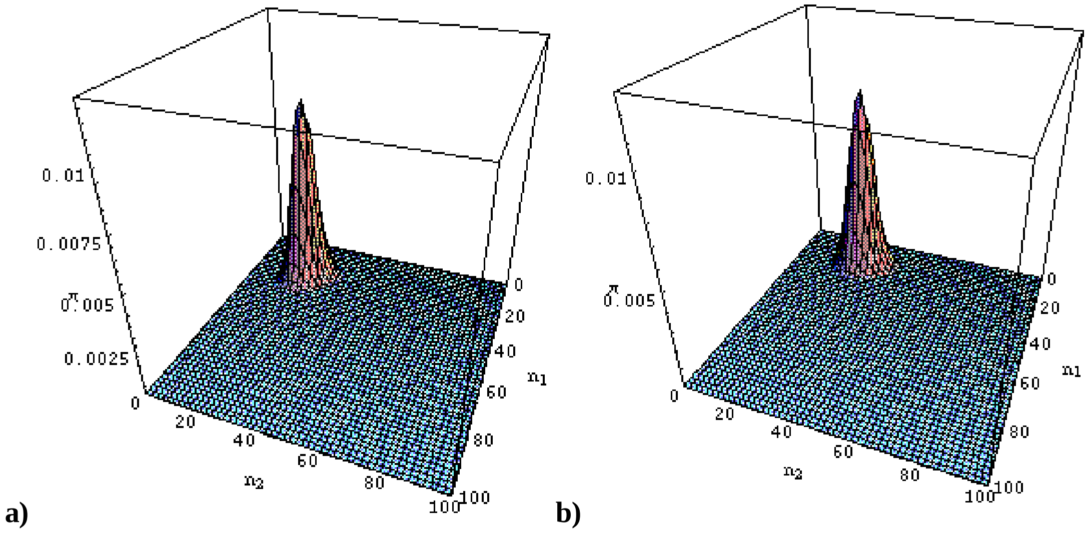


Figure 8.13 The joint probability distribution of the number of agents allocated by the modified SDS to two unequal patterns; (a) N=100, p^+=0.900005, p^-=0.1; (b) N=100, p^+=0.905, p^-=0.1; p_m=p_d=0.0001.

These figures illustrate the sensitivity of the resource allocation of both algorithms to the change in the strength of a parameter p^+ in the proximity of the diagonal $1\text{-}p^-=p^+$. In Chapter 5 a conclusion following from Figure 5.10 (a) and (c) stated that the standard SDS was very sensitive to small change of the parameter corresponding to one of the equally strong patterns. In fact a change of p^+ by 0.000005 caused a significant shift of the joint probability distribution. This has to be compared with Figure 8.11 (a) and Figure 8.13 (a) which illustrate the joint probability distributions of the resource allocation for the modified SDS for the corresponding values of parameters. It is not possible to observe in these graphs any change in the position of the peak of the joint probability distribution. In fact one had to increase the parameter p^+ 1000 fold in order to be able to illustrate an appreciable shift, see Figure 8.13 (b). Thus for very similar patterns the modified SDS allocates more evenly balanced amounts of resources than the standard SDS. However, the fact that the modified SDS detected, by a noticeable shift of the peak of the joint probability distribution, a change in the parameter p^+ of 0.005 implies that the more balanced resource allocation is limited to very similar patterns. Nevertheless, as was observed in the discussion prior to introducing the modified allocation scheme in the case of two patterns similar to the template, one would expect that in some situations one can be interested in detection of sub-optimal solutions, which differ more from the best fit instantiation than in the described case. One can conclude that in such cases the modified SDS will be very strongly biased in the distribution of resources in favour of the stronger pattern and therefore, similarly to the standard SDS, it may not form stable clusters corresponding to sub-optimal solutions.

8.3.2. *Simulations of the context-free SDS.*

In the following the above observations will be illustrated with the simulations of the context-free extension of the modified Stochastic Diffusion Search. The simulations are run with N=100000 agents for 3000 iterations and calculated sample estimates of the activity statistics from the last 2000 data points, discarding first 1000 as a burn-in period.

Figure 8.14 shows the results of simulations in the case of both patterns of equal strength.

Panels (a) and (b) of Figure 8.14 show the behaviour of the context-free modification of SDS for two strong patterns whereas panels (c) and (d) illustrate the corresponding behaviour for weaker patterns. From the inspection of panels (a) and (b) it follows immediately that the algorithm is still in its transient phase. This conclusion is supported by clear serial correlations and large sample standard deviations. The latter are about 8 times bigger than predicted by the theory. The non-stationarity results also in the discrepancy between the expected cluster size and sample averages - the larger of them amounting to about 2033. Although it may seem that the theoretical prediction is not very accurate, at least for the given number of iterations, it is necessary to observe that the relative empirical variation of the cluster sizes, as measured by the sample standard deviations, constitute only about 1% of the total number of agents used in search and the maximal discrepancy between the expected cluster size and the sample average is about 2% of the total number of agents. One can compare Figure 8.14 (a) and (b) with Figure 5.18 showing the corresponding equilibrium evolution of the standard SDS for the same values of parameters. The maximal relative difference between the sample average and the expected cluster size, as seen in Figure 8.14 (a), is about 11% (one has to note, though, that the difference corresponding to the second cluster in Figure 8.14 (b) is about 2% only)- clearly bigger than about 1% of the corresponding difference in Figure 5.18. However, the relative variation given by the sample standard deviation to the sample average is in both cases less than 6%. Moreover, the 14-fold difference between the sample and predicted standard deviations for the standard SDS is much higher than the corresponding 8-fold difference for the context-free extension of the modified SDS. From the comparison it follows that the context-free SDS shows qualitatively similar pattern of behaviour to the standard SDS for given values of parameters.

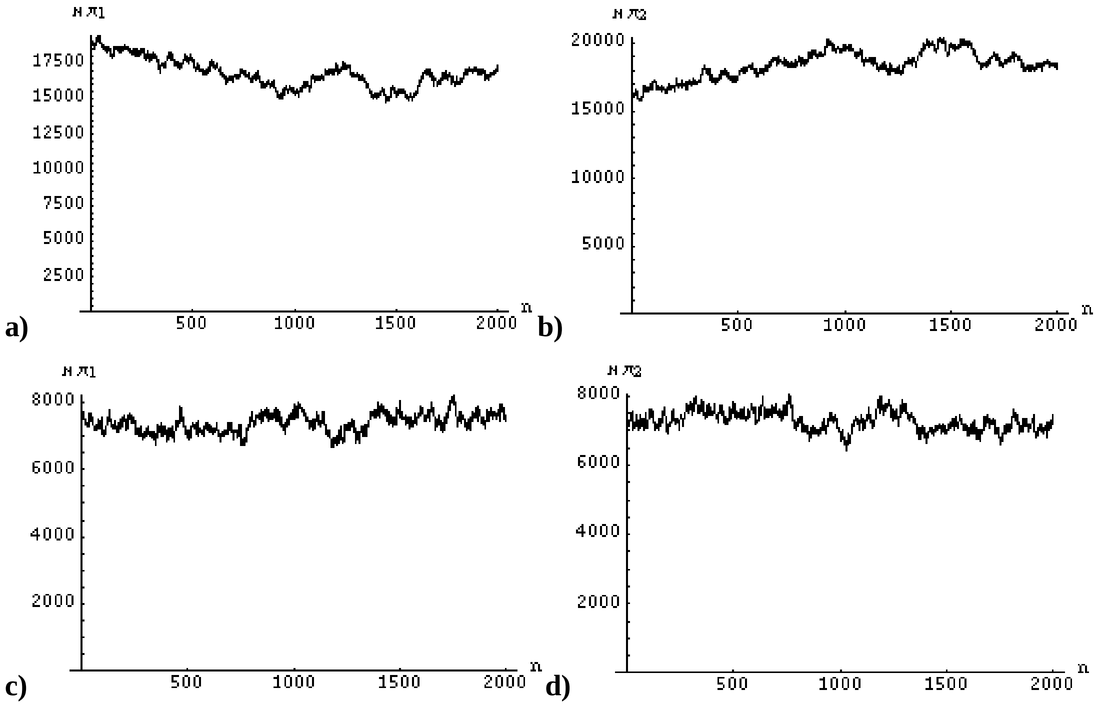


Figure 8.14 The quasi equilibrium evolution of the context-free extension of the modified SDS for two equal patterns; N=100000, p_m=p_d=0.001 and 2000 iterations (of the total 3000 of which first 1000 was discarded as a burn-in period). Panels (a) and (b) correspond to the values of parameters p^+=1-p^-=0.8. The expected, value predicted from the model, in this case is 18820.5 and the standard deviation is 123.61; (a) the evolution of the cluster associated with the first pattern (the sample average is 16787.2 and the sample standard deviation is 1001.24); (b) the evolution of the cluster associated with the second pattern (the sample average is 18416.7 and the sample standard deviation is 1005.9). Panels (c) and (d) correspond to the values of parameters p^+=1-p^-=0.6. The expected value predicted from the model is 8543.05 and the standard deviation is 88.39; (c) the evolution of the cluster associated with the first pattern (the sample average is 7382.52 and the sample standard deviation is 269.302); (d) the evolution of the cluster associated with the second pattern (the sample average is 7285.47 and the sample standard deviation is 274.173).

Overall, one can conclude that the results of the simulations shown in Figure 8.14 (a) and (b) indicate clear non-stationarity of the process rather than an inadequacy of the theoretical prediction.

Panels (c) and (d) show the equilibrium evolution of the context-free SDS in the case of relatively weaker patterns, p^+=1-p^-=0.6. One may note that the relative difference between the expected cluster size and the sample average is still high, amounting to almost 15% in the panel (d), although the empirical relative variability, the ratio of the sample standard deviation to the sample average, is less than 4% and the sample standard deviation is less than 3 times higher than predicted from the theory. These data can be contrasted with the corresponding characteristics of the standard SDS for the same values of parameters (Figure 5.16 (c), (d)) - the relative discrepancy between predicted and sample cluster size of less than 1%, the empirical relative variability of less than 10% and almost 14-fold greater than predicted sample variation. Although the graphs in panels (c) and (d) of Figure 8.14 show a much better concentration around the empirical average than the corresponding graphs of Figure 5.16, nevertheless they still show serial correlations. The latter, together with the high discrepancy between the expected and observed cluster size indicate that the context-free SDS is still in the transient phase.

In general, one could observe in all panels of Figure 8.14 that the sample averages underestimated the expected cluster sizes. Partly one can account for this by referring to the clearly non-stationary behaviour of the context-free SDS in the corresponding graphs. However, graphs corresponding to the weaker patterns, $p^+=1-p^-=0.6$ show less variability around the sample averages which are nevertheless still underestimating the predicted cluster size. The reason for the underestimation of the empirical averages may be in, related to non stationarity, the weak mixing of the algorithm. The mixing, in informal terms, characterises how the stochastic process samples its state space. Processes which visit all the states in the state space mix better than the ones which, in the same time period, evolve in the subset of the state space. In the case of the context-free SDS it is clear that it very quickly converges to the quasi equilibrium (the unique strong peak of the joint probability distribution) and remains virtually always in its vicinity. Taking into account that the context-free SDS, similarly to the standard algorithm, evolves on a two dimensional integer valued grid with maximal co-ordinates equal to N - the total number of agents, one can see that majority of the state space is not visited in the simulations. In fact a similar observation could have been made also with respect to the standard SDS although in the former case it is much more pronounced because the maximal amount of resources that can be allocated by the modified SDS is about 50% of the total number of agents. Thus the modified SDS ‘lives’ most of the time in a small fraction of a quarter of the state space. Thus, in particular, the states with the co-ordinate values higher than 50% of N are not taken into account while counting the sample averages. However, the expected values are obviously calculated over the whole state space and although the contribution of single states with the co-ordinates higher than 0.5N can be very small nevertheless their total influence can amount to the visibly higher, than sample averages, expected value.

Figure 8.15 shows the quasi equilibrium behaviour of the context-free SDS in the case of unequal patterns.

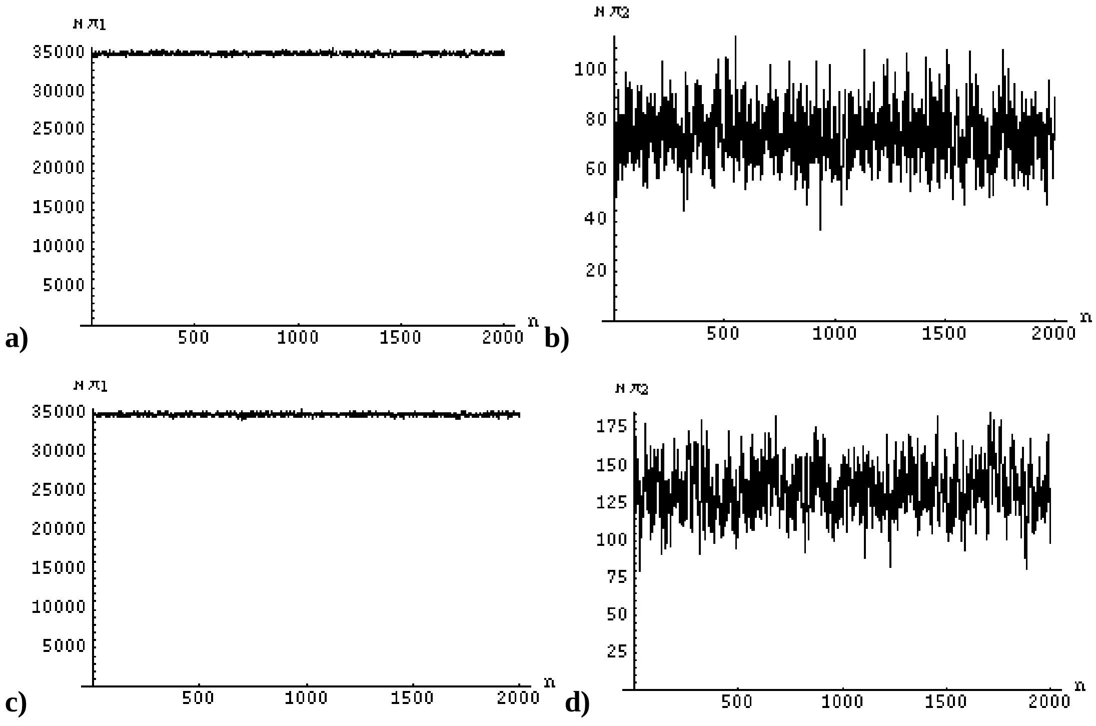


Figure 8.15 The quasi equilibrium evolution of the context-free extension of the modified SDS for two unequal patterns; N=100000, $p_m=p_d=0.001$ and 2000 iterations (of the total 3000 of which first 1000 was discarded as a burn-in period); (a) the evolution of the cluster associated with the best fit pattern for $p^-=0.2$ and $p^+=0.5$ (the expected value is 37500.1, the standard deviation is 153.1, a sample average is 35053.5 and a sample standard deviation is 169.4); (b) the quasi equilibrium cluster associated with the second pattern for $p^-=0.2$ and $p^+=0.5$ (the expected value is 70.65, the standard deviation is 8.4, a sample average is 74.56 and a sample standard deviation is 10.47). The evolution of the clusters associated with the two unequal patterns corresponding to $p.^-=0.2$ and $p^+=0.6$ is shown in panels (c) and (d). For a stronger pattern, the panel (c), the expected value is 37443.9, the standard deviation is 153.04, the sample average is 34987 and a sample standard deviation is 172.29), and for a weaker pattern, the panel (d), the expected value is 126.98, the standard deviation is 11.26 and the sample average is 134.68 and the sample standard deviation is 16.84).

In this figure one can see the allocation of resources in the context-free SDS when the best fit pattern is very strong, $p^-=0.2$ panels (a) and (c), and the second pattern is either strong, $p^+=0.6$, or a boundary case between strong and weak patterns, $p^+=0.5$ (see Figure 8.9).

In all cases the context-free SDS formed stable clusters, although still the clusters corresponding to the weaker patterns are several orders of magnitude smaller. Thus it follows from the model that increasing the strength of the second pattern by 20% almost doubled the size of the corresponding cluster. Also it is interesting to compare these data with the corresponding values for the standard SDS, see Figure 5.13 (b), (d) of Chapter 5. In the latter case one can observe a similar increase of the relative size of the cluster associated with the second pattern in response to the appropriate change of the parameter p^+, although it is worth noting that the corresponding clusters for the context-free SDS are about 10 times bigger than for the standard algorithm. It follows that the context-free SDS allocates more resources to the weaker solutions than the standard SDS.

Nevertheless, the corresponding theoretical increase relative to the appropriate best fit pattern cluster size is only from 0.19% to 0.34%. The sample estimates match very well their theoretical counterparts. Thus, the steady state distribution of agents is still strongly biased towards the best fit solution if the difference between the solutions is relatively high.

One can note that the best fit pattern clusters seem to be very stable and well concentrated around the empirical estimates which are again consistently smaller than the expected values, the discrepancy relative to the theoretical cluster size amounting to about 6.5% or about 2.5% of all agents. Also one can see that their size constitute approximately half of the corresponding clusters of the classical SDS, see Figure 5.13 (a), (d), therefore the distribution of the agents in the context-free SDS cannot be obtained from that of the standard SDS simply by an appropriate rescaling by a constant factor. In fact, for the given values of all parameters, it follows from the corresponding models that the standard SDS allocates roughly 75% of resources to the best fit pattern and of order of 10^{-4} % to the second solution, whereas the corresponding figures for the context-free SDS relative to the maximal allowable cluster size are about 75% for the best fit pattern and 0.15% for the weaker second solution and 0.25% for the stronger one.

The overall conclusion is that although the steady state distribution of resources in the context-free SDS is still strongly biased towards the stronger solution, however this algorithm allocates significantly more agents to weaker patterns and thus enables more easily than the standard SDS recognition of the sub-optimal solutions without substantial impairment of the signal to noise sensitivity.

In the final simulation of the context-free SDS, depicted in Figure 8.16, one can see the quasi equilibrium distribution of agents among two patterns, the stronger one with a fixed parameter

p^-=0.5, panels (a) and (c), and the second one with an increasing strength, reflected by an increase of the parameter p^+ from 0.42, the panel (b), to 0.48, the panel (d).

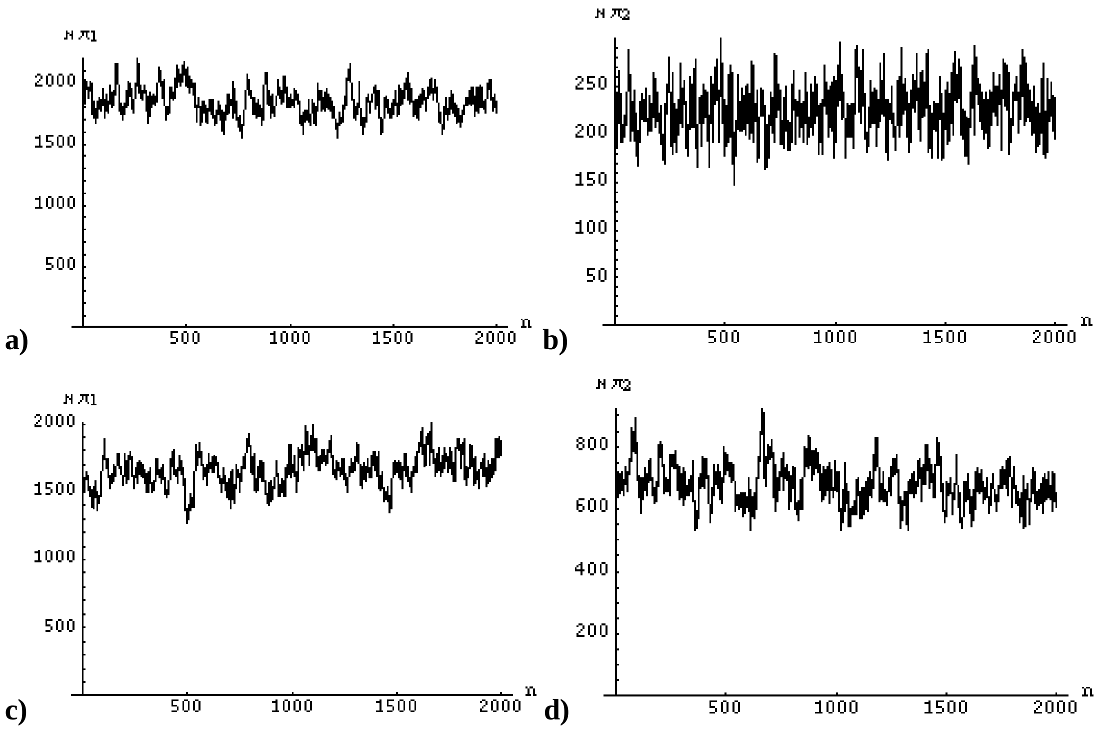


Figure 8.16 The quasi equilibrium evolution of the context-free SDS for two unequal patterns; N=100000, $p_m=p_d$=0.001 and 2000 iterations (of the total 3000 of which first 1000 was discarded as a burn-in period); (a) the evolution of the cluster associated with the best fit pattern for p^-=0.5 and p^+=0.42 (the expected value is 2075.8, the standard deviation is 45.09, the sample average is 1848.42 and the sample standard deviation is 111.38); (b) the quasi equilibrium cluster associated with the second pattern for p^-=0.5 and p^+=0.42 (the expected value is 223.7, the standard deviation is 14.94, the sample average is 224.75 and the sample standard deviation is 23.74). The evolution of the clusters associated with the two unequal patterns corresponding to p^-=0.5 and p^+=0.48 is shown in panels (c) and (d). For a stronger pattern, the panel (c), the expected value is 1854.31, the standard deviation is 42.66, the sample average is 1658.12 and the sample standard deviation is 116.92), and for a weaker pattern, the panel (d), the expected value is 706.25, the standard deviation is 26.48 and the sample average is 679.72 and the sample standard deviation is 62.12).

Again one can observe in the panels (a) and (c) a systematic underestimation by 10-11% of the theoretical expected cluster size for the best fit pattern by the sample averages, although the agreement between the corresponding quantities for the weaker patterns is very good (0.5-3.76% relative difference). One can see also serial correlation in the graphs describing stronger patterns. In all four panels the empirical relative variation ranges between 6% and

10.6% of the average cluster size. In comparison, the standard SDS, Figure 5.14 (a), (b) and Figure 5.15 (c) and (d), shows much better match between the theoretical average cluster sizes and their sample estimates with the maximal relative difference smaller than 2% although the empirical relative variation changes in the same range as for the context-free SDS.

The most interesting feature of Figure 8.16 is the distribution of resources between the patterns. For p^-=0.5 and p^+=0.42 the context-free SDS allocates 4.15% of the maximal possible cluster size to the best fit pattern and 0.45% to the second solution and for the same p^- and p^+=0.48 the corresponding figures are 3.32% and 1.4% respectively. The corresponding resource allocation of the standard SDS is 2.95%, 0.23% for p^-=0.5 and p^+=0.42 and 2.68%, 0.79% for p^-=0.5 and p^+=0.48. One can see that the ratios of the resources allocated to the two patterns by the standard and the context-free SDS are approximately the same for both sets of parameters and change from about 10 fold bias towards the best fit pattern to roughly about 3 fold difference. Although the standard SDS allocates more agents to both patterns especially for the best fit solution, however it is worth noting that the context-free SDS allocates relatively about two times more resources to both patterns than the standard SDS in all four cases.

Overall, it seems that the context-free extension of the modified SDS distributes its resources more evenly than the standard SDS but maintains a comparable stability of clusters supporting solutions. It is more sensitive also to the relative strength of the solutions but retains similar signal to noise ratio sensitivity. However, it seems that a less biased, towards the best fit, resource allocation is also limited to relatively very similar patterns. Thus, the main difference between the standard and context-free SDS is the allocation of a substantial ratio of total resources for the exploration of the search space.

The notes under the definition of the context-free extension of the modified SDS mentioned that this algorithm used only the overall activity information in order to modify cluster sizes.

It is, however, possible to introduce further changes which will take into account more detailed information during formation and maintenance of steady state clusters. Below one such algorithm will be introduced and discussed.

8.4. *Context-sensitive extension of the modified resource allocation.*

In the definition of the context-free extension of SDS one preserved the property of the modified SDS that, in the diffusion phase, the activity of an agent was dependent on the overall activity of the whole population. This, translated into a more-than-one pattern case, meant that the activity of agents in all possible clusters could affect directly (though probabilistically) the activity status of any active agent.

The starting point to the new definition will be the same property of the modified SDS, although it will acquire a different interpretation in the context of many patterns, in the search space, similar to the template. Namely, it will be assumed that in the diffusion phase, the status of a given agent activated by a pattern will only be directly affected by the overall activity associated with the same pattern.

As previously, a discussion of the new extension will concentrate first on the scenario with two patterns in the search space similar to the template. The corresponding model is analogous to that previously introduced for the context-free extension of SDS and thus the probability transition matrix associated with a one step evolution of a single agent is given by

$$P = \begin{array}{c|ccc} & (a_m, m) & (a_d, m) & (n,*) \\ \hline (a_m, m) & p_{11} & p_{12} & 1 - p_{11} - p_{12} \\ (a_d, m) & p_{21} & p_{22} & 1 - p_{21} - p_{22} \\ (n,*) & p_{31} & p_{32} & 1 - p_{31} - p_{32} \end{array} \tag{8.5}$$

where

$$\begin{aligned} p_{11} &= \left(1 - \tfrac{m_1}{N}\right)\left(1 - p^-\right) + \tfrac{m_1}{N} p_m \left(1 - p^-\right), \\ p_{12} &= \tfrac{m_1}{N} p_d p^+, \\ p_{21} &= \tfrac{m_2}{N} p_m \left(1 - p^-\right), \\ p_{22} &= \left(1 - \tfrac{m_2}{N}\right)p^+ + \tfrac{m_2}{N} p_d p^+, \\ p_{31} &= \tfrac{m_1}{N}\left(1 - p^-\right) + \left(1 - \left(\tfrac{m_1}{N} + \tfrac{m_2}{N}\right)\right)p_m \left(1 - p^-\right), \\ p_{32} &= \tfrac{m_2}{N} p^+ + \left(1 - \left(\tfrac{m_1}{N} + \tfrac{m_2}{N}\right)\right)p_d p^+, \end{aligned} \tag{8.6}$$

and m_1, m_2 denote, as before, the size of populations of active agents concentrated on corresponding patterns similar to the template.

Notes.

1. The above matrix is very similar to the definition (8.4). Indeed the dependence on the parameters is of similar form and the last row is identical as in the previous case.

2. The main difference between the context-free SDS and this definition is that in the former case one had self maintenance only of the overall activity level, whereas in the latter - all

clusters are self-stabilising. This, of course, does not imply though that they are independent. In fact, they still interact, although the change of the active agent's status depends here only on the other agents forming the same cluster.

3. Although the change in the formulas defining (8.5)-(8.6) is minor, it nevertheless translates into a quite different interpretation in information processing terms. Namely, in order for the active agents to operate following the new definition, information about the general status of other agents (i.e. implicit 'knowledge' of the overall activity in the system) is not sufficient. Active agents need access to information about the identity of the particular clusters in order to perform an appropriate operation. However, the identity of the clusters is in fact defined in terms of the information they process. Thus it follows that active agents evolution is conditioned by the information they attend to. Therefore the future evolution of active agents is context-dependent, hence this extension of the Stochastic Diffusion Search will be called the 'context-sensitive' SDS.

4. The sensitivity to the context as defined above is a direct extension of the characteristic feature of the modified SDS and therefore it is very simple. However, it is possible to define many other extensions in which context sensitivity would be more intricate.

5. One can contrast this mode of operation with that of the traditional neural models. As one remarked, in note 2 below the definition of the context-free SDS, active agents operated (similarly to standard connectionist models) on the general levels of activity. This corresponds to model neurons discarding any information that could be encoded in the particular pattern of activity. Of course, the overall level of activity is most likely correlated to some extent with the carried message regardless of the particular information encoding scheme. Thus, models based on such 'external' properties of the information carrier stand a chance of capturing the overall signal processing characteristics of the biological nets. This could be especially true in cases, where the correlation between the message content and its presence is strong, e.g. in low level sensory processing. However, in cases, when the correlation between the content and the encoding it level of activity is weaker, one could expect that the models based on the overall activity correspond more crudely to the actual information processing. This would be particularly true, if some part of the encoded message cannot be captured in the level of activity (mean firing rate) but is carried in its particular structure. This is because that part of information encoded in the structure of the message carrier has to influence the way the activity is propagated through the system, at least in some stages of processing. Otherwise, it would have no effect on the

dynamics of the system, and so it would bear no significance, or meaning, and thus be ephiphenomenal. The more the processing of given information is context-dependent, the more the dynamics of a system discarding this context differs from the dynamics of a system taking into account contextual information. The above reasoning suggests that in such situations algorithms, like the context-sensitive SDS, taking into account semantic information are more appropriate for modelling information processing. It seems that, in the case of the information processing in the nervous system this is the case particularly in higher level processing, although it is not possible to draw a clear boundary between the context-free and context-sensitive levels of processing.

6. Although the interpretation of the two extensions of the modified SDS in terms of context sensitivity seems to be very appealing, however one has to stress that this interpretation is relative to the underlying information encoding in the system. The context sensitivity of the current modification of SDS follows from the fact that active agents use effectively not only the general information about the activity but also specific information about the identity of the particular clusters. The claim, that the context-sensitive SDS processes semantic information, whereas standard connectionist models discard this information, rests on the assumption that there is a strong correspondence between particular structure, or pattern, of activity and information content, and such systematic correspondence between the level of activity and semantics is either limited or absent (at least in some situations). In fact, even the context-sensitive SDS may be interpreted in formal signal processing terms without referring to the semantics at all (the latter would follow from the above mentioned correspondence). However, the interpretation proposed here forces explicit consideration of the, often implicit, assumptions about the information encoding and its influence, reflected in the dynamics of the system, on the information processing. It thus can be concluded that this interpretation is useful, however it should be used with care.

One has to solve standard fixed point equations, analogous to (8.3), in order to analyse the steady state behaviour of the context-sensitive SDS.

These equations can be rewritten as a system of two non-linear equations with two unknowns

$$\begin{aligned}\pi_1 &= \left(1-p^-\right)\left[-2\left(1-p_m\right)\pi_1^2+2p_m\pi_2^2+\left(2p_m-1\right)\pi_1\pi_2-2p_m\pi_2+2\left(1-p_m\right)\pi_1+p_m\right],\\ \pi_2 &= p^+\left[2p_d\pi_1^2-2\left(1-p_d\right)\pi_2^2+\left(2p_d-1\right)\pi_1\pi_2+2\left(1-p_d\right)\pi_2-2p_d\pi_1+p_d\right]\end{aligned} \tag{8.7}$$

where π_1 and π_2 are the corresponding equilibrium probabilities of finding an agent activated on one of the two patterns.

Notes.

1. In order to relate the above extension of the modified SDS to the context-free SDS one can define the evolution of the probability distribution π in terms of a non-linear mapping. Thus, one can write

$$\begin{aligned} \pi_1^{n+1} &= f_1(\pi_1, \pi_2), \\ \pi_2^{n+1} &= f_2(\pi_1, \pi_2), \end{aligned}$$

   where f_1 and f_2 are defined by the right hand sides of (8.7).

   The current extension of the modified SDS is related to the context-free SDS in the following way

$$\begin{aligned} f_1(\pi_1, \pi_2) &= f_1^{CF}(\pi_1, \pi_2) - (1 - p^-)(2p_m - 1)\pi_1\pi_2, \\ f_2(\pi_1, \pi_2) &= f_2^{CF}(\pi_1, \pi_2) - p^+(2p_d - 1)\pi_1\pi_2 \end{aligned}$$

   where f_1^{CF} and f_2^{CF} denote the corresponding mappings of the context-free SDS.

2. For p_m=p_d, the above extended algorithm also enjoys symmetry analogous to that of the context-free and standard SDS. Thus

$$\begin{aligned} f_1(\pi_1, \pi_2)\frac{1}{1 - p^-} &= f_2(\pi_2, \pi_1)\frac{1}{p^+}, \\ f_2(\pi_1, \pi_2)\frac{1}{p^+} &= f_1(\pi_2, \pi_1)\frac{1}{1 - p^-} \end{aligned}$$

8.4.1. *Characterisation of the properties of the context-sensitive SDS.*

From the solution of (8.7), obtained in Mathematica, the surface describing the dependence of the allocation of agents to the two patterns was plotted. The corresponding graphs are shown in Figure 8.17.

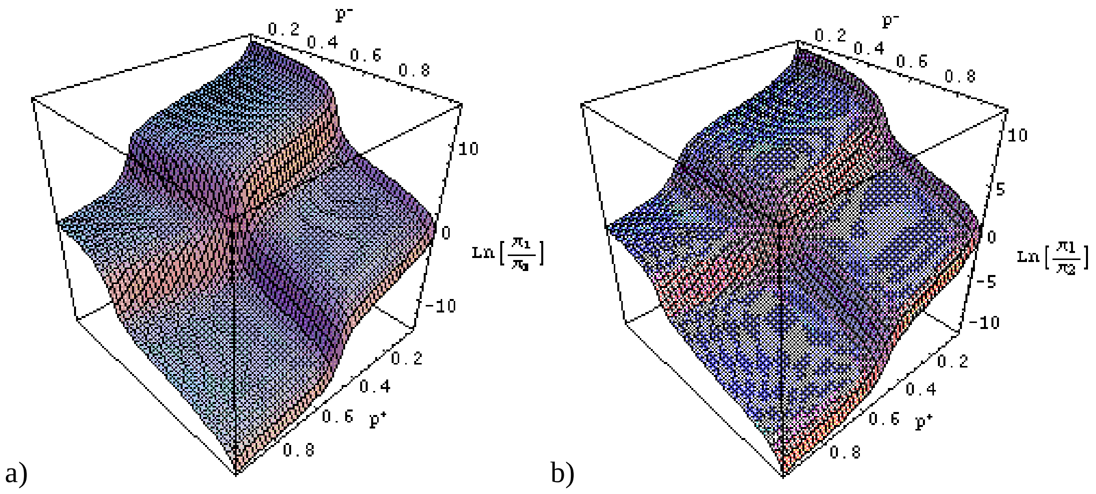


Figure 8.17 The surface describing an average difference of orders of magnitude of resources allocated to both patterns by the context-sensitive SDS, as a function of parameters p^- and p^+, (a) p_m=p_d=0.0001, (b) p_m=p_d=0.001.

In contrast to the corresponding Figure 8.9, which showed a striking resemblance to Figure 5.7, one can immediately notice in Figure 8.17 that the context-sensitive SDS changes dependence of the allocation of resources on parameters p^- and p^+ to a much greater extent than the context-free extension although all three figures bear some overall similarities.

The surface in Figure 8.17 is, like the one corresponding to the context-free or standard SDS, symmetric along the line parallel to the diagonal $1-p^-=p^+$ and corresponding to equal orders of magnitude of allocated resources. Also areas of the surface corresponding to at least one weak pattern (i.e. $p^->0.5$ or $p^+<0.5$) are in practice indistinguishable from the corresponding areas in Figure 8.9. This implies that for a range of parameters the steady state allocation of resources is qualitatively similar to that of the context-free SDS. Thus, one can expect similar, to the latter algorithm, distribution of agents between two patterns if at least one of them is weak. Also it follows, from the range of heights spanned by the graph in Figure 8.17, that for these areas the difference between the amounts of agents allocated to both patterns for fixed values of the parameters is more similar to that of the context-free SDS rather than the standard Stochastic Diffusion Search.

Another similarity of Figure 8.17 to Figure 8.9 and Figure 5.7 is the change of the surface following change of the parameter p_m=p_d, as seen in panels (a) and (b) of all these figures.

In contrast, however, to both the context-free and the standard SDS, the surface in Figure 8.17 has a well defined fourth area roughly corresponding to both strong patterns, i.e. $p^-<0.5$ and

$p^{+} > 0.5$. In the standard SDS, for example, there existed a very sharp transition between the areas in which both patterns were strong but unequal. This reflected the fact that this algorithm was very strongly biased towards the best fit solution, allocating resources more evenly only to very similar patterns. In the case of the context-sensitive SDS, the sharp transition in the corresponding part of Figure 8.17 is replaced by a very flat area on the null level. This means that the resource allocation in the context-sensitive SDS is much less biased towards the best fit pattern and strong solutions differing considerably from each other should be supported by comparable orders of magnitude of agents in the corresponding clusters. This feature of the context-sensitive SDS is important for finding all strong sub-optimal solutions. Although the context-free SDS was capable of more even distribution of resources to unequal solutions than the standard SDS, nevertheless both of these algorithms would in practice form stable clusters corresponding only to the best solutions. This results from the constraint of finite resources and best fit bias. In contrast, from the inspection of Figure 8.17 one would expect that even with finite resources the context-sensitive SDS would effectively support different solutions provided that all of them are sufficiently strong. This seems a heuristically reasonable feature, as it is equivalent to placing different criteria for the strong and the weak patterns. The weak patterns are processed in a way similar to that of the context-free or the standard SDS, which is characterised by the very high amplification of the signal to noise ratio. On the other hand, all strong patterns are potentially interesting and thus the context-sensitive SDS tries to discover and support all such solutions.

Thus in conclusion, of all three algorithms, only the context-sensitive SDS allows for parallel processing of many solutions, or in other words for multitasking.

The surface in Figure 8.17 formed two regions of abrupt transition from the ‘fair’ to the biased resource allocation areas similar to the transitions between the areas corresponding to at least one pattern being strong and both weak patterns. However, the transitions between different modes of resource allocation in Figure 8.17 do not follow lines of constant parameters, but are dependent on the parameter. Namely, it seems that these transitions show weak linear dependence in the difference between the strength of both patterns. Also the magnitude of the change in the resource distribution along these transitions is much smaller than it was the case for the transition between areas corresponding to both strong, unequal patterns in Figure 8.9 or 5.7.

From the solution of the set of equations (8.7) it follows that the context-sensitive SDS will also allocate, like the context-free algorithm, equal numbers of agents to equally strong patterns. It will also always maintain the population of agents exploring the search space. However, in contrast to the previous algorithm, the amount of resources allocated to two ideal instantiations of the template in the search space is greater than the size of the maximal possible cluster allocated to a single pattern.

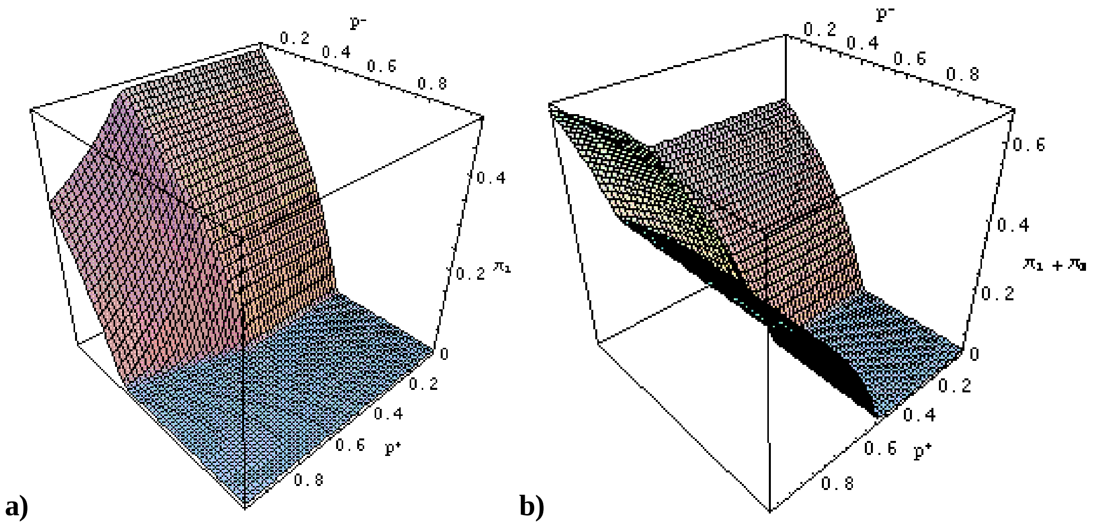


Figure 8.18 The equilibrium probability of the context-sensitive SDS of finding an agent activated by the first pattern, the panel (a), and the total probability of finding an agent activated, i.e. a sum of π_1 and π_2, the panel (b); $p_m=p_d=0.0001$.

This can be seen also in Figure 8.18 showing the equilibrium probability of activation of an agent by one, the panel (a), or by any of the patterns, the panel (b).

In Figure 8.18 (a) one can see that the probabilities of finding an agent activated by a given pattern are smaller for the context-sensitive than the context-free SDS in the region of the diagonal $1-p^->p^+$, $p^+>0.5$, compare Figure 8.10 (a), but this relationship is reversed in the vicinity of the diagonal for $1-p^-<p^+$, $p^+>0.5$. Also it is clear that, in those regions, for a given set of parameters p^-, p^+ the total steady state probability of activating an agent is significantly greater for the context-sensitive SDS than for the context-free algorithm, see panels (b) of the corresponding figures.

Figure 8.17 and Figure 8.18 allowed one for a qualitative description of the resource allocation characteristics of the context-free SDS, independently of the total amount of agents. In order to characterise the distribution of agents in more detail the remainder of this sub-section will present, similarly to the previous case, the joint probability distributions of agents between two patterns for chosen sets of parameters.

Figure 8.19 shows the joint probability distribution of the allocation of resources to the two patterns of equal strength.

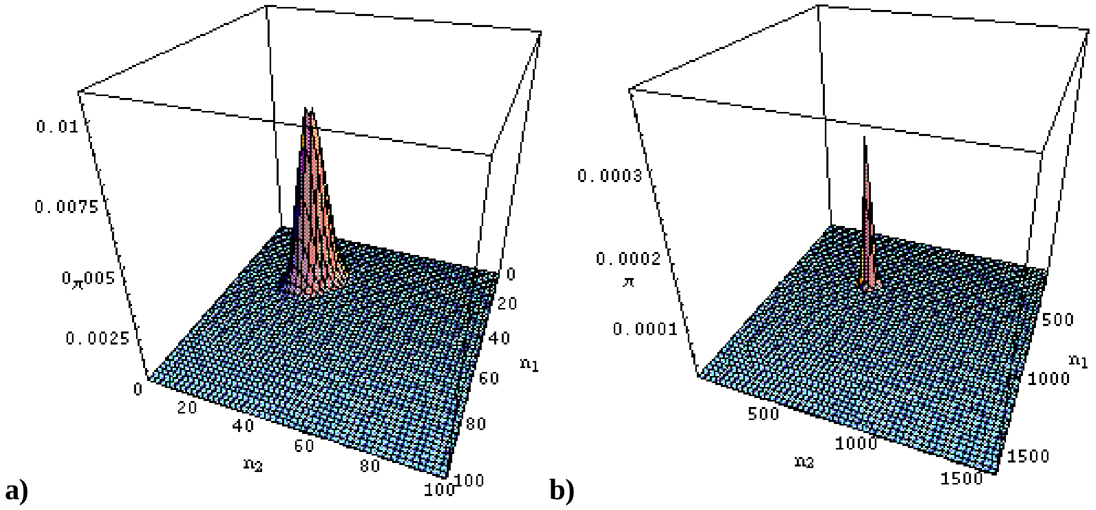


Figure 8.19 The joint probability distribution of the number of agents allocated by the context-sensitive SDS to two patterns of equal strength; (a) N=100, $p^+=1-p^-=0.9$; (b) N=100000, $p^+=1-p^-=0.5$; $p_m=p_d=0.0001$.

One can see in Figure 8.19 that indeed the joint probability distribution of the context-sensitive SDS has a unique sharp peak around equal cluster sizes for equally strong patterns. Comparing Figure 8.19 with Figure 8.11 and Figure 5.11 (a) and Figure 5.10 (a) one can draw a conclusion that the context-sensitive SDS distributes equal amounts of resources to equal patterns in a manner similar more to the context-free extension than to the standard SDS.

The joint probability distributions in cases of unequal patterns are depicted in Figure 8.20 and Figure 8.21.

One can see immediately that panels (a), (b) and (d) of Figure 8.20 resemble the corresponding panels of Figure 8.12 depicting the joint probability distributions of the context-free SDS for the same values of parameters. In particular it follows that for two significantly different patterns, as in panels (a) and (d), the distribution of agents in the context-sensitive SDS does not differ substantially from that of the context-free SDS or the standard Stochastic Diffusion Search. Also, as before, a significant increase of the weaker pattern may not shift appreciably the peak of the joint probability distribution if the resulting pattern is still significantly weaker than the best fit solution.

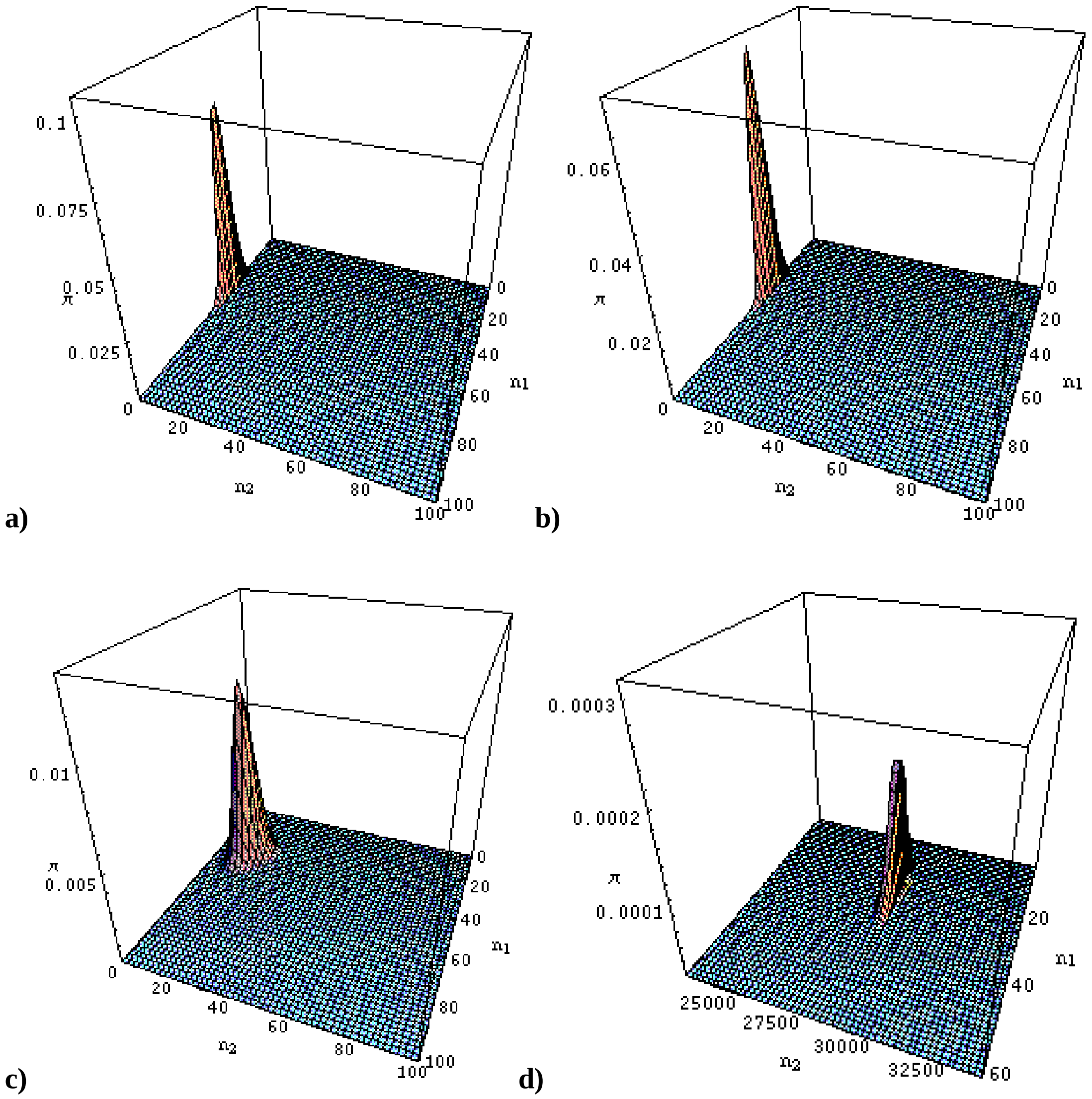


Figure 8.20 The joint probability distribution of the number of agents allocated by the context-sensitive SDS to two unequal patterns; (a) N=100, p^+=0.2, p^-=0.2; (b) N=100, p^+=0.6, p^-=0.2; (c) N=100, p^+=0.7, p^-=0.2; (d) N=100000, p^+=0.7, p^-=0.5; p_m=p_d=0.0001.

However, one can note a different behaviour of the context-sensitive SDS, already mentioned during a discussion of Figure 8.17, by comparing panels (c) of Figure 8.20 and Figure 8.12. Namely, one can see in the appropriate panels that the context-sensitive SDS can allocate comparable amounts of agents to considerably different patterns. This resource distribution reflects the fact that the two patterns are sufficiently similar to the template to be processed in parallel. One may note that still the distribution is slightly biased towards the stronger solution thus it retains the information about the relative strength of the patterns. It is worth stressing that both the standard SDS and its context-free extension would form, in practice, in a similar situation only one cluster corresponding to the best fit solution only due to limited resources and a strong bias. The context-free SDS had relatively smaller bias than the standard Stochastic Diffusion Search, however it was limited to much more similar patterns. The more fair distribution of resource between two unequal patterns results in lesser sensitivity of the context-sensitive SDS to changes in the strength of a pattern, which can be appreciated in Figure 8.21.

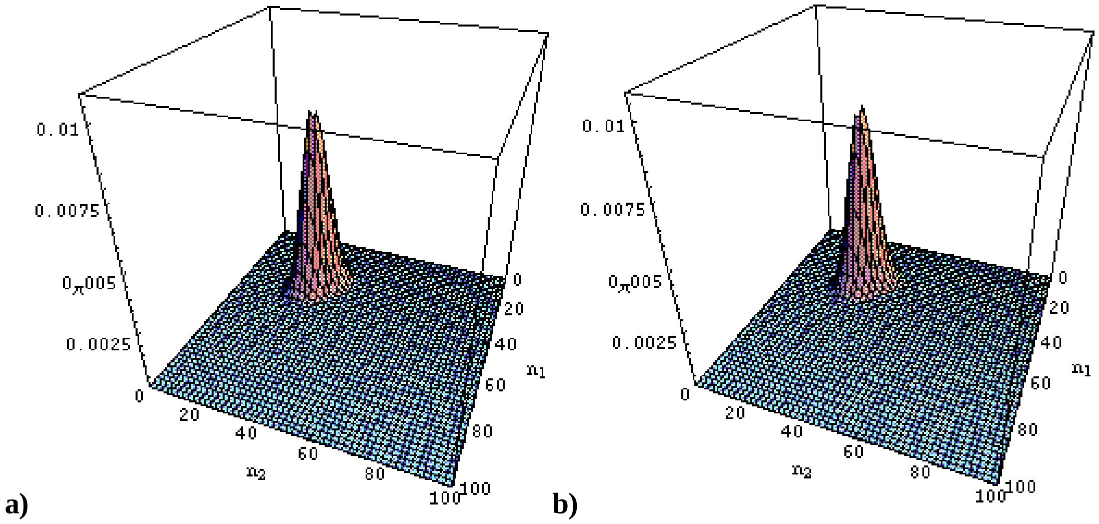


Figure 8.21 The joint probability distribution of the number of agents allocated by the context-sensitive SDS to two unequal patterns; (a) N=100, p^{+}=0.900005, p^{-}=0.1; (b) N=100, p^{+}=0.905, p^{-}=0.1; p_m=p_d=0.0001.

This figure should be compared with Figure 8.11 (a), Figure 8.13 and Figure 5.10 (a) and (c) showing the shift in distribution of resources in response to strengthening one of the patterns by two other algorithms. Clearly, of all three, the context-sensitive SDS is the least sensitive to the small changes of the parameter reflecting strength of one of the patterns - there is no noticeable shift in the peak of the joint probability distribution in Figure 8.21.

8.4.2. Simulations of the context-sensitive SDS.

The discussion of the joint probability distributions, depicted in Figure 8.19 - Figure 8.21, allowed one to infer more detailed properties of the context-sensitive SDS. However, in order

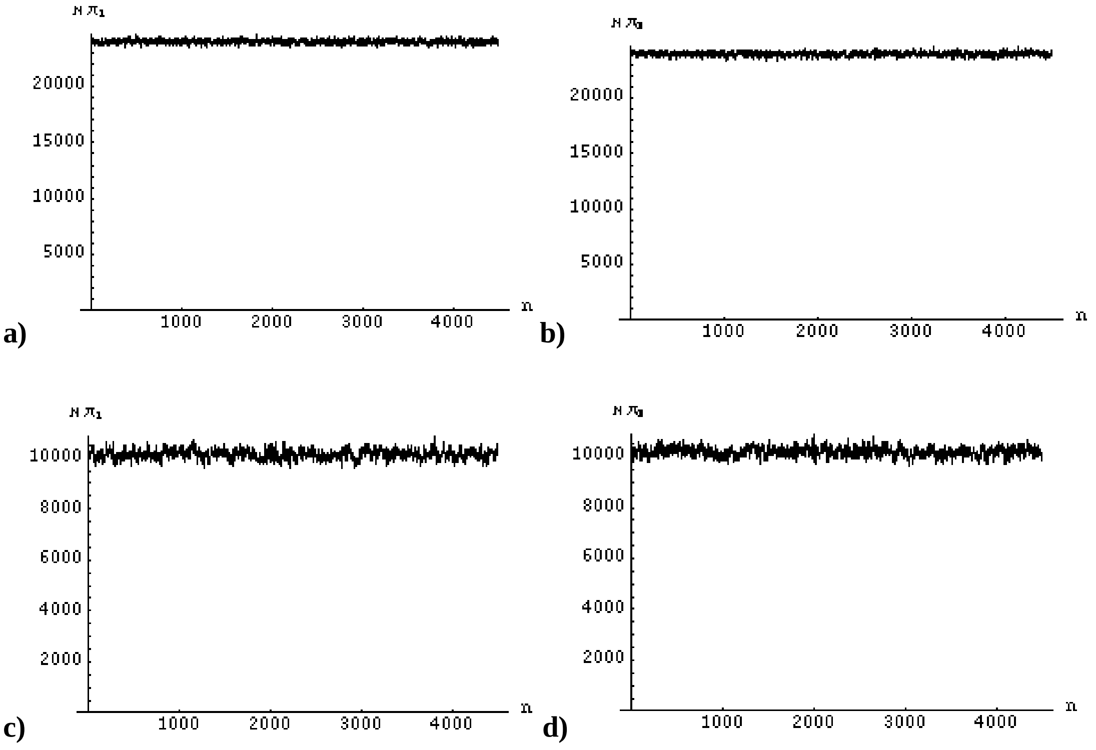


Figure 8.22 The quasi equilibrium evolution of the context-sensitive SDS for two equal patterns; N=100000, $p_m=p_d=0.001$ and 2000 iterations (of the total 3000 of which first 1000 was discarded as a burn-in period). Panels (a) and (b) correspond to the values of parameters $p^+=1-p^-=0.8$. The expected, value predicted from the model, in this case is 25049.8 and the standard deviation is 137; (a) the evolution of the cluster associated with the first pattern (the sample average is 23985.9 and the sample standard deviation is 172.59); (b) the evolution of the cluster associated with the second pattern (the sample average is 23973.5 and the sample standard deviation is 176.19). Panels (c) and (d) correspond to the values of parameters $p^+=1-p^-=0.6$. The expected value predicted from the model is 11295.5 and the standard deviation is 100.09; (c) the evolution of the cluster associated with the first pattern (the sample average is 10142.2 and the sample standard deviation is 176.58); (d) the evolution of the cluster associated with the second pattern (the sample average is 10193.4 and the sample standard deviation is 173.35).

to see how it behaves in concrete settings the simulations, analogous to the ones run for the previous two algorithms, will be discussed . Again the context-sensitive SDS is run with N=100000 agents and all the statistics are calculated from the last 2000 iterations, discarding first 1000 as a burn-in period.

Figure 8.22 shows the results of simulations in the case of both patterns of equal strength.

One can notice that for both sets of parameters, 1-p^-=p^+=0.8 in panels (a), (b) and 1-p^-=p^+=0.6 in panels (c), (d), the context-sensitive SDS, like the context-free or standard SDS, forms equal clusters corresponding to equally strong patterns. One can also note the systematic relative difference between the expected cluster sizes and their empirical estimates amounting to slightly above 4% in both panels corresponding to the values of parameters 1-p^-=p^+=0.8 and about 10% in panels (c) and (d). This behaviour is similar to the one observed in the case of the context-free SDS and most likely can be accounted for by the bias of the empirical averages caused by taking into account only a small fraction of the state space corresponding to relatively low sizes of clusters. However, one can note also here that these differences are in 3 out of 4 cases smaller for the context-sensitive SDS than for the context-free algorithm and show high consistency across the panels showing clusters corresponding to equally strong patterns. This suggests that in cases depicted in Figure 8.22 the algorithm achieved closer proximity of the steady state than was the case of the context-free SDS in Figure 8.14, where the relative differences between the empirical estimates and theoretical prediction of the clusters sizes were more variable. This conclusion seems to be supported also by the smaller relative variation of the cluster sizes in Figure 8.22 varying from below 1% in panels (a) and (b) to below 2% in panels (c) and (d) which is much smaller than the corresponding levels of the relative variation of the context-free or the standard SDS. Also it is worth noting both the context-free and the standard SDS showed higher, than the context-sensitive SDS, differences between the empirical and predicted standard deviations, the differences ranging from almost 3- to about 8-fold in the case of the context-free SDS and about 14-fold for the standard SDS. The corresponding maximal difference for the context-sensitive SDS is below 2-fold.

All the above suggests that, of all three algorithms, the context-sensitive SDS first went through the phase of fast transients and achieved a quasi equilibrium. However, in order to obtain a better match between the empirical averages and the predicted average cluster sizes, because of a bias, one would have to collect data from a much larger sample.

It is worth noting that the context-sensitive SDS, although, similarly to the two previous algorithms, distributes equal amounts of agents between two equal patterns, nevertheless its distribution is different to both context-free and the standard SDS. Namely, it allocates about 50% of the maximal single cluster size to both patterns with 1-p^-=p^+=0.8, and about 23% to both patterns with 1-$p.^-$=$p.^+$=0.6. This can be compared with about 37% and 17% of resources allocated, by both context-free and the standard SDS, correspondingly to patterns with 1-p^-

$=p^+=0.8$ and $1-p^-=p^+=0.6$. Thus in cases of two equal patterns the context-sensitive SDS seems to allocate to them relatively more resources than the other two algorithms.

Figure 8.23 shows the quasi equilibrium behaviour of the context-sensitive SDS in the case of two unequal patterns.

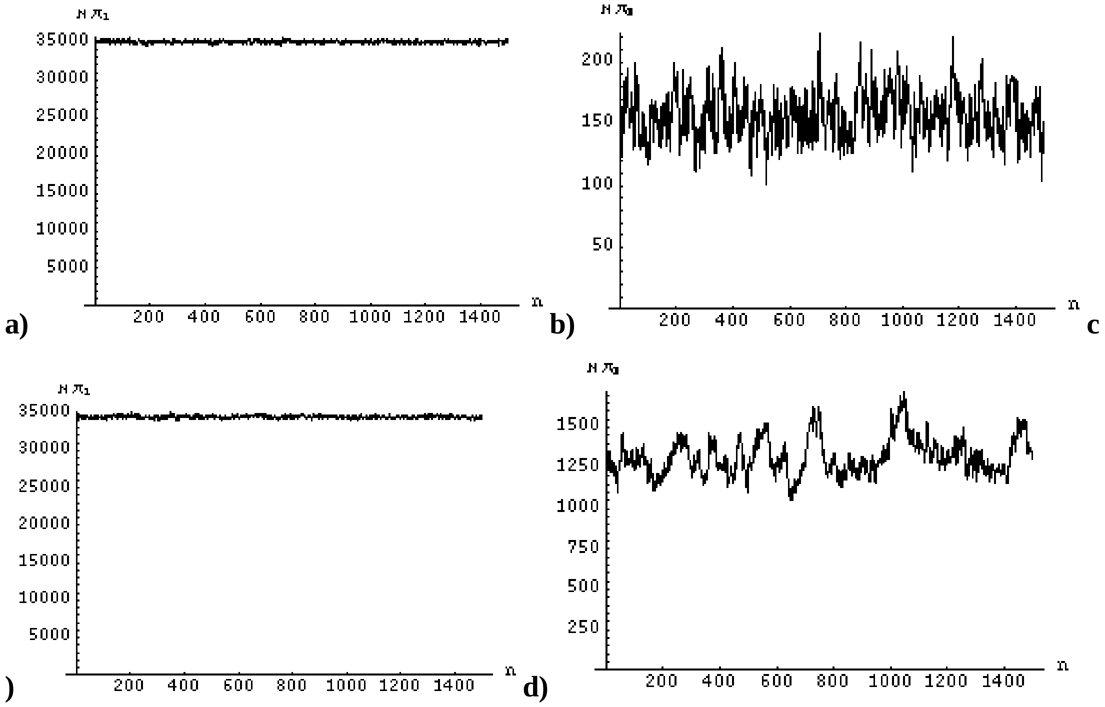


Figure 8.23 The quasi equilibrium evolution of the context-sensitive SDS for two unequal patterns; N=100000, $p_m=p_d=0.001$ and 2000 iterations (of the total 3000 of which first 1000 was discarded as a burn-in period); (a) the evolution of the cluster associated with the best fit pattern for $p^-=0.2$ and $p^+=0.5$ (the expected value is 37500.5, the standard deviation is 153.1, a sample average is 35050.7 and a sample standard deviation is 165.96); (b) the quasi equilibrium cluster associated with the second pattern for $p^-=0.2$ and $p^+=0.5$ (the expected value is 140.15, the standard deviation is 11.83, a sample average is 156.54 and a sample standard deviation is 18.73). The evolution of the clusters associated with the two unequal patterns corresponding to $p^-=0.2$ and $p^+=0.6$ is shown in panels (c) and (d). For a stronger pattern, the panel (c), the expected value is 37106.6, the standard deviation is 152.77, the sample average is 34513.4 and a sample standard deviation is 173.77), and for a weaker pattern, the panel (d), the expected value is 927.41, the standard deviation is 30.26 and the sample average is 1318.88 and the sample standard deviation is 113.63).

This figure again presents the case when one of the patterns is very similar to the template, p^- $=0.2$ (panels (a) and (c)), and the other is either relatively strong, $p^+=0.6$ (panel (b)), or a boundary case, $p^+=0.5$ (panel (d)). Panels (a) and (c), showing the best fit pattern clusters are very similar to the corresponding panels of Figure 8.15, depicting behaviour of the context-free SDS. From the comparison of the statistics predicted from the models of both algorithms and their corresponding sample estimates one can see that the behaviour, with respect to the best fit pattern, of both, the context-free and the context-sensitive SDS, is almost identical. This is reflected in practically the same relative difference between the theoretical and sample average cluster size (6-7%), its relative variation (0.5%), or the ratio of the sample standard deviation to its theoretical value (1.1%). Also one can note that, in this case, both algorithms allocate the same relative amounts of resources to the best fit pattern, about 75%.

The high stability of the clusters depicted in panels (a) and (c) can be contrasted with the higher variability of the clusters corresponding to weaker patterns, as shown in Figure 8.23 (b), (d). The relative difference between the estimated and theoretical cluster size is about 12% in the panel (b) and about 42% in the panel (d). The cluster size in the panel (b) shows definitely higher stability than the one in the panel (d), the ratio of the empirical to the theoretical standard deviations being about 1.6 in the former and about 3.8 in the latter case. Also one can see much stronger serial correlations in the panel (d) implying strong non stationary behaviour. This is probably caused by the fact that the panel (d) represents the case for which the parameters p^-, p^+ describe the point close to the transition between the 'fair' and 'biased' resource distribution, as discussed in relation to Figure 8.17. Indeed, the relative amounts of resources allocated to both patterns by the context-sensitive SDS does not change significantly for the best fit pattern (74-75%) and changes from about 0.3% for the values of parameters $p^-=0.2$, $p^+=0.5$ to about 1.9% for $p^-=0.2$, $p^+=0.6$. The relative insensitivity to the changes of the sub-optimal solution of the relative allocation of resources to the best fit pattern is characteristic for the biased resource distribution, whereas the 6-fold increase of the allocation of agents to the weaker pattern, reflecting increase of its strength, is characteristic of the 'fair' distribution of agents. It is interesting to observe, that the sample average in the panel (d) is significantly higher than the expected cluster size.

Thus the context-sensitive SDS shows in Figure 8.23 similar distribution of agents to the one observed for the previous algorithm, although one can already observe that for sufficiently similar patterns it allocates less disproportionate amounts of agents than the standard, or even the context-free, SDS.

The final simulation of the context-sensitive SDS, shown in Figure 8.24, mirrors the one depicted in Figure 8.16 and shows the quasi equilibrium distribution of agents among two patterns, the stronger one with a fixed parameter $p^-=0.5$, panels (a) and (c), and the second one with an increasing strength, reflected by an increase of the parameter p^+ from 0.42, the panel (b), to 0.48, the panel (d).

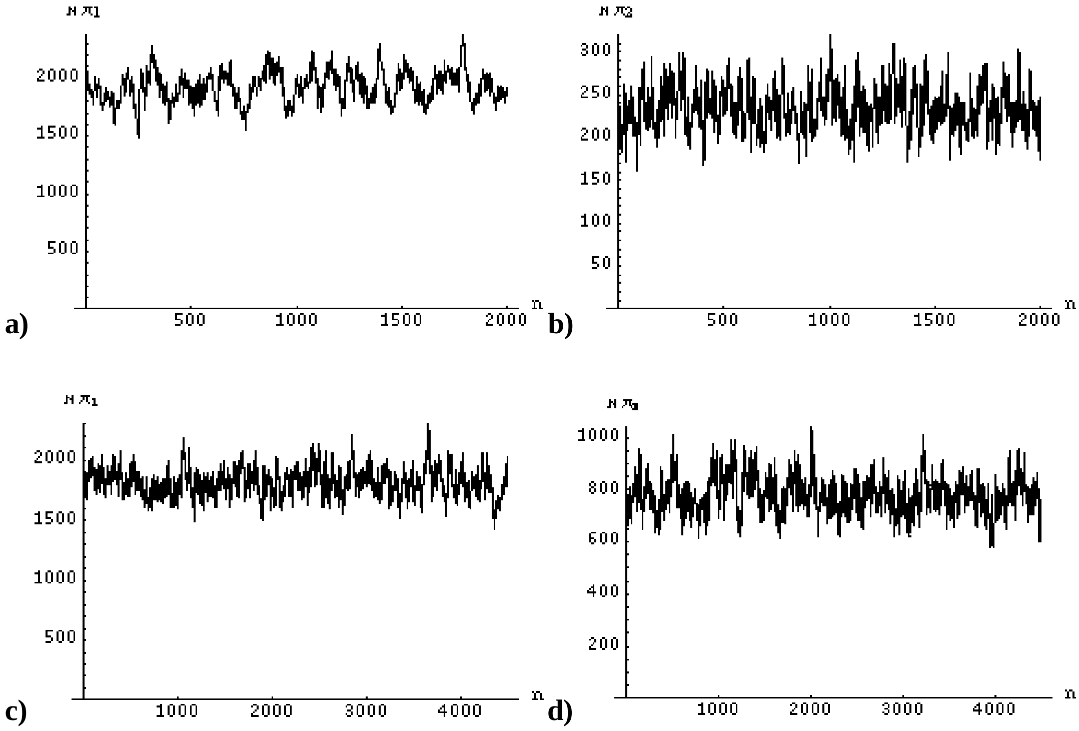


Figure 8.24 The quasi equilibrium evolution of the context-sensitive SDS for two unequal patterns; N=100000, $p_m=p_d=0.001$ and 2000 iterations (of the total 3000 of which first 1000 was discarded as a burn-in period); (a) the evolution of the cluster associated with the best fit pattern for $p^-=0.5$ and $p^+=0.42$ (the expected value is 2125.53, the standard deviation is 45.61, a sample average is 1925.1 and a sample standard deviation is 132.6); (b) the quasi equilibrium cluster associated with the second pattern for $p^-=0.5$ and $p^+=0.42$ (the expected value is 234.41, the standard deviation is 15.29, a sample average is 233.56 and a sample standard deviation is 24.92). The evolution of the clusters associated with the two unequal patterns corresponding to $p^-=0.5$ and $p^+=0.48$ is shown in panels (c) and (d). For a stronger pattern, the panel (c), the expected value is 1984.44, the standard deviation is 44.1, the sample average is 1813.95 and the sample standard deviation is 113.2), and for a weaker pattern, the panel (d), the expected value is 794.31, the standard deviation is 28.1 and the sample average is 778.1 and the sample standard deviation is 67.38).

One can see that basically the context-sensitive SDS behaves in this case in a manner analogous to the context-free SDS. In all four cases the relative differences between the expected average cluster sizes and their empirical estimates are on the same levels, 10-11% for the panels (a) and (c), less than 1% for the panel (b) and less then 4% for the panel (d). Also the relative variations of the cluster sizes in both algorithms match each other for the corresponding sets of parameters. Even the relative distribution of resources is similar and changes analogously together with the change of the parameters, the appropriate numbers are about 4% and 0.5% in panels (a) and (b), decreasing slightly below 4% in the panel (c) and increasing to about 1.5% in the panel (d).

The main difference between the two algorithms can be noticed by comparing the panels (c) and (d) of Figure 8.24 and Figure 8.16. In the case of the context-sensitive SDS both graphs show much less serial correlations than the corresponding graphs depicting the context-free SDS.

Thus, it follows that the comparison of Figure 8.24 and Figure 8.16 confirms the conclusion from the discussion of Figure 8.17 that in the areas of the parameter space corresponding to at least one pattern being weak one can expect basically similar distribution of resources between context-free and context-sensitive SDS.

Thus the context-sensitive SDS shares a lot of similarities with the context-free algorithm. First of all, the most distinctive feature of both of them is a different, to the standard SDS, balance between the exploration and exploitation of the search space. In general both allocate consistently significantly more agents for the exploration of the search space, in many cases amount of agents exploring the search space exceeds 50% of their total number. Both will also support the ideal instantiations of the template with equally strong clusters, although the context-sensitive SDS will allocate more agents to the solutions than the context-free SDS. In cases of a significant difference in strength of the patterns or when at least one of the patterns is weak both of the algorithms will distribute resources in a similar way, showing a bias towards the best fit solution, not unlike the standard SDS. The context-free and context-sensitive algorithms distribute agents more evenly, than the standard Stochastic Diffusion Search, for the patterns which do not differ significantly.

The context-sensitive SDS differs considerably in the resource management for the case of both different but strong patterns. Whereas the context-free and the standard SDS both were showing a strong bias towards the best fit solution, the context-sensitive SDS distributes resources in a more ‘fair’ manner.

8.5. *Characterisation of the dependence of the convegence rate on the false positives for different resource allocation schemes.*

All the properties of both extensions of the modified SDS, discussed so far, were important for the discovery of many sub-optimal solutions or parallel processing of (almost) equal alternatives. The final part of this chapter will discuss how these extensions influence the dynamics of the search. This will have a direct relevance, as was discussed before, to the searches in large, noisy search spaces or alternatively when the search space or the template change dynamically.

An experiment was performed, for each of the algorithms, which investigated the influence of the initial location of a distractor by the algorithms on the average convergence time. The reason for this experiment is that in a large search space with many distractors it is reasonable to assume that initially the search would locate, as the best fit solution, one of the distractors, and would allocate accordingly the amount of resources to form a supporting cluster, leaving the rest of them to explore the search space in the search for alternative solutions. Thus, the search was initialised by forming a cluster of active agents supporting a distractor with the amount of agents that would be normally, on average, allocated by the corresponding algorithm to the unique best fit solution of this strength. In all cases N=1000 agents were used, as well as the ideal instantiation of the template, $p^-=0$, as the best fit solution and a distractor with $p^+=0.8$, whereas p_m and p_d were changing accordingly to the search space size. A measure of the convergence time of algorithms was defined as a number of iterations necessary to form a cluster supporting the ideal instantiation of the template that would be formed normally by the algorithm during a search with unbiased initialisation in search conditions characterised by the appropriate set of parameters.

The results of the experiment with all three algorithms, averaged over 1000 runs for each search space size, are summarised in Table 8.3.

search	standard SDS		context-free SDS		context-sensitive SDS	
space	$\overline{n}$	$\overline{\sigma}$	$\overline{n}$	$\overline{\sigma}$	$\overline{n}$	$\overline{\sigma}$
1000	84.57	18.13	43.68	6.88	25.84	5.2
21000	376.86	314.24	128.36	81.54	63.4	38.99
41000	720.32	662.03	200.84	161.81	96.63	72.37
61000	1024.36	950.69	287.03	238.12	129.13	97.3
81000	1361.09	1241.28	357.16	310.28	170.66	143.47

Table 8.3 The average time, $\overline{n}$, and average standard deviation, $\overline{\sigma}$, of forming a cluster on the ideal instantiation of the template in the search space as a function of the search space size; N=1000, p_m, p_d change accordingly to the search space size. All

three algorithms are initialised with a cluster of active agents, supporting a distractor with p^+=0.8, of the size equal to the expected value of the cluster formed by them for the best fit solution of that strength.

From Table 8.3 one can see that search times of all the three algorithms increased with the increase of the search space size. This was also the case for the standard deviations. In all

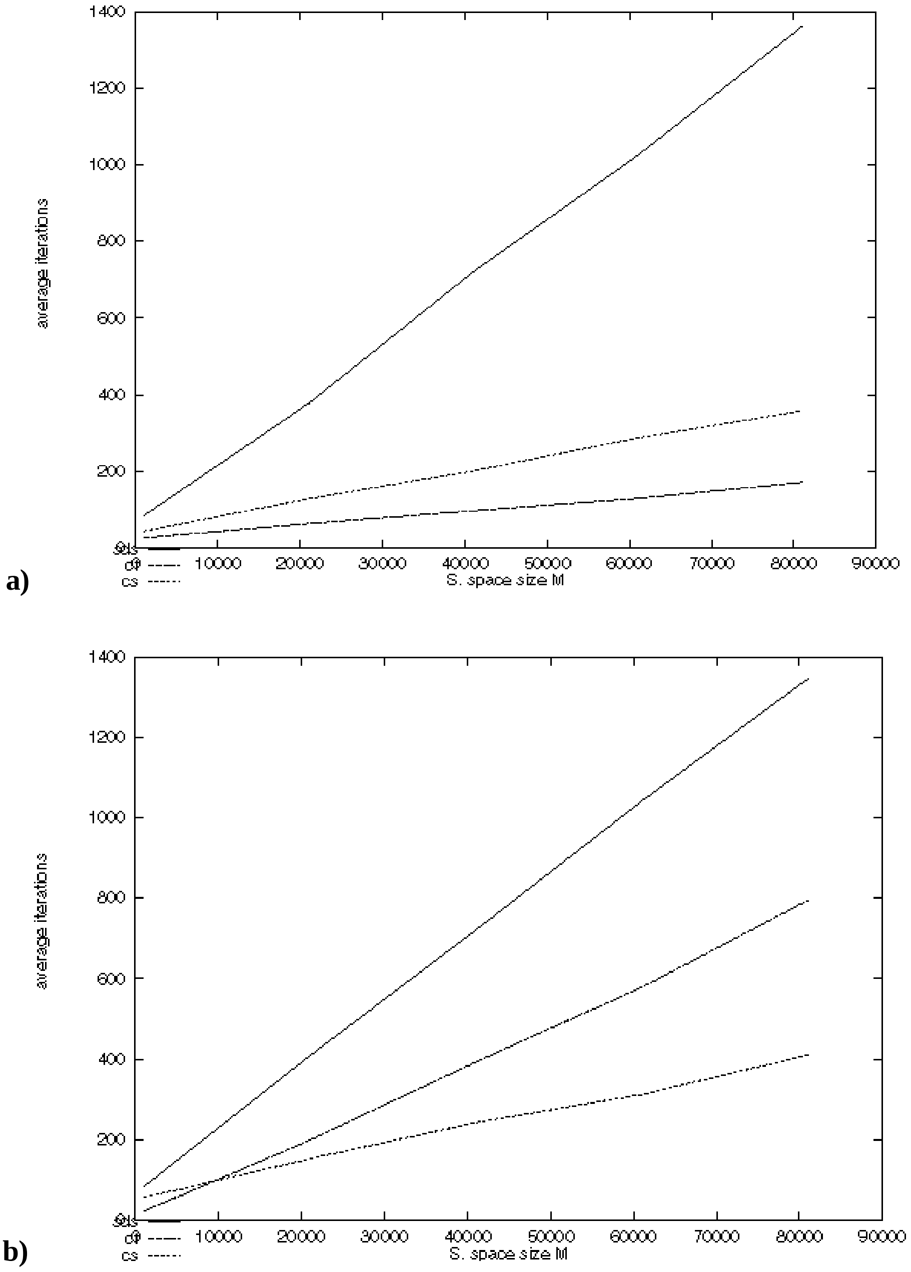


Figure 8.25 The convergence time, averaged over 1000 runs, of the standard, the context-free and the context-sensitive Stochastic Diffusion Search as a function of the search space size. (a) the time complexity of algorithms in a case of initialisation on a distractor $p^+=0.8$ and no other distractors in the search space; (b) the time complexity of algorithms in a case of initialisation on a distractor $p^+=0.8$ and the amount of noise in the search space characterised by the parameters $p^+=0.8$ and $p_d=0.012$.

cases the increase of both quantities seems to be approximately linear. Moreover, the standard deviations are in all cases of the order of the average convergence time. Thus the initial convergence on the sub-optimal solution does not change the characteristic for the Stochastic Diffusion Search time complexity. However, one can note that both forms of the modified SDS are, on average, much faster than the standard SDS and the increase of the convergence time with the search space size is smaller for them. This is clearly visible in Figure 8.25 (a) showing the time complexity of all three algorithms in a single graph.

In this figure the points corresponding to the average convergence on the best fit solution were connected for a clear visibility. One can see that the context-sensitive SDS is fastest of all three algorithms and the context-free SDS is second fast. What is more important, however, is that the increase of the convergence time with the search space size induces the same order on the three algorithms as can be seen from n inspection of the slopes of the time complexity lines. The slope of the context-sensitive SDS is about two times smaller than that of the context-free SDS and almost 9-fold smaller than the slope of the standard SDS. Thus both extensions of the modified SDS are significantly faster than the standard SDS. This is not surprising if one considers this as a result of the different balance, achieved by these algorithms, between exploration and exploitation of the search space. Due to the strong bias towards the best fit solution the standard SDS allocated a vast majority of agents to the initial false positive solution and thus there were relatively few agents left for the exploration of the search space in search for the alternative solutions. On the contrary, both extensions of the modified SDS allocate substantially more resources for the exploration of the search space and thus were able to discover more quickly the real best fit solution. The clear dominance of the context-sensitive SDS over the context-free extension follows from the qualitatively similar resource distribution by the latter to that of the standard SDS, as could be anticipated from comparison of Figure 8.9 and Figure 5.7. The context-sensitive SDS reserves about 50% of the total resources for the exploration of the search space, but introduces a relatively strong bias towards the best fit solution in the distribution of active agents among all supported patterns. The latter feature seems to slow down the flow of agents from the relatively strong sub-optimal solution to the real best fit pattern resulting in slower performance of the context-free SDS in the experiment.

In order to see how the above summarised results are affected by noise a second experiment was performed, in which the conditions from the first one were repeated, except that, this time, a small amount of other distractors was introduced (the corresponding parameters are $p^+=0.8$, $p_d=0.012$). The results are summarised in Table 8.4 and depicted in Figure 8.25 (b).

search	standard SDS		context-free SDS		context-sensitive SDS	
space	$\overline{n}$	$\overline{\sigma}$	$\overline{n}$	$\overline{\sigma}$	$\overline{n}$	$\overline{\sigma}$
1000	83.74	18.36	56.46	13.7	22.29	7.38
21000	408.77	342.06	152.41	95.44	198.63	208.66
41000	721.21	604.99	242.25	174.74	393.56	395.79
61000	1041.54	961.79	312.3	260.51	578.88	572.92
81000	1345.36	1271.76	409.77	361.46	794.97	809.59

Table 8.4 The average time, $\overline{n}$, and average standard deviation, $\overline{\sigma}$, of forming a cluster on the ideal instantiation of the template in the search space as a function of the search space size; N=1000, p_m, p_d change accordingly to the search space size. All three algorithms are initialised with a cluster of active agents, supporting a distractor with p^+=0.8, of the size equal to the expected value of the cluster formed by them for the best fit solution of that strength. The noise is characterised by the parameters p^+=0.8, p_d=0.012.

From Table 8.4 one can see that the small amount of strong noise does not change the time complexity characteristic of the algorithms, which remains linear. Again both context-free and context-sensitive SDS are, on average, faster than the standard SDS and their rate of change of convergence time is also significantly lower. From Figure 8.25 (b) one can see that, in this case, for the context-sensitive SDS the increase of the convergence time with the increase of the search space size is faster than for the context-free SDS. From the comparison of panels (a) and (b) of this figure one can make interesting observations. It follows, that both the standard and the context-free SDS practically did not change the slopes of the time complexity lines whereas, the context-sensitive SDS increased the slope more than 5 times.

This is again a result of the particular resource allocation management by these algorithms. The very bias towards the best fit solution, common to the context-free and the standard SDS, that made them, in the first experiment, slower than the context-sensitive SDS, makes these algorithms much more robust to the noise, which is reflected in the constancy of the slopes of the time complexity lines. On the other hand, the context-sensitive SDS distributes the resources evenly among the patterns of comparable strength, therefore the presence of strong distractors decreases the total number of agents allocated for the exploration of the search space and thus results in a lower robustness of this algorithm to the strong noise.

Thus these experiments show that, under a constraint of limited resources, there is a trade off between the ability of parallel processing of several objects (‘fair’ distribution of resources to comparable solutions), and other features of the processing, in this case the processing time.

8.6. *Summary.*

1. It is possible to relate the operation of the extensions of SDS to the mode of information processing referred to as a winner-takes-all processing in which signals from many nodes are compared and the strongest one is multiplied by one and all other by 0. The context-free SDS, as well as the standard Stochastic Diffusion Search, seem to process the information from the search space in an analogous way - in most situations they select from the search space a single best solution by forming a single cluster which is usually bigger (by many orders of magnitude) than clusters corresponding to other sub-solutions. This huge difference, in practice, would mean that the information about other sub-solutions would be suppressed in potential further stages of processing because those would have to be insensitive to small clusters due to the inherent fluctuations of the maximal signal.

2. The operation of the context-sensitive SDS clearly allows for combination of information from many alternative sub-solutions due to a more 'fair' distribution of resources to strong patterns. One would be tempted to compare this mode of operation to the soft weighting, as opposed to hard 0/1 weighting of the winner-takes-all mechanism. However, there is a significant difference between soft weighting and the operation of the context-sensitive SDS. The former would need to have a predefined number of possible alternatives reflected in the predefined output nodes being a subject of soft weighting, whereas in the case of the context-sensitive SDS the alternative solutions need not to be predefined and are dynamically discovered and supported. This allows for a much higher flexibility than the soft weighting. One of the characteristic features of the context on the information processing seems to be its dynamic impact - the same information can be processed in very different ways if presented in different contexts. Moreover, often a small change of the contextual information can lead to very big changes in processing and it is often impossible to state beforehand what influence a given piece of contextual information would have on the overall processing. The winner-takes all mechanisms do not fit very well in the framework of this type of processing and the soft weighting allows also for a limited and predefined contextual processing. However, the context-sensitive SDS allows for use of the contextual information (coming from alternative solutions) in further processing stages by dynamically finding in parallel all potentially relevant information.

3. One could be tempted to make some claims about the optimality of one of the algorithms or its overall better performance than the other two. However, it seems that such claims are of value limited only to the simplest cases, in which the optimality is understood in terms of a

single or scalar criterion. It seems clear that it is possible to formulate such criteria according to which in turn all three algorithms would dominate over the others. This is because the optimality is always relative to the criterion applied and to the particular ‘cost’ functional used to formalise this criterion. However, it is much more difficult to talk about optimality in cases when several, often contradictory criteria apply. In such cases, it may be not possible to obtain an algorithm which would be optimal with respect to all criteria taken alone - some trade off’s has to apply. However, the trade off’s do not necessarily lead to satisfactory solutions because they can result in mechanisms performing poorly under all single criteria. Thus, from the foregoing discussion it follows that the optimality of any of the algorithms discussed here depends on the particular ‘context’ of its use.

8.7. *Conclusions.*

This Chapter proposed a modification of the Stochastic Diffusion Search, which achieved a different balance between the exploration and exploitation of the search space. In particular it allocated consistently more resources for the exploration of the search space than the standard SDS. A model of the modified SDS was defined and several properties of the algorithm were derived and discussed.

The modification introduced in this chapter lead to two novel extensions of SDS, arising in the context of more potential solutions in the search space. They were termed the context-free and the context-sensitive SDS, accordingly to the way they utilised available information in order to form clusters supporting potential solutions.

The context-free SDS appeared to be in many respects similar to the standard SDS, showing analogous resource distribution between unequal patterns, however it inherited from the initial modification of SDS the more balanced management of agents with respect to exploration and exploitation of the search space. It differed from the standard algorithm by creating more comparable clusters corresponding to sufficiently close patterns. Also it seems to cope better with the search in the dynamic scenarios or in the large, noise search spaces.

The context-sensitive SDS differed substantially from both context-free and the standard SDS in its distribution of resources between the different but comparably strong patterns. It showed much less biased allocation of agents in these cases, than any of the former algorithms, thus it allowed for parallel processing of several potentially interesting alternative solutions whose number did not have to be prespecified but was dynamically determined by the algorithm. It seemed also more suitable for modelling an influence of context sensitivity also on a macrolevel of the system response, in contrast to the context-free and the standard SDS which were more comparable to the winner-takes-all mechanism. It seems to be also very promising mechanism for use in situations in which the search space or the template change dynamically but shows in such cases much smaller robustness to noise than the context-free or even the standard SDS.

Finally it is necessary to observe that the algorithms introduced in this chapter are very simple extensions of the standard Stochastic Diffusion Search and by no means exhaust the range of possible modifications and hence different characteristics which can be useful in very different search scenarios.

CHAPTER 9

CONCLUSIONS

9.1. *Summary of the work.*

The previous chapter completed the study of the resource allocation management of the Stochastic Diffusion Search by presenting two alternative extensions of the algorithm. The use of the Ehrenfest Urn model as a main tool of investigation of the Stochastic Diffusion Search appeared to be very fruitful. It resulted not only in a formal characterisation of an important aspect of SDS operation, but also in the aquisition of a new perspective on the algorithm. This alternative interpretation of the SDS operation (in terms of a balance of exploration and exploitation of the search space) has opened a way for possible extensions of the algorithm, of which two were discussed in detail.

The thesis presented a detailed analysis of the Stochastic Diffusion Search, a generic search technique. As follows from the Introduction and a detailed review of different paradigms and their representative mechanisms, presented in Chapter 2, one can conclude that the driving force which creates whole new research areas in Artificial Intelligence, is, in fact, a quest for appropriate and powerful internal representations. The methods of solving problems are constrained by the type of representations, on which they operate and often are by them partially determined. It seems also that there is a broad convergence of efforts coming from seemingly different paradigms, as to what characteristics such representations should hold.

It should be clear from the discussion of the Stochastic Diffusion Search, presented in this thesis, that it posseses many such features. It operates on sub-symbolic entities, or micro-features, which by themselves do not have a semantic interpretation. It is only by creation of stable clusters of agents that meaningful representations of external events or objects are formed. These clusters acquire a meaningful interpretation because they correspond to semantic entities and because they are grounded in the external environment, in which the system operates. Indeed, it should be clear that the representations in SDS are not self-sustaining, as they are perpetually re-constructed by a constant flow of information from the search space. This is related to the fact that the representations, although stable, are nevertheless not static but dynamic due to continuing restructuring caused by agent's interaction. The clusters, or representations, are not defined by identities of constituent agents, because the latter constantly change their state. The identity of these representations follows as an emergent collective property of agents, which happen, at a given instant, to process information relevant for defining the semantic content of a cluster, which they constitute.

By the very nature of the method, the search for a solution is coupled with creation of representations. In other words, the search in fact consists of parallel exploration of multiple alternatives, arising in the attempt of the system to create their corresponding representations. It is by competition of these emerging representations, that the sub-optimal solutions are being greatly suppressed and the system settles on the solution that fits to the highest degree to a pre-defined specification.

It follows, that the Stochastic Diffusion Search performs a global search based on a biased random walk performed by the agents. As a result of supporting dynamic representations by SDS follows its great flexibility and capability of reacting to changes in the search space, e.g. the change of the position of the template.

Another view of SDS discussed in the thesis is in the context of its multi-agent architecture. Indeed, the operation of the system can be described in terms of seeking a solution to a problem by co-operation and competition between autonomous agents performing the search in parallel. The generic formulation of SDS may potentially allow to characterise many distributed, multi-agent systems as performing some form of a Stochastic Diffusion.

The thesis constitutes the first detailed and systematic attempt at a theoretical characterisation of this system. It reports results pertaining to stability of the search and its time complexity. It has been proven that SDS converges to a steady state probability distribution which has a unique sharp peak. This implies that the system stays most of the time within a small fraction of its state space, thus exhibiting in practice a quasi-equilibrium behaviour. The prefix ‘quasi’ should not be automatically interpreted in a pejorative sense. As follows from the generalised Ehrenfest Urn model, SDS can be directly compared to a quasi-equilibrium of gas particles contained in a system of two identical, connected containers, in which the distribution of particles between the containers is approximately equal, subject to small fluctuations. The quasi-equilibrium state of SDS corresponds to the global best fit solution to a pre-specified template.

The dynamic character of the system, even when settled on the solution, is responsible for its adaptability and for its ability to revise the solution, subject to changes either in the template specification or in the environment. From the analysis it also follows, that the Stochastic Diffusion Search usually finds the solution very fast and its time complexity is sub-linear in the search space size.

Both quasi-stability and time complexity as well as reliability of a solution found by the system are related to the amount of resources used in search. Therefore the thesis concentrated on investigation of the resource allocation management of SDS.

Chapters 4 through 6 contain a detailed analysis of resource allocation to the best fit pattern in different search conditions. In Chapter 4 the generalised Ehrenfest Urn model is introduced and applied in order to obtain a detailed steady state allocation of agents to the unique best fit match pattern in the search space. It appears that the allocation of resources by the Stochastic Diffusion Search is nonlinearly dependent on the match between the best fit solution and the template. Although the amount of allocated resources changes smoothly with the parameter characterising the fit, nevertheless the character of this change allows to distinguish two different regimes of SDS operation. For the best fit pattern with an overlap of at least 50% with the template the amounts of, allocated to it, agents constitute in an equilibrium a significant part of the overall resources, whereas below this threshold the cluster associated with the representation of the desired solution constitutes only a small fraction of the resources. Chapter 4 characterises also the span of fluctuations of the best fit cluster in the steady state.

Chapter 5 discusses the model under a relaxed condition of a unique best fit solution. It presents a detailed analysis of the steady state distribution of resources between two patterns similar to the template. The two possible interpretations of results reported here constitute a particularly interesting aspect of this work. In the first interpretation one can consider one of the patterns, with the higher overlap with the template, as a desired solution and the second one as a distractor. Thus in this context the analysis presented in Chapter 5, and continued in Chapter 6, describes the behaviour of SDS in a presence of noise. It appears that SDS performs a highly nonlinear discrimination between the patterns, which is biased towards the best fit solution. It follows from the above feature that the algorithm is highly sensitive to the signal to noise ratio and retains its discriminative characteristics even in a noisy search space.

The second interpretation of results presented in Chapter 5 is in the context of searching for multiple sub-optimal solutions. In this case, we see that the very strong bias towards the best fit solution is responsible for an emergence of serial processing out of this massively distributed, parallel system. The serial processing appears as a result of the sequential, in practice, operation of the largest cluster of active agents. However, the fact that this serial behaviour is an emergent, and not built-in, or pre-specified, property of SDS, follows from the underlying parallel operation of all of the agents. Also, in the case of very similar patterns, we can see that the allocation of resources is more even and for patterns of equal overlap with the template, the difference between corresponding cluster sizes, statistically, disappears. Thus there is a potential in SDS operation to switch smoothly between sequential and parallel operation, an interesting property from the point of view of modelling attention postulated by Horwitz in (Selman, 1996).

Chapter 7 presents an approximate solution to the steady state distribution of agents, which enables one to obtain a simpler analytic formulas expressing the dependence of the resource allocation on the parameters characterising search conditions. It follows that the approximate solutions are able to match closely the model’s predictions of SDS’s behaviour and, in the regions of the parameter space, where the numerical accuracy brakes down, they are still showing an appropriate rate of change in the response to the changes in parameters. The comparison between the exact model and the approximation is further discussed and illustrated.

As a natural consequence of the analysis performed in the proceeding chapters, Chapter 8 introduces two extensions of the basic resource allocation scheme used in the Stochastic Diffusion Search. They differ from each other with respect to the way in which they generalise an agent’s communication of the standard SDS, on which the resource allocation is dependent. These changes of interaction of agents have very interesting implications for the behaviour of these systems.

Their common feature, distinguishing them from the standard SDS, is a much smaller bias towards the exploitation as opposed to the exploration of the search space. Whereas the original algorithm allocates the majority of resources to the target solution, leaving out comparatively few agents for the exploration of alternative solutions, both modifications will allocate substantial amounts of agents for the latter mode of searching for the solution, while retaining capability of maintaining the stable representations of the best fit solution found so far.

Both modifications are indistinguishable in the case of a single object with an overlap with the template and differ from the standard SDS in the amount of resources they allocate to the best fit solution in this case. The algorithms start to differ from each other in the case of more objects similar to the template existing in the search space.

In the first extension, the so called ‘context free’ SDS, the equilibrium distribution of agents exhibits qualitatively similar dependence on parameters characterising the search condition to the standard SDS. Thus, while not compromising the exploration of the search space for new solutions, it retains the very high sensitivity of the standard Stochastic Diffusion Search to the difference between potential solutions. From this follows a strong bias towards the best fit solution and towards the emergence of sequential processing.

In contrast, the ‘context sensitive’ SDS, differs from its predecessor in addition in the smaller bias in allocation of resources to the best fit pattern and thus supports a more parallel mode of operation on the level of emerging representations. This feature could potentially enable taking into account contextual information, which would normally be highly suppressed in both standard and ‘context free’ SDS. The price for this interesting feature is, however, a lower signal from noise discriminability of this algorithm.

The resource allocation has an impact on the speed with which the best fit solution is being found and thus the three algorithms seem to differ with respect to this characteristic. The convergence time of the standard SDS seems to be the least affected by disturbances or noise but also was the longest for the reported simulations. Out of the two modifications, the ‘context free’ SDS maintained an equilibrium between the mean convergence time’s robustness to noise and the overall speed, whereas the ‘context sensitive’ SDS showed the highest, of all three, sensitivity to noise in the mean convergence time although, in the reported simulation, it remained approximately twice as fast as the standard SDS (the speed-up with respect to the classic SDS in a noiseless search space was approximately 5-fold).

9.2. *Open questions and future research.*

There are many possible directions which could be further explored on the basis of the research reported in the thesis.

A very important topic that needs further elaboration is a more detailed characterisation of temporal behaviour of the Stochastic Diffusion Search. The property of sub-linear time complexity reported here has been proven in a special case of a noiseless search space. Although the simulations suggest that SDS is robust to the noise with respect to this property, it is necessary to establish this result formally. Many simulations reported in the thesis suggested that the convergence time to the steady state by SDS may be quite sensitive to the particular set-up of parameters. It seems that this issue also deserves further formal elaboration.

A related issue is an analysis of the convergence time in the presence of sub-optimal solutions very similar to the best fit pattern. The argument provided in Chapter 8 and supported by a simulation, maintains that one should expect in such a scenario a considerable slow-down of the mean convergence time due to a premature convergence of SDS to the false positive solution. The exact dependence of the mean search time on the amount and strength of disturbances is worth investigation.

The undesirable, from the point of view of potential applications, property of the convergence time of SDS is that its variation, as characterised by the standard deviation, grows proportionally to the mean time. It would be worthwhile to investigate the potential modifications to the algorithm, which could lead to the decrease of this dispersion of the convergence time.

This thesis focused on the behaviour of SDS in the steady state. However, the transient behaviour may be an interesting issue worth separate studies. A somewhat related topic is the behaviour of SDS in the case when the number of agents is not sufficient to maintain stable representations.

All of the above questions could be investigated also in case of both modifications of the standard SDS proposed in Chapter 8.

Many of the multi agent architectures incorporate learning of the agents, as discussed in Chapter 2. At present SDS exhibits solely an activity dynamics as opposed to the weight dynamics, which is incorporated in the overwhelming majority of connectionist systems. Therefore investigating potential extension of SDS in this direction seems an interesting topic on its own. An extremely interesting and challenging topic would be to analyse the potential interactions between the two types of dynamics in SDS.

The thesis stressed the resource allocation management capability of the Stochastic Diffusion Search. The potential extension of the research in this spirit seems to be particularly interesting in connection with incorporating learning into the Stochastic Diffusion Search. In this context it would be very interesting to investigate how learning of agents would affect allocation of resources by the system. Such an analysis could potentially suggest different interaction patterns between the activation and learning dynamics, the latter being responsible for the specialisation of agents and the former for their allocation to the task.

Finally it is worth mentioning that the model of the Stochastic Diffusion Search, the generalised Ehrenfest urn model, seems to be sufficiently general and flexible to be a promising framework for an analysis of the resource allocation as a standalone issue. Thus it has a potential for generalisation as a framework of analysis of multi agent systems. In particular, incorporation of learning in SDS and, consequently, extending the Ehrenfest urn model, could provide a general conceptual framework to reason about many existing multi agent systems.

9.3. *Ongoing research related to the thesis.*

In the characterisation of the Stochastic Diffusion Search given in the thesis, the stress was laid on the features that it shared with many current approaches towards the creation of intelligent systems. The very question of possibility of creation of a truly intelligent system is a controversial and hotly debated subject, some works presenting a negative view were mentioned in the Introduction and in Chapter 2. Most of the philosophical questions revolving around this issue remain open problems, partly due to the elusive content of the concepts involved, which most often escape the attempts of their precise definition. A related problem is our general impoverished knowledge about the nature of intelligence and the conditions necessary for it to arise. It would be probably much easier to get a grasp on the subject, if we could observe in nature many totally different varieties of systems and physical media, which would support intelligent behaviour. We could then try to delineate some common characteristic of them, which would in turn enable us to see what are the necessary and sufficient properties for a system to be intelligent. Unfortunately, we are not in such a position. In fact, there exist only one system we know of, and only a very specific medium for its instantiation, that undoubtfully can support intelligence - the nervous system. Thus it is difficult to define features which are truly essential for emergence of intelligence.

It seems a natural question therefore to investigate if the information processing in the Stochastic Diffusion Search is related to the information processing in the brain, as the only known site of intelligence.

This line of research, related to the subject of this thesis, resulted in the paper (Nasuto et al, 1999) discussing a network of spiking neurons which was designed to imitate the operation of the Stochastic Diffusion Search.

In fact the hypothesis put forward in this article maintains that we should revise our understanding of how the information may be handled in the brain. The traditional understanding of the type of information processing performed by neural structures is intimately biased by the assumptions made in neuroscience about the nature of the information encoding. The mainstream neuroscience literature bases interpretation of observed facts and experiments on the rate coding of information by integrative neurons. The closely associated 'labelled lines' assumption holds that the nerve fibres have some distinctive property, that enables neurons up the processing stream to interpret correctly what features are being encoded in messages.

The rate coding, the labelled lines hypothesis and a hebbian type of learning constitute the very core of both the interpretation of experiments by neurobiologists and modelling of information processing in the brain by computational neuroscientists.

However, the labelled lines hypothesis, although so common that almost never explicitly stated, has been recently criticised by Rose, (Rose, 1998; Rose, 1998), as not really supported by a solid evidence. Cariani also points to the problems connected with the labelled lines hypothesis and discusses growing evidence for, alternative to rate, temporal encoding in both peripheries and in the cortex, (Cariani, 1997). Also, the mechanism supposed to be a biological counterpart of a standard hebbian learning and memory, namely the Long Term Potentiation (LTP), is not unproblematic and is criticised in (Anderson, 1997; Russo, 1999). Shors and Matzel, (Shors, Matzel, 1997) suggest that the role of LTP in learning is unsupported and propose an alternative interpretation in terms of a general arousal mechanism. The alternative, to standard hebbian learning, mechanisms include processing via diffusive messengers, e.g. the work on Nitric Oxide, (Krekelberg, Taylor, 1997), volume learning, (Mitchell, 1999; Nicholson, 1999), resource consumption principle, (Montague, 1996), neural darwinism, (Edelman, 1989), or chemotaxis, (Bremmermann, Anderson, 1991;

Anderson, 1997). In parallel, the intensive work on alternative memory mechanisms is carried out, see e.g. (Alkon et al., 1998; Geary, 1997).

The model discussed in (Nasuto et al., 1999), takes into account the recent neurobiological evidence suggesting an alternative interpretation of information processing by single neurons. It proposes to revise the classic McCulloch-Pitts model neuron and puts forward a model which stresses much more an active role played by communication and routing of messages in the information processing.

It is therefore related to another challenge for the Artificial Intelligence community posed by Rodney Brooks in a panel discussion reported in (Selman, 1996), where he proposed to revise the McCulloch-Pitts model of neurons in a search for alternative connectionist systems. The proposal put forward in (Nasuto et al., 1999) states that one has to take into account the information processing performed in dendritic and axon trees, which are not passive cable structures, as assumed previously. On this view and under the assumption of temporal encoding of information it is suggested that neurons effectively may act as spatio-temporal filters capable of selecting both the particular messages from the incoming information for further processing and the target neurons which will receive the results of processing.

The paper proposed a simple realisation of this model neuron in a network that processes information in a way similar to the classical Stochastic Diffusion Search, the exact relationship between the two systems is a subject of ongoing research.

It seems that the work started in (Nasuto et al., 1999) opens a whole host of interesting questions worth further investigation and related to the Stochastic Diffusion Search and research reported in this thesis. One of the broad areas that this research suggests is an investigation of alternative, asynchronous formulations of SDS. An asynchronous mode of operation may be more appropriate for a description of operation of the network outlined in this article and seems more plausible from a biological perspective. The extension of the Ehrenfest urn model developed here should allow for a detailed investigation of this, and many other questions related to grounding the network with more biologically inspired constraints.

It is hoped that the research in this direction, by exploring alternative and biologically inspired modes of information processing, may help us in the ultimate understanding of the emergence of human intelligence.

APPENDIX 1

NOTATION

f_n^s - the number of active agents pointing to the same position in the search space,

$z_n = max_{s \in S}(f_n^s)$,

$\forall$, $\exists$ -logical quantifiers,

$\wedge$ - logical conjunction,

p_m the probability that, in the uniformly random draw from the search space one would select the best fit solution,

p_d the probability that, in the uniformly random draw from the search space a disturbance would be selected,

p_w the probability that, in the uniformly random draw from the search space we would select a position that can be firmly rejected,

p^+ the probability of accepting a disturbance as a possible solution during testing,

p^- the probability of rejecting the data model during testing,

$\lfloor x \rfloor$ the floor of x, i.e. the largest integer smaller than x

APPENDIX 2

MATHEMATICAL DEFINITIONS

This appendix collects definitions and theorems used in proofs, which may not be found in standard textbooks describing Markov Chain theory.

In his treatment of nonhomogenous Markov Chains Seneta introduces the following notation

$$P_k = \{p_{ij}(k)\}$$

for the stochastic matrix at time step k. He also defines the product of stochastic matrices

$$T_{0,k} = P_1 P_2 \dots P_k = \prod_{r=1}^{k} P_r$$

and analyses the asymptotic behaviour of $T_{p,k}$ as K tends to infinity, (Seneta, 1973).

Definition 1 (Seneta, 1973).

We shall say that weak *ergodicity* obtains for the nonhomogenous Markov Chain (i.e. sequence of stochastic matrices P_i) if

$$t_{i,s}^{(p,k)} - t_{j,s}^{(p,k)} \to 0$$

as $k \to \infty$ for each i,j,s,p.(Note that it is sufficient to consider $i \neq j$).

Definition 2 (Seneta, 1973).

We shall say that the sequence {P$_i$} (or a corresponding Markov Chain) is *asymptotically stationary* if there is a probability vector ***D*** such that

$$\lim_{k \to \infty} D' T_{p,k} = D', p \geq 0.$$

Definition 3 (Seneta, 1973).

If weak ergodicity obtains, and the $t_{i,s}^{(p,k)}$ themselves tend to a limit for all i, s, p as $k \to \infty$, then we say *strong ergodicity* obtains. Hence strong ergodicity requires the elementwise existence of $T_{p,k}$ as $k \to \infty$ for each p, in addition to weak ergodicity.

Theorem 1 (Seneta, 1973, THEOREM 4.8)

For a sequence $S=\{P_i\}$ of *(n x n)* stochastic matrices, there is tendency to stability of the inhomogeneous $T_{p,k}$ as $k \to \infty$, p=0,1,2,...(i.e. weak ergodicity obtains) if

$$\sum_{i=1}^{\infty} \lambda(P_i)$$

diverges, where

$$\lambda(P) \equiv \max_j \left\{ \min_i p_{ij} \right\}$$

Theorem 2 (Seneta, 1973, THEOREM 4.11)

Suppose weak ergodicity obtains for the sequence $\{P_i\}$. Then a necessary and sufficient condition for strong ergodicity is asymptotic stationarity of $\{P_i\}$.

Theorem 3 Birkhoff-Kellog-Schauder fixed point theorem, (Saaty, 1981, theorem 1-21)

If:

1. X is a Banach space,
2. E is a convex compact subset of X,
3. T is a continuous mapping of E into E,

then T has a fixed point.

APPENDIX 3

COMPARISON OF APPROXIMATE AND EXACT MODELS' PREDICTIONS OF THE RESOURCE ALLOCATION

This appendix contains additional tables comparing the approximate steady state resource distribution (derived in Chapter 7) with extended Ehrenfest Urn model predictions and with statistics from simulations for corresponding values of parameters.

p^+	$N\pi_1$	σ_1	$N\pi_2$	σ_2
0.2	75006.7	136.9184	1.66552	1.29053
0.4	75003.3	136.9246	4.99634	2.2352
0.5	75000	136.9306	8.32686	2.8855
0.6	74993.3	136.9429	14.987	3.871

Table 1 Theoretical prediction of the average cluster sizes and their variation for p^-=0.2 and varying p^+. π_1 and π_2 are equilibrium probabilities of agent's activation on the best fit and the second pattern respectively and σ_1 and σ_2 are corresponding standard deviations; N=100000, p_m=p_d=0.001.

p^+	$N\pi'_1$	σ'_1	$N\pi'_2$	σ'_2
0.2	75008.3	136.93	2.07825	1.439
0.4	75008.3	136.93	12.3806	3.518
0.5	75008.3	136.93	191.265	13.817
0.6	75008.3	136.93	2088.12	45.216

Table 2 Approximate prediction of the average cluster sizes and their variation for p^-=0.2 and varying p^+. π'_1 and π'_2 are equilibrium probabilities of agent's activation on the best fit and the second pattern respectively and σ'_1 and σ'_2 are corresponding standard deviations; N=100000, p_m=p_d=0.001.

p^+	n_1	σ_1	n_2	σ_2
0.2	75006.2	142.292	1.63824	1.33791
0.4	75004.4	141.377	4.98934	2.3248
0.5	74992.5	144.474	8.04997	3.12535
0.6	74994.1	146.497	14.9367	4.36737

Table 3 Statistics from SDS simulation for p^-=0.2 and varying p^+. n_1 and n_2 are equilibrium probabilities of agent's activation on the best fit and the second pattern respectively and σ_1 and σ_2 are corresponding standard deviations; N=100000, p_m=p_d=0.001.

p^+	$N\pi_1$	σ_1	$N\pi_2$	σ_2
0.54	33229.2	148.954	237.334	15.3874
0.56	33099.4	148.808	367.706	19.1404
0.58	32716.1	148.367	752.483	27.328
0.59	31978.1	147.486	1493.63	38.3577

Table 4 Theoretical prediction of the average cluster sizes and their variation for p^-=0.4 and varying p^+. π_1 and π_2 are equilibrium probabilities of agent's activation on the best fit and the second pattern respectively and σ_1 and σ_2 are corresponding standard deviations; N=100000, $p_m=p_d=0.001$.

p^+	$N\pi'_1$	σ'_1	$N\pi'_2$	σ'_2
0.54	33465.6	149.218	6759.35	79.3881
0.56	33465.6	149.218	9601.42	93.1641
0.58	33465.6	149.218	12282.8	103.798
0.59	33465.6	149.218	13561.4	108.269

Table 5 Approximate prediction of the average cluster sizes and their variation for p^-=0.4 and varying p^+. π'_1 and π'_2 are equilibrium probabilities of agent's activation on the best fit and the second pattern respectively and σ'_1 and σ'_2 are corresponding standard deviations; N=100000, $p_m=p_d=0.001$.

p^+	n_1	σ_1	n_2	σ_2
0.54	33192.0	233.089	238.543	28.8443
0.56	33075.8	222.486	365.881	43.0159
0.58	32709.2	244.555	771.833	90.5579
0.59	31919.5	277.386	1555.49	141.414

Table 6 Statistics from SDS simulation for p^-=0.4 and varying p^+. n_1 and n_2 are equilibrium probabilities of agent's activation on the best fit and the second pattern respectively and σ_1 and σ_2 are corresponding standard deviations; N=100000, p_m=p_d=0.001.

REFERENCES

Abeles, M., Prut, Y., Bergman, Vaadia, E., “Synchronisation in neuron transmission and its importance for information processing”, Progress in Brain Research, 102, pp. 395-404, 1994.

Ackley, D.H., Hinton, G.E., Sejnowski, T.J., “A learning Algorithm for Boltzmann Machines”, Cognitive Science, 9, pp. 147-169, 1985.

Aleksander, I., Stonham, T.J., “Guide to Pattern Recognition using Random Access Memories”, Computers & Digital Techniques, 2(1), pp. 29-40, 1979.

Alkon, D.L., Neslon, T.J., Zhao, W., Cavallaro, S., “Time domains of neuronal Ca^{2+} signalling and associative memory: steps through a calexcitin, ryanodine receptor, K^{+} channel cascade”, Trends in Neurosciences, 21(12), pp.529-537, 1998.

Alt, W., “Biased Random Walk Models for Chemotaxis and Related Diffusion Approximations”, Journal of Mathematical Biology, 9, pp.147-177, 1980.

Anderson, R.W., “Random walk learning: A neurobiological correlate to trial and error”. In J. Dayhoff, (ed.), Progress in Neural Networks, Ablex, Norwood, NJ, 1997.

Arbib, M.A., “Brains, Machines and Mathematics”, Springer Verlag, London, 1987.

Baars, B.J., “A Global Workspace Theory of Conscious Experience”. In Consciousness in Philosophy and Cognitive Neuroscience, Revonsuo, A., Kamppinen, M., (eds.), Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1994.

Back, T., “Evolutionary Algorithms in Theory and Practice”, Oxford University Press, 1996.

Bailey, T.J., The elements of Stochastic Processes, New York: John Wiley & Sons, Inc, 1964.

Bak, P, Tang, C., Wiesenfeld, K., "Self-Organized Criticality: An Explanation of *1/f* Noise", Phys. Rev. Lett., 59, pp 381- ??, 1987.

Ballard, D.H., et al, "Parallel visual computation", Nature, 306(3), pp. 21-26, 1983.

Beattie, P.D., Bishop, J.M., "Self-Localisation in the 'Scenario' Autonomous Wheelchair", Journal of Intelligent and Robotic Systems, 22, pp. 255-267, 1998.

Bird, R.S., "Two dimensional Pattern Matching", Information Processing Letters, 6(5), pp. 168-170, 1977.

Bishop, J.M., Anarchic Techniques for Pattern Classification, PhD Thesis, University of Reading, 1989.

Bishop, J.M., Torr, P., The Stochastic Search Network. In R. Linggard, D.J. Myers, C. Nightingale (eds.), Neural Networks for Images, Speech and Natural Language. New York: Chapman & Hall, 1992.

Boettcher, S., Perkus, A., "Nature's Way of Optimizing", xxx-cm/9901351, 1999.

Boyer, R.S., Moore, J.S., "A fast string searching algorithm", Communications of Association of Computing Machinery, 20, pp. 762-772, 1997.

Bremermann, H.J., Anderson, R.W., "How the Brain Adjust Synapses - Maybe". In Automated Reasoning. Essays in Honor of Woody Bledsoe, R.S. Boyer (ed.), Kluwer, Dordrecht, 1991.

Brooks, R.A., "Intelligence without Representation", Artificial Intelligence Journal, 47, pp.139-160, 1991.

Buchholz, P., "Exact and ordinary lumpability in finite Markov chains", Journal of Applied Probability, 31, pp. 59-75, 1994.

Cao,W.-L., Stewart, J., "Iterative Aggregation/Disaggregation Techniques for Nearly

Uncoupled Markov Chains", Journal of the Association for Computing Machinery, 32(3), pp. 702-719, 1985.

Cariani, P.A., "Temporal Coding of Sensory Information", Computational Neuroscience, 92, pp. 591-598.

Daugman, J. G., "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters", Journal of Optical Society of America, A-2(7), pp. 1160-1169, 1985.

Davis, L., Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York, 1990.

Debenham, R.M., Garth, S.C.J., The detection of eyes in facial images using radial basis functions. In R. Linggard, D.J. Myers, C. Nightingale (eds.), Neural Networks for Images, Speech and Natural Language. New York: Chapman & Hall, 1992.

Dennett, D., "The Hard Question: And Then What Happens ?", International Conference on Consciousness, KCL, London, 1999.

Dennett, D., Kinsbourne, M., "Time and observer: ", Behavioral and Brain Sciences, 1995.

DeValois, K.K., et al, "Responses of striate cortex cells to gratings and checkerboard patterns", Journal of Physiology, 291, pp. 483-505, 1979.

Eckhorn, R., "Oscillatory and non-oscillatory synchronisations in the visual cortex and their possible roles in associations of visual features", Progress in Brain Research, 102, pp. 405-426, 1994.

Edelman, G.M., "The Remembered Present: A biological theory of Consciousness", Basic Books, New York, 1989.

Engel, A.K., Konig, P., Kreiter, A.K., Schillen, T.B. and Singer, W., "Temporal coding in the visual cortex: New vistas on integration in the nervous system", Trends in Neurosciences, 15:218-226, 1992.

Fodor, J., Pylyshyn, Z., "Connectionism and Cognitive Architecture: A Critical Analysis". In M.A. Boden (ed.), The Philosophy of Artificial Intelligence, Oxford University Press, 1990.

Fogel, L.J., “Autonomous automata”, Industrial Research, 4, pp. 14-19, 1962.

Franklin, S., Graesser, A., “Is It an Agent or just a Program?: In A Taxonomy for Autonomous Agents”, Intelligent Agents III. Proceedings of the Third International Workshop on Agents Theories, Architectures, and Languages, Springer Verlag, 1997.

Franklin, S., private communication, 1999.

Galil, Z., Giancarlo, R., “Data Structures and Algorithms for Approximate String Matching”, Journal of Complexity, 4, pp. 33-72, 1988.

Geary, J., “A trip down the memory’s lanes”, Time, pp. 43-49, 5 May, 1997.

Gilks, W.R., Richardson, S., Spiegelhalter, D.J., (eds.), “Markov Chains Monte Carlo in Practice”, London, Chapman & Hall, 1996.

Goldberg, D., “Genetic Algorithms in search, optimization and machine learning”, Addison Wesley, Reading, MA, 1989.

Grech-Cini, E., Motion tracking in video films. PhD Thesis, University of Reading, 1995.

Hand, C.C., et al, A neural network feature detector using multi-resolution pyramid. In R. Linggard, D.J. Myers, C. Nightingale (eds.), Neural Networks for Images, Speech and Natural Language. New York: Chapman & Hall, 1992.

Hinton, G.E., “A parallel computation that assigns canonical object-based frames of reference”, Proceedings of 7th International Joint Conference on Artificial Intelligence, 1981.

Hoffman, Ch.M., O’Donnell, J., “Pattern Matching in Trees”, Journal of ACM, 29(1), pp. 68-95, 1982.

Hofstadter, D., “Fluid Concepts and Creative Analogies. Computer Models of the Fundamental Mechanisms of Thought”, The Penguin Books, 1998.

Holland, J.H., “Adaptation in natural and artificial systems”, The University of Michigan Press, Ann Arbor, MI, 1975.

Horn, R., Johnson, Ch., Matrix Analysis, Cambridge, Cambridge University Press, 1994.

Hubel, D.H., Wiesel, T.N., “Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex”, Journal of Physiology, 160, pp. 106-154, 1962.

Iosifescu, M., Finite Markov processes and their applications, Chichester, Wiley, 1980.

Iosifescu, M., Theodorescu, R., Random processes and learning, Berlin, Springer, 1969.

Jackson, J.V., “Idea for a mind”, SIGART Newsletter, 101, 1987.

Johnson, N.L., Kotz, S., Distributions in statistics: discrete distributions, New York Chichester, Wiley, 1969.

Jones, J.P., Palmer, L.A., “The Two-Dimensional Spatial Structure of Simple Receptive Fields in Cat Striate Cortex”, Journal of Neurophysiology, 58(6), pp. 1187-1211, 1987a.

Jones, J.P., Palmer, L.A., “The Two-Dimensional Spatial Structure of Simple Receptive Fields in Cat Striate Cortex”, Journal of Neurophysiology, 58(6), pp. 1211-1232, 1987b.

Jones, J.P., Palmer, L.A., “The Two-Dimensional Spatial Structure of Simple Receptive Fields in Cat Striate Cortex”, Journal of Neurophysiology, 58(6), pp. 1233-1258, 1987c.

Karp, R.M., Rabin, M.O., “Efficient randomized pattern-matching algorithms”, IBM Journal on Research Development, 31(2), pp. 249-260, 1987.

Kemeny, J.G, Snell, J.L., Finite Markov Chains, New York: Springer-Verlag, 1976.

Kim, D.S., Smith, R.L., “An Exact Aggregation/Disaggregation Algorithm for Large Scale Markov Chains”, Naval Research Logistics, 42, pp. 115-1128, 1995.

Kirkpatrick, S., Gelatt, C.D. Jr., Vecchi, M.P., “Optimization by simulated annealing”, Science, 220, pp. 671-680, 1983.

Knuth, D.E., Morris, J.H., Pratt, V.R., “Fast pattern matching in strings”, SIAM Journal of Computation, 6(2), pp. 323-350, 1977.

Koenig, P., Engel, A.K., Singer, W., “Integrator or coincidence detector? The role of the cortical neuron revisited”, Trends in Neurosciences, 19(4), pp. 130-137, 1996.

Krekelberg, B., Taylor, J.G., “Nitric oxide: what can it compute?”, Network: Computations in Neural Systems, 8, pp. 1-16, 1997.

Larsen, A., Bundesen, C., “A template-matching pandemonium recognises unconstrained handwritten characters with high accuracy”, Memory and Cognition, 24(2), pp. 136-143, 1996.

Larsen, A., Bundesen, C., “The efficiency of holistic template matching in the recognition of unconstrained handwritten digits”, Psychological Research, 545, pp. 187-193, 1992.

Li, S.Z., ”Matching Invariant to Translations, Rotations and Scale Changes”, Pattern Recognition. 25(6), pp. 583-594, 1992.

Lindsay, P.H. Norman, D.A., “Human Information Processing. An Introduction to Psychology”, Academic Press, London, 1977.

Lisboa, P.J.G., Image classification using Gabor representation with a neural net. In R. Linggard, D.J. Myers, C. Nightingale (eds.), Neural Networks for Images, Speech and Natural Language. New York: Chapman & Hall, 1992.

Liu, J., “Reactive agents for Adaptive Image Analysis”, Proceedings of the Second International Conference on Autonomous Agents, Minneapolis, 1998.

Lucas, J.R., “Minds, Machines and Gödel: A Retrospect”. In P.J.R. Millican, A. Clark, (eds.), Machines and Thought. The Legacy of Alan Turing, Clarendon Press, Oxford, 1996.

Marcelja, S., “Mathematical description of the responses of simple cortical cells”, Journal of Optical Society of America, 70(11), pp. 1297-1300, 1980.

Marr, D., Vision, San Francisco: W.H. Freeman and Co, 1982.

Maturana, H., Varela, F.J., “Autopoiesis and Cognition: The realisation of the living”, Boston, D. Reidel, 1980.

Minsky, M. “A Framework for Representing Knowledge”. In P.H. Winston, (ed.), The Psychology of Computer Vision, McGraw-Hill, New York, pp. 211-277, 1975.

Mitchell, A., “Liquid genius”, New Scientist, pp. 26-30, 13 March 1999.

Montague, P.R., “General properties of the resource consumption principle of neural function”, Journal of Physiology, 90, pp.239-242, 1996.

Nasuto, S.J., Bishop, J.M., “Convergence Analysis of Stochastic Diffusion Search”, Journal of Parallel Algorithms and Applications, 14, pp. 89-107, 1999.

Nasuto, S.J., Bishop, J.M., Lauria, S., “Time Complexity of Stochastic Diffusion Search”. Neural Computation ‘98, Vienna, Austria.

Nasuto, S.J., Dautenhahn, K., Bishop, J.M., Communication as an emergent metaphor for neuronal operation. In Ch. L. Nehaniv (ed.), Computation for Metaphors, Analogy, and Agents, Lecture Notes in Artificial Intelligence 1562, Springer Verlag, 1999.

Nicholson, C., “Signals that go with the flow”, Trends in Neurosciences, 22(4), pp. 143-145, 1999.

Penrose, R., “Shadows of the Mind. A Search for the Missing Science of Consciousness”, Oxford University Press, Vintage, 1995.

Penrose, R., “The Emperor’s New Mind. Concerning computers, minds, and the laws of physics”, Oxford University Press, Vintage, 1990.

Pollen, D.A., Ronner, S.F., “Visual Cortical Neurons as Localised Spatial Frequency Filters”, IEEE on System, Man and Cybernetics, 13(5), pp. 907-916, 1983.

Rose, D., “’Labelled lines’ as a Ryleian category error”, Presented at the European conference on Visual Perception, Oxford, UK, 1998. Abstracted in Perception, 27, pp. 76-77, 1998.

Rose, D., “The ghost in the nerve cell: a critique of labelled line theory”. Presented at: Towards a Science of Consciousness III, Tucson, USA, 1998. Journal of Consciousness Studies: Consciousness Research Abstracts 97, 1998.

Rubino, G., Sericola, B., “On weak lumpability in Markov chains, Journal of Applied Probability, 27, pp. 744-456, 1989.

Rubino, G., Sericola, B., “On weak lumpability in Markov chains”, Journal of Applied Probability, 26, pp. 446-457, 1989.

Rumelhart, D.E, McClelland, J.L., and The PDP group, Parallel Distributed Processing. Explorations in the Microstructure of Cognition. Cambridge, MA: The MIT Press, 1986.

Russo, E., “Controversy Surrounds Memory Mechanism. Has the long-held theory of long-term potentiation become obsolete?”, the Scientist, 13(5), pp. 1-5, 1999.

Saaty, T.L., “Modern Nonlinear Equations”, New York, Dover Publications, 1981.

Schweitzer, P., Aggregation methods for large Markov chains. In Iazeolla, G., et al., Mathematical Computer Performance and Reliability. Amsterdam: Elsevier, 1984.

Selfridge, O.G., “Pandemonium: A paradigm for learning”, H.S.M.O., pp. 511-527, 1958.

Selman, B., “Challenge Problems for Artificial Intelligence”, Proceedings of AAAI-96, Menlo Park, CA, USA, 1996.

Seneta, E., Nonnegative Matrices, London, George Allen & Unwin Ltd., 1973.

Serra, R., Zanarini, G., “Complex Systems and Cognitive Processes”, Springer, Berlin, 1990.

Shapiro, L.G., Haralick, R.M., “Structural Description and Inexact Matching”, IEEE Transactions on Pattern Analysis and Machine Intelligence. 3, pp. 504-519, 1981.

Shastri, L., “Structured Connectionist Models”, In M.A. Arbib (ed.), The Handbook of Brain Theory and Neural Networks”, The MIT Press, Cambridge Massachusetts, 1995.

Shastri, L., Ajjanagade, V., “From simple associations to systematic reasoning: A connectionist representation of rules, variables and dynamic bindings”, Behavioral and Brain Sciences, 16, pp. 417-494, 1993.

Smieja, F., “The Pandemonium System of Reflective Agents”, IEEE Transactions on Neural Networks, 7(1), pp. 97-106, 1996.

Sneppen, K., Bak, P., Flyvbjerg, H, Jensen, M.H., “Evolution as a self-organised critical phenomenon“ Proc. Natl. Acad. Sci., 92, pp. 5209-5213, 1995.

Tarhio, J., “A sublinear algorithm for two-dimensional string matching”, Pattern Recognition Letters, 17(8), pp. 833-846, 1996.

Thorpe, S.J., Imbert, M., “Biological constraints on connectionist modelling”. In R. Pfeifer et al. (eds.), Connectionism in Perspective, Elsevier, 1989.

Ullmann, S., “High-level Vision”, The MIT Press, Cambridge Massachusetts, 1996.

Ullmann, S., “Sequence Seeking and Counterstreams: A Model for Bidirectional Information Flow in the Cortex”. In Large-Scale Neuronal Theories of the Brain, Ch. Koch, Davis, J.L (eds.), MIT Press, 1995.

van Laarhoven, P.J.M., Aarts, E.H.L., Simulated Annealing: Theory and Applications. Kluwer, Dordrecht, 1987.

van Leeuwen, J. (ed.), Handbook of Theoretical Computer Science. Algorithms and Complexity, Amsterdam: Elsevier, 1990.

Varela, F.J., “Patterns of Life: Interwinning Identity and Cognition”, Brain and Cognition, 34, pp. 72-87, 1997.

Varga, R.S., “Matrix Iterative Analysis”, 1962.

Vincent, J.M., et al, Image feature location in multi-resolution images using a hierarchy of multilayer perceptrons. In R. Linggard, D.J. Myers, C. Nightingale (eds.), Neural Networks for Images, Speech and Natural Language. New York: Chapman & Hall, 1992.

von der Malsburg, Ch., “Dynamic Link Architecture”, In The Handbook of Brain Theory and Neural Networks”, M.A. Arbib (ed.), The MIT Press, Cambridge Massachusetts, 1995.

von der Malsburg, Ch., “The correlation theory of the brain function”, Internal report 81-2, Gottingen, Germany: Max-PLanck-Institute fur Biophysicalishe Chemie. Reprinted in Models of Neural Networks, K. Schulten, H.-J. van Hemmen (eds.), Springer Verlag, 1994.

Waite, J.B., Training multilayer perceptrons for facial feature location: a case study. In R. Linggard, D.J. Myers, C. Nightingale (eds.), Neural Networks for Images, Speech and Natural Language. New York: Chapman & Hall, 1992.

Wechsler, H., Computational Vision, London:Academic Press Inc, 1990.

Wegner, P., “Interactive Software Technology”. In Handbook of Computer Science and Engineering, CRC Press, 1997.

Whittle, P., “Systems in Stochastic Equilibrium”, London, John Wiley & Sons, 1986.

Wilson, H.R., Bergen, J.R., “A four mechanism model for spatial vision”, Vision Research, 19, pp. 19-32, 1979.

Winograd, T., Flores, F., “Understanding Computers and Cognition. A New Foundation for Design”, Adison-Wesley, Reading Massachusetts, 1987.

Wiskott, L., von der Malsburg, Ch., “Face Recognition by Dynamic Link Matching”. In Lateral Interactions in the Cortex. Structure and Function Sirosh, J., Miikkulainen, R., Choe, Y., (eds.), Hypertext Book, ISBN 0-9647969-0-8, http://www.cs.utexas.edu/users/nn/web-pubs/htmlbook96/, 1996.

Wooldridge, M., “Intelligent Agents: Theory and Practice”. Knowledge Engineering Review, 10(2), 1995.