Steady State Resource Allocation Analysis of the Stochastic Diffusion Search

Slawomir J. Nasuto[c], J. Mark Bishop[d]

[a] *University of Reading, UK*
[b] *Goldsmiths, University of London, UK*

Abstract

This article presents the long-term behaviour analysis of Stochastic Diffusion Search (SDS), a distributed agent based Swarm Intelligence meta-heuristic for best-fit pattern matching. SDS operates by allocating simple agents into different regions of the search space. Agents independently pose hypotheses about the presence of the pattern in the search space and its potential distortion. Assuming a compositional structure of hypotheses about pattern matching agents perform an inference on the basis of partial evidence from the hypothesised solution. Agents posing mutually consistent hypotheses about the pattern support each other and inhibit agents with inconsistent hypotheses. This results in the emergence of a stable agent population identifying the desired solution. Positive feedback via diffusion of information between the agents significantly contributes to the speed with which the solution population is formed.

The formulation of the SDS model in terms of interacting Markov Chains enables its characterisation in terms of the allocation of agents, or computational resources. The analysis characterises the stationary probability distribution of the activity of agents, which leads to the characterisation of the solution population in terms of its similarity to the target pattern.

Keywords: Generalised Ehrenfest Urn model, interacting Markov Chains, non-stationary processes, resource allocation, best-fit search, distributed agents based computation,

[*] Please address correspondence to Nasuto or Bishop
Email addresses: s.j.nasuto@reading.ac.uk (Slawomir J. Nasuto), m.bishop@gold.ac.uk (J. Mark Bishop)

Preprint submitted to BICA *April 7, 2015*

Steady State Resource Allocation Analysis of the Stochastic Diffusion Search

Slawomir J. Nasuto[c], J. Mark Bishop[d]

[c] *University of Reading, UK*
[d] *Goldsmiths, University of London, UK*

1. Introduction

In recent years there has been growing interest in swarm intelligence, a distributed mode of computation utilising interaction between simple agents (Kennedy et al., 2001). Such systems have often been inspired by observing interactions between social insects: ants, bees, birds (cf. Ant Algorithms and Particle Swarm Optimisers) see Bonabeau (Bonabeau et al., 1999) for a comprehensive review. Swarm Intelligence algorithms also include methods inspired by natural evolution such as Genetic Algorithms (Goldberg, 1989) (Holland, 1975) or indeed Evolutionary Algorithms (Back, 1996). The problem solving ability of Swarm Intelligence methods, emerges from positive feedback reinforcing potentially good solutions and the spatial/temporal characteristics of their agent interactions.

Independently of these algorithms, Stochastic Diffusion Search (SDS), was first described in 1989 as a population-based, swarm intelligence pattern matching algorithm (Bishop, 1989). Unlike stigmergetic communication employed in Ant Algorithms, which is based on modification of the physical properties of a simulated environment, SDS uses a form of direct communication between the agents similar to the tandem calling mechanism employed by one species of ants, Leptothorax Acervorum, (Moglich et al., 1974).

SDS is an efficient probabilistic multi-agent global search and optimisation technique (de Meyer et al., 2006) for solving the best-fit matching problem. Many fundamental problems in Computer Science, Artificial Intelligence or Bioinformatics may be formulated in terms of pattern matching or search. Examples abound in DNA or protein structure prediction, where virtually all the deterministic methods employed are solving variations of the string matching problem (Gusfield, 1997). In addition the classical exact string matching problem has been extended to the approximate matching, where one allows for a pre-specified number of errors to occur. Other variations include consideration of various distances used for the determination of similarity between patterns, see (Navarro, 1998) for an extensive overview of an approximate string matching algorithms. String matching can be generalised to tree matching and many algorithms used for string matching are easily adapted to this significant class

[*] Please address correspondence to Nasuto or Bishop
Email addresses: s.j.nasuto@reading.ac.uk (Slawomir J. Nasuto), m.bishop@gold.ac.uk (J. Mark Bishop)

of problems (van Leeuwen, 1990). Thus, it is very important to develop new efficient methods of solving string-matching problems.

SDS has now been successfully deployed across a wide variety of application domains including: site selection for wireless networks (Whitaker and Hurley, 2002), mobile robot self-localisation (Beattie and Bishop, 1998), object recognition (Bishop and Torr, 1992) and text search (Bishop, 1989). Additionally, a hybrid SDS and n-tuple RAM (Aleksander and Stonham, 1979) technique has been used to track facial features in video sequences (Bishop and Torr, 1992; Grech-Cini, 1995); other hybrids have been explored to comprehensively characterise the behaviour of merger of SDS with a variety of local optimisers on continuous optimisation problems across a broad range of modern benchmark suites; amongst other these include SDS-PSO hybrid systems (al-Rifaie et al., 2011) and SDS-Differential Evolution hybrid systems (al-Rifaie and Bishop, 2013). Furthermore SDS has been successfully deployed in NP-Hard problems: for example, in 2012 SDS was applied to the rectilinear Steiner minimum tree problem (Li and J., 2012); conversely, as an illustration of the diversity of its application, SDS has also been successfully deployed in 'artistic' applications (al-Rifaie et al., 2012). In addition a connectionist implementation of SDS was described in (Nasuto et al., 2009). See (al-Rifaie and Bishop, 2013) for a recent comprehensive review paper detailing SDS, the algorithm variants and its numerous fielded applications.

The last decades have witnessed an increased interest in various forms of heuristic search methods, as alternative solutions to hard computational problems. This arose due to recognition of the limitations of fully specified deterministic methods based on sequential operation and computation. Such heuristic methods include Genetic Algorithms, Evolutionary Strategies, Ant Colony, or Simulated Annealing (Holland, 1975) (Back, 1996) (Colorni et al., 1991) (Kirkpatrick et al., 1983). They often base their computation on some form of iterative parallel random sampling of the search space by a population of computational agents, where the sampling bias is a function of coupling mechanisms between the agents and thus is specific to the particular method employed. For example Genetic Algorithms (Goldberg, 1989) are loosely inspired by the natural evolution and the mechanisms by which GA agents sample the search space are most often described in terms of gene recombination and mutation. Ant Colony Optimisation is based on the modelling of the interaction between simple organisms like ants or termites (Dorigo, 1992). These methods can be collectively described as heuristic search methods, as their generic formulation is based on very simplified intuitions about some natural processes. However, in spite of considerable interest in these algorithms and their wide areas of applications, they still lack a standard formal framework enabling principled inference about their properties. It can be argued that the very central and indeed appealing feature of these heuristic search methods (i.e. imitating solutions to hard computational problems found by Nature) impedes development of their theoretical basis.

However, randomness in computation has also been employed in more classical computational schemes, like RANSAC[1], Random Global Optimisation,

[1] Coupled SDS - a variant of the algorithm described herein - evaluated well against RANSAC on a variety of classical 'parameter estimation' problems (Williams and Bishop,

Monte Carlo Markov Chains or related to them Particle Filters (Fischler and Bolles, 1981) (Zhigljavsky, 1991) (Gilks et al., 1996) (Doucet et al., 2000). In these algorithms biased random sampling is employed in order to avoid local minima over the search spaces or to approximate values of otherwise intractable functions. The theoretical framework of these algorithms is usually within the area of discrete stochastic processes and the understanding of their properties and behaviour is much more sound than that of their heuristic counterparts.

Simulated Annealing and Extremal Optimisation belong to a small class of heuristic search algorithms taking inspiration from physical processes, which nevertheless enjoy a sound theoretical basis (van Laarhoven and Aarts, 1987) (Bak et al., 1987).

Some fundamental properties of SDS have been previously investigated within the framework of Markov Chain theory (Nasuto and Bishop, 1999) (Nasuto et al., 1998). In particular, it has been proven that SDS converges to the globally best fit matching solution to a given template (Nasuto and Bishop, 1999) and that its time complexity is sub-linear in the search space size (Nasuto et al., 1998) under a variety of search conditions (Myatt et al., 2004), thus placing it among the state of the art search algorithms (van Leeuwen, 1990).

SDS operates by allocating simple agents into different regions of the problem search space. Agents independently pose hypotheses about the presence of a target pattern and its potential distortion in the search space. Agents utilise the compositional structure of hypotheses about matching patterns by performing an inference on the basis of only partial evidence. A population of agents posing mutually consistent hypotheses about the pattern signifies the desired solution. This population is formed very rapidly as the search progresses due to diffusion of information between the agents.

In this article we describe steady state analysis of Stochastic Diffusion Search (SDS), characterising how SDS achieves its tasks by distributing computational resources while focusing on the solution. By complementing previous results on global convergence and time complexity of SDS, the resource allocation analysis constitutes a significant advancement of our understanding of this algorithm operation. It appears that SDS is very robust and is capable of differentiating between very small differences between patterns. This is achieved due to positive feedback leading to a rapid growth of clusters inducing strong competition between them due to the limitation of agent resources. Thus SDS converges to the best-fit solution by allocating to it the largest cluster of agents. However, the dynamic aspect of the process ascertains that the remaining amount of agents, currently not assigned to any cluster, continue to explore the rest of the search space, thus ensuring the global convergence. In this paper we characterised quantitatively the distribution of agents during the search and obtained the expected size and variance of the cluster corresponding to the desired solution. We also characterised their dependence on the characteristics of the search space and the quality of the best solution.

The model of SDS used in this paper is based on a novel concept of interacting Markov Chains (Nasuto and Bishop, 1999). It concentrates on modelling the behaviour of individual agents as opposed to modelling the whole population. This is a clear advantage as the size of the state space corresponding to

2014).

the whole population of agents grows quadratically with their number; the state space of single agents is very small. However, in SDS agents are interacting with each other and their actions depend on actions of other agents. The evolution of the whole population of agents is thus reflected in coupling between the Markov Chains. The dependence of the transition probabilities of single chains not only on their immediate past but also on the state of other chains introduces non-linearity into the system description. However, it is possible to show that there exists a system of decoupled chains such that its long-term behaviour is the same as that of interacting Markov Chains modelling SDS. The decoupled system of Markov Chains with such behaviour forms a generalisation of an Ehrenfest Urn model of irreversible processes introduced in statistical physics at the beginning of this century (Ehrenfest and Ehrenfest, 1907).

The formulation of an Ehrenfest Urn model with an equivalent long-term behaviour constitutes a basis for the quantitative characterisation of SDS steady state distribution of the computational resources discussed in this paper.

The paper is organised as follows. Section 2 introduces Stochastic Diffusion Search and presents the relationship between its model based on interacting Markov Chains and the generalised Ehrenfest Urn model. Section 3 characterises the long-term behaviour of an agent in SDS. This, together with characterisation of the dependence of the steady state agent distribution on the search conditions, presented in the next section, forms the basis of the characterisation of resource allocation. The numerical simulations of SDS steady state behaviour are presented in section 5. The final section discusses the results and future work.

2. Stochastic Diffusion Search and the Ehrenfest Urn Model

Stochastic Diffusion Search employs a population of simple computational units or agents searching in parallel for the pre-specified template in the search space. Agents take advantage of the compositional nature of the statistical inference about templates. This property ensures that the hypothesis about the template can be factored into hypotheses about its constituents. At each step agents perform an inference about the presence of the template in the currently investigated region of the search space by posing and testing hypothesis about a randomly selected templates sub-component. Each of the agents makes a decision about which region of the search space to explore on the basis of its history and the information it gets from other agents in the so called diffusion phase (see Algorithm 1). This stage enables agents that failed their inferential step (inactive agents) to bias their consecutive sampling of the search space towards the region of the search space, which led to a successful test by another, randomly chosen agent (active agent). This mechanism ensures that regions with higher probability of positive testing will gradually attract higher numbers of agents; regions of low positive testing attracting agents only sporadically. Alternatively, had there been no sufficient reason to copy the position inspected by the chosen agent, inactive agents have a possibility to resample entirely new positions from the search space. While in the next iteration active agents attend to the same position they tested previously thus exploiting current patterns for further evidence regarding matching to the template. The random resampling of the search space by inactive agents ensures that it will be sufficiently explored.

Patterns of highest overlap with the template in the search space induce a form of local positive feedback, because the more agents investigate these regions the more likely is an increase of their number in the next step due to the diffusion phase. This local positive feedback induces a competition between best-fit patterns due to limited (computational) resources; the largest cluster of agents concentrated on the globally best solution emerges rapidly suppressing sub-optimal solutions. Thus, SDS performs the best-fit pattern matching; it will find the pattern with the highest overlap with the template. This is an extension of a concept of exact and approximate matching in computer science in which the problem is to locate either exact copy or an instantiation differing from the template by a pre-specified number of components (van Leeuwen, 1990).

Algorithm 1 Schematic of Stochastic Diffusion Search

```
initialise;
repeat
    diffuse;
    test;
until termination.
```

3. Interacting Markov Chains Model of the Stochastic Diffusion Search

Consider a noiseless search space in which there exists a unique object with a non-zero overlap with the template - a desired solution. Assume that upon a random choice of a feature for testing a hypothesis about the solution location, an agent may fail to recognise the best solution with a probability $p^- > 0$. Assume further that in the n^{th} iteration m out of a total of N agents are *active* (i.e. tested successfully). Then the following transition probability matrix describes a one-step evolution of an agent:

$$P_n = \begin{array}{c c} & \begin{array}{cc} a & n \end{array} \\ \begin{array}{c} a \\ n \end{array} & \left[\begin{array}{cc} 1-p^- & p^- \\ p_1^n & 1-p_1^n \end{array} \right] \end{array},$$

where,

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

p_m is a probability of locating the solution in the search space by uniformly random sampling and the active agent pointing to the correct solution is denoted by a and an *inactive* agent (i.e. agent that failed the test) by n.

Thus, if an agent was active in the previous iteration then it will continue to test the pattern at the same position but using another, randomly chosen sub-component. Therefore with probability $1 - p^-$ it may remain active (tests positive), otherwise it becomes inactive. An inactive agent may become active either by choosing at random an active agent and testing positively at its position (the first term in p_1^n) or otherwise by testing positively upon a random resampling of the search space.

It is apparent that the above stochastic process modelling an evolution of an agent is a non-homogenous Markov Chain. Non-homogeneity 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. The time dependence of P reflects the interactions with other agents in the population. Thus, the population of Markov Chains defined above describes the evolution of a population of agents in SDS.

The model capturing the agent's behaviour is similar to the model formulated by Ehrenfests in 1907 in order to resolve an apparent irreversibility paradox in the context of statistical mechanics (Ehrenfest and Ehrenfest, 1907). In fact, we will demonstrate that the long-term behaviour of SDS can be recovered from an appropriately constructed generalised Ehrenfest Urn model.

The Ehrenfest Urn 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. Whittle discussed a generalisation of this model consisting 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 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.

There is an important difference between the generalised Ehrenfest Urn Model and Interacting Markov Chains model of SDS. In the former the Markov Chains are independent and stationary, whereas in the latter the Markov Chains are interacting. In addition, the coupling between Markov Chains induces their non-homogeneity - their transition probability matrices change in time. However, we will prove that the long-term behaviour of the Interactive Markov Chains model can be described in terms of an appropriately constructed generalised Ehrenfest Urn model.

The analysis based on this model will characterise the stationary probability distribution of the activity of a population of agents. This will allow calculation of the expected number of agents forming a cluster corresponding to the best-fit solution as well as the variation of the cluster size. The model will also enable characterisation of the long-term behaviour of SDS in terms of the statistical properties of the search space.

4. Long term behaviour of the Stochastic Diffusion Search

In order to investigate the Markov Chain model of agent's evolution we will establish first its long-term behaviour. The following propositions about single agent evolution can be proven:

Proposition 1 The sequence $\{P_n\}$ of stochastic matrices describing the evolution of an agent in the Stochastic Diffusion Search is weakly ergodic. *Proof.* See Appendix.

Proposition 2 $\{P_n\}$ is asymptotically stationary. *Proof.* See Appendix.

Proposition 3 The sequence $\{P_n\}$ is strongly ergodic. *Proof.* Strong ergodicity of $\{P_n\}$ follows as a consequence of the above propositions, and theorem 4.11 in Seneta (Seneta, 1973).

From Propositions 1-3 it follows that the Markov Chain corresponding to the evolution of a single agent is asymptotically homogenous, i.e. the limit

$$P = \lim_{i \to \infty} P_i$$

exists. Thus this process behaves, as it approaches the equilibrium, more and more like a homogenous Markov Chain with the transition probability matrix P. Therefore, instead of considering a population of interacting Markov Chains we will construct and consider a generalised Ehrenfest Urn model consisting of homogenous Markov Chains with the transition probability matrix P.

First we will obtain an explicit formula for the matrix P. The strong ergodicity of a non-homogenous Markov Chain amounts to the existence of its (unique) equilibrium probability distribution. In other words, a stochastic process describing the time evolution of an agent as it visits states of its state space with frequencies, which can be characterised via the limiting probability distribution.

Thus, in order to find the limit probability distribution one has to find the solution of the fixed-point equation

$$S(\rho, P) = (\rho, P)$$

(see Appendix for the definition of the mapping S).

This amounts to solving a system of two equations

$$\begin{cases} \pi P = \pi, \\ P_1 = (1 - p^-)\pi_1 + (1 - p^-)(1 - \pi_1)p_m \end{cases} \tag{1}$$

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

$$\pi = (\pi_1, 1 - \pi_1) = \left(\frac{p_1}{P_1 + p^-}, \frac{p^-}{P_1 + p^-} \right) \tag{2}$$

This makes it possible to find the solution in the special case when $p_m = 0$ (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 (2) and the second equation of (1) it follows that

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

which after rearrangement leads to a quadratic equation in π_1

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

This can be written in the form

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

This equation has two solutions because the condition

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

is always fulfilled. These solutions are as follows

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

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

$$\pi = (\pi_1, \pi_2) = (r, s) \tag{3}$$

where

$$r = \frac{2(1-p^{-})(1-p_m) - 1 \pm \sqrt{[2(1-p^{-})(1-p_m)-1]^2 + 4(1-p^{-})^2(1-p_m)p_m}}{2(1-p^{-})(1-p_m)}$$

$$s = \frac{1 - \sqrt{[2(1-p^{-})(1-p_m)-1]^2 + 4(1-p^{-})^2(1-p_m)p_m}}{2(1-p^{-})(1-p_m)}$$

As the long term evolution of an agent is approaching the evolution of a homogenous Markov Chain with the transition matrix P, we can characterise the limit behaviour of the Interactive Markov Chains model of SDS by finding the behaviour for the corresponding generalised Ehrenfest Urn model, in which the probability transition matrix P governing the transitions of a ball is

$$P = \begin{array}{cc} & \begin{array}{cc} a & n \end{array} \\ \begin{array}{c} a \\ n \end{array} & \left[\begin{array}{cc} 1-p^{-} & p^{-} \\ p_1 & 1-p_1 \end{array}\right] \end{array},$$

where,

$$p_1 = (1-p^{-})\pi_1 + (1-p^{-})(1-\pi_1)p_m$$

and π_1 is given by (3).

It is important to note that in the above Ehrenfest Urn model the transition of balls between the urns is now mutually independent. This is because the dependence between Markov Chains in the Interacting Markov Chains model of SDS was reflected in the changes of their probability transition matrices. By using a limit probability transition matrix P we make the balls independent. Thus in order to characterise the limit probability distribution we will proceed analogously to Whittle (Whittle, 1986).

Modelling SDS via evolution of single agents, as proposed in the model, implies distinguishability of agents. This is true also in the case of the balls in the corresponding generalised Ehrenfest Urn model. From this it follows that the state space of the ensemble of Markov Chains corresponding to N balls in the constructed Ehrenfest Urn model consists of N-tuples $x = (x_1, ..., x_N)$ where the i^{th} component takes on a value 1 or 0 depending on whether the i^{th} ball (agent in SDS) was in urn 1 or 2 (was active or not). Because of ergodicity of Markov Chains describing evolution of balls and their independence there will be a unique asymptotic limit probability distribution given by:

$$\Pi(X) = \pi_1^{a[x]}\pi_2^{N-a[x]} \tag{4}$$

where $a[x]$ denotes number of balls in the urn 1 (active agents) corresponding to the state x and (π_1, π_2) is the two state asymptotic distribution given by (3).

In order to establish the equilibrium probability distribution of SDS one has to abandon the distinguishability of balls implied by the construction of the generalised 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 balls in the first urn 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 balls, which preserve the statistics of the process, [Whittle (Whittle, 1986) theorem 3.7.2]. This can be seen from the fact, that permuting two arbitrary balls in the same urn 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) over all configurations x corresponding to the same number of balls in the first urn $a[x]$ one obtains a probability distribution of the generalised Ehrenfest Urn model and hence that of SDS:

$$\pi(n) = \binom{N}{n}\pi_1^n\pi_2^{N-n} \tag{5}$$

which is a binomial distribution.

Equation (5) 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. We can characterise the resource allocation of SDS in the steady state by computing the expected distribution of agents. The deviations from the expected distribution can be characterised in terms of the standard deviation of the ensemble probability distribution. From equation (5) it follows immediately that the expected number of active agents in the equilibrium will be

$$E[n] = N\pi_1 \tag{6}$$

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

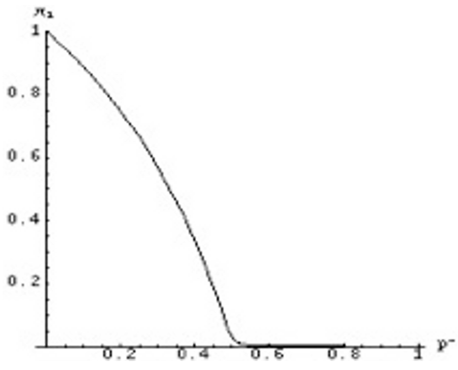

Figure 1: The normalised average number of active agents in SDS as a function of the parameter p^-. Plot obtained for $N = 1000$ and $p_m = 0.001$

Similarly the standard deviation defined for binomial distribution as

$$\sigma = \sqrt{N\pi_1\pi_2} \tag{7}$$

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 a 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 quasi-equilibrium.

The above discussion closely parallels the reasoning motivating the 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 thermodynamic quantities.

The next section will illustrate some of these points with a numerical example and will characterise the resource allocation of SDS in terms of the search space and the properties of the best solution.

5. Characterisation of the resource allocation

The population of active agents converging to the best-fit solution is very robust. The convergence rate and the best-fit population size clearly differentiate SDS from a pure parallel random search.

Below we will characterise the behaviour of the average number of active agents as a function of the p^- parameter characterising the possibility of false negative response of agents for the best-fit solution. Figure (1) illustrates this relationship for the value of $p^m = 0.001$ and $N = 1000$ agents and Figure (2) shows the two dimensional plot of the average number of active agents as a function of both parameters p_m and p^-.

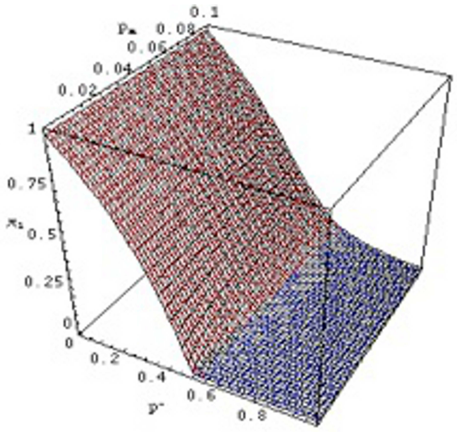

Figure 2: The normalised mean number of active agents as a function of both the false negative parameter p^- and the probability of hit at the best instantiation p_m, plotted for $N = 1000$.

Figure (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 (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 (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 inspection of this figure it follows that changing p_m does not significantly alter 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 of the boundary between two regions of behaviour of SDS clearly visible in Figure (1).

Similarly, it is possible to investigate, using the equations (3) and (7), dependence of the standard deviation on parameters of SDS. Figure (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) shows the 3D plot of the standard deviation as a function of p^- and N.

From Figure (3) one can deduce that standard deviation is also a non-linear function of p^-, first rapidly increasing 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 (3)

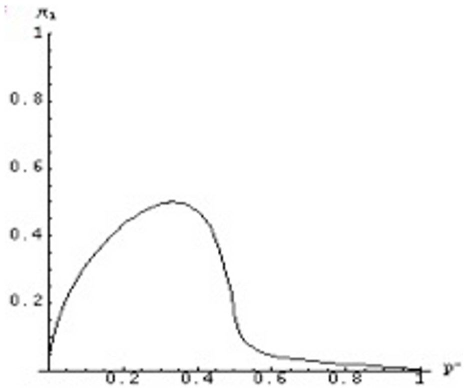

Figure 3: The rescaled standard deviation of the number of active agents calculated from the model for $N = 1000$ agents and $p_m = 0.001$; the scaling factor is $\alpha = N^{\frac{-1}{2}} \approx 0.0316$

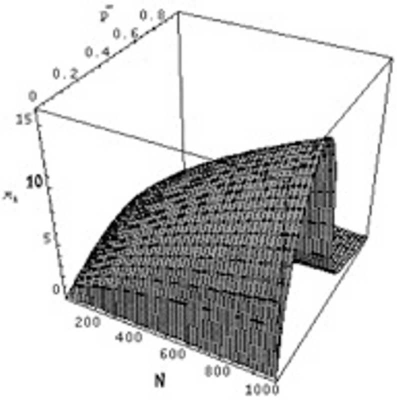

Figure 4: The standard deviation of the number of active agents as a function of the total number of agents N in SDS and of false negative probability p^-, plotted for $p_m = 0.001$

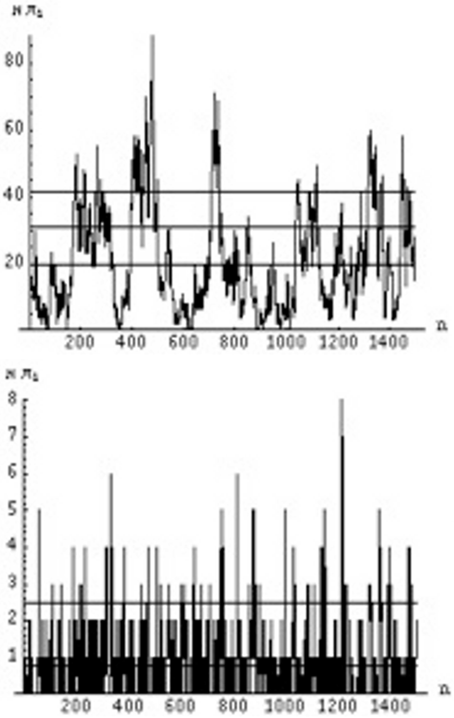

Figure 5: Evolution of the number of active agent in SDS with $N = 1000$ agents and $p_m = 0.001$. The false negative parameter p^- is 0.5 (top panel) and 0.7 (lower panel). The straight lines correspond to the average activity predicted by the model surrounded by the ±2 standard deviations band.

corresponds in fact to a cross-section of Figure (4) along the line of a fixed number of agents.

6. Simulations

In this section we will evaluate the predictions of the Interacting Markov Chains SDS model by applying SDS to perform a string search, a simple yet nontrivial problem with very important applications in bioinformatics (Gusfield, 1997), information retrieval and extensions to other data structures (van Leeuwen, 1990). The task will be to locate the best instantiation of the given string template in the longer text. The agents will perform tests by comparing single characters in the template and the hypothesised solution and during the diffusion phase they will exchange information about potential location of the solution in the text. This set-up allows us to control precisely the parameters characterising the search space and the best-fit solution, hence it constitutes an appropriate test-bed for the evaluation of our model.

We run a number of simulations in order to compare reliably 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 for 2000 iterations. In all cases the first 500 samples were discarded as a burn-in period. This method was suggested in Gilks (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.

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

Table 1: Average activities and standard deviations estimated from the 1500 iterations of SDS. $N = 1000$, $p_m = 0.001$ 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 value of the burn-in period, used in these simulations, was chosen on the basis of visual inspection, as a heuristic remedy against the bias of estimates.

The results of the simulations are summarised in Table 1 and Table 2 and in Figure (5) and Figure (6).

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

Table 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).

It follows that the model predicts very well the steady state behaviour of SDS. The biggest deviation between model prediction and empirical estimates seem to be for the value of $p^- = 0.5$, i.e. in the case when agents have only 50% chance of successful testing of the best-fit solution. This may be because this value is located in the boundary of two regions of markedly different characteristic of resource allocation exhibited by SDS and small fluctuations in the number of agents in the cluster can have significant effects.

It appears that the expected number of active agents and two standard deviations, as calculated from the model, correspond well to the empirical parameters used in (Bishop, 1989) to define statistical stability of SDS and thus defining the halting criterion used in that work. Therefore our model provides a firm ground for the calculation of these parameters and for the characterisation of their dependence on the parameters of SDS and the search space.

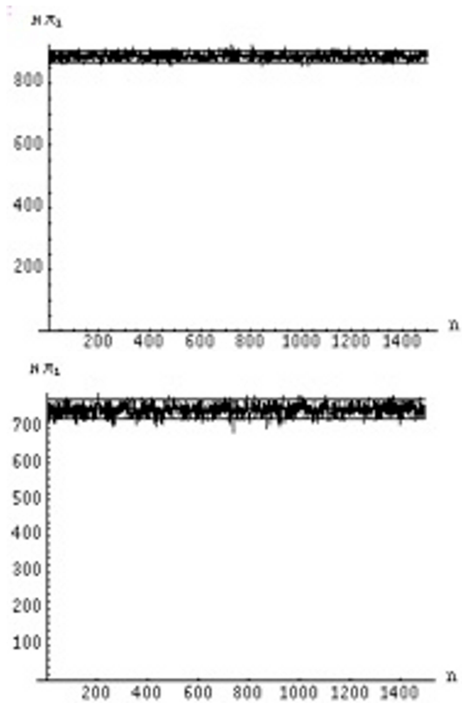

Figure 6: Evolution of the number of active agents in SDS with $N = 1000$ agents and $p_m = 0.001$ (top) The false negative parameter p^- is 0.1, (lower) $p^- = 0.2$. The straight lines correspond to the average activity predicted by the model, surrounded by the ±2 standard deviations band.

7. Conclusions

We have discussed in detail the resource allocation characteristics of SDS. It appears that SDS can rapidly and reliably allocate a large amount of agents to the best-fit pattern found in the search space, while the remaining agents continually exploring the search space for potentially better solutions. This property underlies SDS success in dealing with dynamic problems such as lip tracking, where the desired solution changes continually its position in the search space (Grech-Cini, 1995). The analysis herein revealed that SDS is very sensitive to the quality of the match of the hypothesised solution. This property combined with the limited resource induced competition is responsible for the robustness of the algorithm to the presence of sub-optimal solutions, or noise in the search space (Nasuto et al., 1998). Two key properties of SDS - the exchange of information between the agents and their relative computational simplicity illustrate the fact that the communication-based systems are as powerful as systems based on distributed computation, a thesis discussed in (Mikhailov, 1993). Stochastic Diffusion Search can be easily extended to incorporate more sophisticated or heterogeneous agents, different forms of information exchange, including local communication, voting, delays etc. This makes it an appealing tool for both solving computationally demanding problems and simulating complex systems. The latter often are composed of many interacting subcomponents and it may be easier to define the operation of subcomponents and their mutual interactions, than find emergent properties of the system as a whole. We envision that SDS can constitute a generic tool for simulating such systems, e.g. in models of ecological systems, epidemics or agent based economic models.

SDS belongs to a broad family of heuristic algorithms, which utilise some form of feedback in their operation. Neural networks, a well-known class of such algorithms are often using some form of negative feedback during learning, i.e. during process of incremental change of their parameters guided by the use of disparity between desired and current response. In contrast, SDS, together with Genetic Algorithms and Ant Colonies, belongs to category of algorithms utilising positive feedback (Dorigo, 1994). Positive feedback is responsible in SDS for its rapid convergence and, together with limited resources, for the non-linear dependence of the resource allocation on the similarity between the best-fit solution and the template. The resource allocation analysis presented in this paper, complements previous results on the global convergence and time complexity (Nasuto and Bishop, 1999) (Nasuto et al., 1998) setting a solid foundation for our understanding of its behaviour.

The resource allocation management of SDS was characterised by finding the steady state probability distribution of the (long-term behaviour) equivalent generalised Ehrenfest Urn model and observing that the actual distribution of agents is well described by the first two moments of this distribution.

It thus appeared that the resource allocation by SDS corresponds to the quasi-equilibrium state of the generalised Ehrenfest Urn model. This quasi equilibrium 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.

This analysis allowed us to understand how the resource allocation depends on the search conditions. This analysis also provides a theoretical background for the halting criterion introduced in (Bishop, 1989).

Apart of the use in analysis of SDS, the concept of interacting Markov Chains is interesting in its own right. Many computational models based on distributed computation or agent systems utilise a notion of semi autonomous computational entities, or agents. Such models are utilised in machine learning and computer science but also in models of economic behaviour (Young et al., 2000). At some level of abstraction the evolution can be described in terms of Markov Chains. Interacting Markov Chains correspond to the situation when agents are not mutually independent but interact with each other in some way. This is often the case when agents try collectively to perform some task or achieve a desired goal. The basic formulation of interacting Markov Chains is flexible and can be extended to account for other types of interactions including heterogenous agents, delays in communication, rational as well as irrational agents, etc. Thus, it offers a very promising conceptual framework for analysing such systems.

8. Appendix

Proof of Proposition 1. The proposition will be first proven for p-¿0. Recall that

$$P_n = \begin{array}{c|cc} & a & n \\ \hline a & 1-p^- & p^- \\ n & p_1^n & 1-p_1^n \end{array}, \tag{8}$$

where,

$$p_1^n = \frac{m}{n}(1-p^-) + (1-\frac{m}{n})p_m(1-p^-) \tag{9}$$

Define after Senata (Seneta, 1973):

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

Rearranging (9) leads to

$$p_1^n = (1-p^-)((1-p_m)x + p_m)$$

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

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

so

$$\begin{aligned} p_1^n \leq 1-p &\implies \min(p_1^n, 1-p^-) = p_1^n \\ &\implies \min(p, 1-p_1^n) = p \end{aligned}$$

and therefore

$$\lambda(P_n) = \max(p_1^n, p^-).$$

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

$$(\forall n \geq 0)(\lambda(P_n) \geq p^-) \implies \sum \lambda(P_n) \geq \sum p^-$$

The last series diverges and therefore the weak ergodicity of P_n follows from theorem 4.8 in (Seneta, 1973).

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 (9) it follows that p_m is a lower bound for p_1^n.

Proof of Proposition 2. 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

$$
\begin{aligned}
&K = X \times M_2^{p^-}\\
&X = \left\{ (p, 1-p) | 0 \leq p \leq 1 \right\},\\
&M_2^{p^-} = \left\{ \left[\begin{array}{cc} 1-p^- & p^- \\ p_{21} & 1-p_{21} \end{array} \right] \middle| \begin{array}{c} p_{21} = p(1-p^-) + (1-p)p_m(1-p^-) \\ = (1-p^-)(1-p_m)p + (1-p^-)p_m, \\ 0 \leq p_m, p^- \leq 1, (p, 1-p) \in X \end{array} \right\}
\end{aligned}
$$

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

$$
\begin{aligned}
S : K &\implies K,\\
S(p, P) &= (pP, P_{pP}),
\end{aligned}
$$

where

$$
\begin{aligned}
p &= (p, 1-p)\\
p &= \left[\begin{array}{cc} 1-p^- & p^- \\ p_{21} & 1-p_{21} \end{array} \right],\\
P_{pP} &= \left[\begin{array}{cc} 1-p^- & p^- \\ p_{21}^* & 1-p_{21}^* \end{array} \right],
\end{aligned}
$$

and

$$
\begin{aligned}
&pP = (p^*, 1-p^*) = (p(1-p^-) + (1-p)p_{21}, pp^- + (1-p)(1-p_{21})),\\
&p_{21} = p(1-p^-) + (1-p)(1-p^-)p_m\\
&\quad = (1-p_m)(1-p^-)p + p_m(1-p^-), \qquad (10)
\end{aligned}
$$

It follows that

$$
\begin{aligned}
p^* &= p(1-p^-) + (1-p)((1-p_m)(1-p^-)p + p_m(1-p^-))\\
&= (1-p^-)\left[p + (1-p_m)(1-p)p + (1-p)p_m\right]\\
&= (1-p_m)(1-p^-)p(2-p) + p_m(1-p^-)\\
&= (1-p_m)(1-p^-)p^* + p_m(1-p^-) \qquad (11)
\end{aligned}
$$

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 non-homogenous 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 $\epsilon > 0$ and fix an arbitrary point in $K, (q, Q)$ and chose another point (p, P), such that

$$\|(q, Q) - (p, P)\| \leq \delta$$

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 (10)

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

Together the above give

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

Now

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

Both terms in the above equation will be considered separately.

$$\begin{aligned}\|qQ - pP\|_x = 2|q^* - p^*| &= 2(1 - p_m)(1 - p^-)|q(2 - q) - p(2 - p)| \\ &= 2(1 - p_m)(1 - p^-)|2(q - p) - (q^2 - p^2)| \\ &= 2(1 - p_m)(1 - p^-)|2 - (q + p)||q - p| \\ &= (1 - p_m)(1 - p^-)|2 - (q + p)|\|Q - P\|_M\end{aligned}$$

Similarly

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

Finally it follows that

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

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

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

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

9. References

al-Rifaie, M. M. and Bishop, J. M. (2013). Stochastic diffusion search review. *Journal of Behavioral Robotics*, 4(3):155–173.

al-Rifaie, M. M., Bishop, J. M., and Caines, S. (2012). Creativity and autonomy in swarm intelligence systems. *Cognitive Computation*, 4(3):320–331.

al-Rifaie, M. M., Bishop, M. J., and Blackwell, T. (2011). An investigation into the merger of stochastic diffusion search and particle swarm optimisation. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, GECCO '11, pages 37–44, New York, NY, USA. ACM.

Aleksander, I. and Stonham, T. (1979). Guide to pattern recognition using random access memories. *Computers & Digital Techniques*, pages 29–40.

Back, T. (1996). *Evolutionary Algorithms in Theory and Practice.* Oxford University Press, Oxford, UK.

Bak, P., Tang, C., and Wiesenfeld, K. (1987). Self-organized criticality: an explanation of 1/f noise. *Physics Review Letters*, 59:381.

Beattie, P. and Bishop, J. (1998). Self-localisation in the senario autonomous wheelchair. *Journal of Intelligent and Robotic Systems*, 22:255–267.

Bishop, J. M. (1989). Stochastic searching networks. In *Proceedings of the First IEE Conference on Artificial Neural Networks*, pages 329–331, London, UK.

Bishop, J. M. and Torr, P. H. (1992). The stochastic search network. In Linggard, R., Myers, D., and Nightingale, C., editors, *Neural Networks for Images, Speech and Natural Language*, pages 370–387. Chapman & Hall, New York.

Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999). *Swarm intelligence: from natural to artificial systems.* Oxford University Press, Oxford, UK.

Colorni, A., Dorigo, M., and Maniezzo, V. (1991). Distributed optimisation by ant colonies. In *Proceedings of the 1st European Conference on Artificial Life.* MIT Press/Bradford Books.

de Meyer, K., Nasuto, S., and Bishop, J. (2006). Stochastic diffusion optimisation: the application of partial function evaluation and stochastic recruitment in swarm intelligence optimisation. In Abraham, A., Grosam, C., and Ramos, V., editors, *Stigmergic Optimization*, volume 31 of *Studies in Computational Intelligence*, pages 185–207. Springer Verlag, Berlin, Heidelberg, New York.

Dorigo, M. (1992). *Optimisation, Learning and Natural Algorithms*. PhD thesis, Politechnico di Milano-PM-AI & R Project, Milano, Italy.

Dorigo, M. (1994). Learning by probabilistic boolean networks. In *Proceedings of the World Congress on Computational Intelligence*, IEEE international Conference on Neural Networks, Orlando.

Doucet, A., de Freitas, J., and Gordon, N., editors (2000). *Sequential Monte Carlo Methods in Practice*. Springer-Verlag.

Ehrenfest, P. and Ehrenfest, T. (1907). Uber zwei bekannte einwande gegen das boltzmannsche h therem. *Physic Z*, 8:311.

Fischler, M. and Bolles, R. (1981). Random sample consensus: a paradigm for model fitting with application to image analysis and cartography. *Communications of the ACM*, 24:381–395.

Gilks, W., Richardson, S., and Spiegelhalter, D., editors (1996). *Markov Chains Monte Carlo in Practice*. Chapman & Hall, London.

Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Boston, MA.

Grech-Cini, E. (1995). *Motion tracking in video films*. PhD thesis, University of Reading, Reading, UK.

Gusfield, D. (1997). *Algorithms on Strings, Trees, and Sequences: computer science and computational biology*. Cambridge University Press, Cambridge UK.

Holland, J. (1975). *Adaptation in natural and artificial systems*. University of Michigan Press, Ann Arbor, MI.

Johnson, N. and Kotz, S. (1969). *Distributions in statistics: discrete distributions*. Wiley, New York.

Kennedy, J. F., Eberhart, R. C., and Shi, Y. (2001). *Swarm intelligence*. Morgan Kaufmann Publishers, San Francisco.

Kirkpatrick, S., Gelatt, C., and Vecchi, M. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680.

Li, S. and J., Z. (2012). Cellular sds algorithm for the rectilinear steiner minimum tree. In *2012 Third International Conference onDigital Manufacturing and Automation (ICDMA)*, pages 272–276. IEEE.

Mikhailov, A. (1993). Collective dynamics in models of communicating populations. In Haken, H. and Mikhailov, A., editors, *Interdisciplinary Approaches to Nonlinear Complex Systems*, volume 62 of *Series in Synergetics*. Springer-Verlag, Berlin.

Moglich, M., Maschwitz, U., and Holldobler, B. (1974). Tandem calling: A new kind of signal in ant communication. *Science*, 186(4168):1046–1047.

Myatt, D., Bishop, J., and Nasuto, S. (2004). Minimum stable convergence criteria for stochastic diffusion search. *Electronics Letters*, 40(2):112–113.

Nasuto, S. and Bishop, J. (1999). Convergence analysis of stochastic diffusion search. *Parallel Algorithms and Applications*, 14(2):89–107.

Nasuto, S., Bishop, J., and de Meyer, K. (2009). Communicating neurons: A connectionist spiking neuron implementation of stochastic diffusion search. *Neurocomputing*, 72(4-6):704–712.

Nasuto, S., Bishop, J., and Lauria, S. (1998). Time complexity analysis of stochastic diffusion search. In *Neural Computation '98 - Proceedings of the International ICSC/IFAC Symposium on Neural Computation*, Vienna, Austria.

Navarro, G. (1998). *Approximate Text Searching*. PhD thesis, University of Chile.

Saaty, T. (1981). *Modern nonlinear equations*. Dover Publications, New York.

Seneta, E. (1973). *Nonnegative Matrices*. George Allen & Unwin Ltd., London.

van Laarhoven, P. and Aarts, E. (1987). *Simulated Annealing: theory and applications*. Kluwer, Dordrecht.

van Leeuwen, J. (1990). *Handbook of Theoretical Computer Science: Algorithms and Complexity*. Elsevier, Amsterdam.

Whitaker, R. and Hurley, S. (2002). An agent based approach to site selection for wireless networks. In *Proceedings of the 2002 ACM symposium on Applied computing*, pages 574–577, Madrid, Spain. ACM Press.

Whittle, P. (1986). *Systems in Stochastic Equilibrium*. John Wiley & Sons, London.

Williams, H. and Bishop, J. (2014). Stochastic diffusion search: a comparison of swarm intelligence parameter estimation algorithms with ransac. *Algorithms*, 7:206–228.

Young, P., Kaniovski, Y., and Kryazhimskii, A. (2000). Adaptive dynamics in games played by heterogenous populations. *Games and Economic Behavior*, 31:50–96.

Zhigljavsky, A. (1991). *Theory of Global Random Search*. Kluver Academic Publishers, Dordrecht.