Convergence Analysis of Stochastic Diffusion Search

S. Nasuto [1]& M. Bishop,
Department of Cybernetics,
University of Reading,
Reading RG6 2AE, UK

Abstract

In this paper we present a connectionist searching technique- the Stochastic Diffusion Search (SDS), capable of rapidly locating a specified pattern in a noisy search space. In operation SDS finds the position of the prespecified pattern or if it does not exist - its best instantiation in the search space. This is achieved via parallel exploration of the whole search space by an ensemble of agents searching in a competitive co-operative manner. We prove mathematically the convergence of stochastic diffusion search. SDS converges to a statistical equilibrium when it locates the best instantiation of the object in the search space. Experiments presented in this paper indicate the high robustness of SDS and show good scalability with problem size. The convergence characteristics of SDS makes it a fully adaptive algorithm and suggests applications in dynamically changing environments.

[1] **sjn@cyber.rdg.ac.uk**

1. Introduction

In Connex (a privately funded connectionist research programme by British Telecomm) several neural architectures have been proposed as possible solutions for locating facial features in images [1, 2, 3, 4]. Typically they use a standard multilayer perceptron architecture trained with the classical steepest descent algorithm. Due to slow learning on the raw data several pre-processing techniques were used in order to reduce the learning time. Hand [1] uses several representations of images with different resolutions. Waite [2] applies principal component representation of images, thus reducing the dimensionality of the input data. Lisboa [3] uses Gabor functions for image representations. Finally Vincent [4] uses heuristics to speed up the learning. In all cases the network scans sequentially over the image with a predefined window. Thus when locating the desired feature in these techniques time and reliability are influenced by the pre-processing and scanning of images. They also are prone to converging on false positives corresponding to local minima in the error surface used by the steepest descent algorithm.

Stochastic Diffusion Search (SDS) was introduced in [5] as a part of the Stochastic Diffusion Network (SDN) for solving visual search tasks. The SDN was used in the visual domain for facial feature location [5] and in on line, real time locating and tracking of lips images in video films [6]. In both cases the SDN exhibited a high level of reliability and fast convergence in spite of imperfections in the test images. SDN did not use any pre-processing of the data for locating target features. The Stochastic Diffusion Network as shown in Figure 1 is a hybrid system comprising of a set of n-tuple neurons [7], guided by Stochastic Diffusion Search towards potentially correct positions in the search space.

In this article we will present a modified version of SDS derived from the algorithm described in [5]. The modification enables SDS to act as a standalone technique able to locate any prespecified object within a given space.

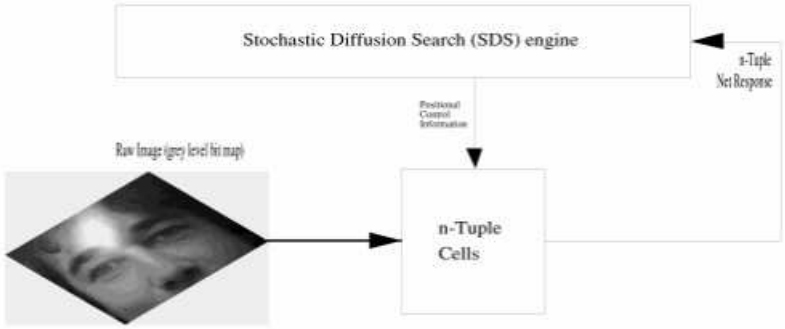

FIGURE 1 Block diagram of SDN

SDS can be placed in the family of directed random search algorithms defined by Davies in [8] to include Simulated Annealing [9] and Genetic Algorithms [10].
Simulated Annealing is a general purpose global optimisation technique for very large combinatorial problems. Most of the development to date has been in the area of 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) are stochastically adjusted, with adjustments that worsen the system performance by a factor (z) being accepted with a probability defined by the Boltzmann distribution:

$$P\{accept\} = e^{\frac{z}{KT}},$$

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

Genetic Algorithms are search algorithms utilising the mechanics of Darwinian natural selection and genetics. The Parameter Set is encoded in a manner isomorphic to genetic encoding and a population of encoded parameter sets is evaluated on the optimisation problem to define a fitness measure for each of the parameter sets. New parameters sets are probabilistically derived from the old in proportion to this fitness measure. Each such cycle of evaluation and derivation is defined as one cycle of evolution. Over a series of such cycles, the parameter sets self adjust to define robust and efficient solution(s) to the problem.

Stochastic Diffusion Search is a connectionist, probabilistic model comprising of a number of processing elements called agents. All agents synchronously perform their tasks and exhibit competitive- cooperative behaviour.
Unlike most connectionist models the agents do not process information by summing up inputs and applying to the result a non-linear transfer function. Agents in SDS do not have fixed connections, so information processing is not carried out by adjusting weights according to some learning rule. Instead agents can communicate with each other and the information processing emerges as a collective property arising from the ability of individual agents to selectively choose other agents for this information transmission.
Most connectionist models provide a solution by converging to a certain point in weight space. Once they have converged, their performance will decrease if the problem is not static in time but changes dynamically. Thus after a sufficient amount of time such a network has to be retrained to produce consistently reliable solutions. Therefore, in contrast to widespread opinion, these models are not adaptive in a true sense of this word - they do not possess the capability of detecting dynamic changes in the input and adapting to them to produce a new solution.
SDS by contrast is capable of continuous exploration of the search space even after finding the optimal solution. Therefore effectively it can adapt to changes in the environment and find the new best solution. This claim is empirically supported in [6] where SDS was used for tracking in real time video films. Successive video frames can be regarded as a dynamic environment where the optimal solution changes over time. In this article we will explain how this adaptability is achieved by SDS.

We claim that SDS on its own is an interesting connectionist algorithm showing robustness towards noise and fast performance. In this article we will give a full mathematical model of SDS and will discuss its convergence. It will be shown that if the target exists in the search space, all agents will eventually converge to its position. We will also discuss how to extend the notion of the convergence of the algorithm in the case when there is no ideal instantiation of the target in the search space and we will prove that convergence also occurs in this case. Finally we will illustrate the behaviour of SDS on a simple text string search task which nevertheless is easily extendible to more interesting problems. From the presented simulations we can draw some preliminary conclusion concerning the time complexity of SDS.

2. Stochastic Diffusion Search

In this section we will introduce SDS and explain in detail the fundamental mechanism underlying its performance.
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.

The space and the object are defined in terms of Atomic Data Units (ADU's) which constitute the set of basic features. Each object in the search space and the target object are described in terms of ADU's and cannot contain any other features.
ADU's can be thought of as single pixels 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 prespecified neighbourhood).

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 coordinates of predefined 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 computational resources dynamicaly 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 access to the search space and to the description of the target object.

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 Brittannica, then agents will point to possible positions of the first 'A'. We assume that the word ABRACADABRA does occur in Encyclopaedia Brittannica). They are also initially all set as inactive.
Each of the agents then independently performs a probabilistic check of the information at the reference point by comparing a random ADU from the target object (e.g. one of the agents could have chosen a letter 'R' from ABRACADABRA) with a corresponding ADU 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 (activation on nontarget 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 (failing to activate on the best match to the target in the case the ideal instantiation of the target does not exist in the search space -e.g. if 'ABRACADABRA' was missspelt). In this way Stochastic Diffusion Search can escape from local minima corresponding to objects partially matching the description of the target.

Subsequently in the diffusion phase each inactive agent selects, at random, another agent with which to 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 (in our example 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 state is achieved. The halting criterion used in [5] monitored the maximal number of agents pointing to the same position in the search space (in our example, 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 cooperative behaviour. In order to establish the convergence of the Stochastic Diffusion Search we will introduce in the next section a mathematical model of SDS based on Markov Chain theory [11].

3. Model.

In the most general case, stochastic diffusion search is supposed to locate the target or if it does not exist in the search space its best instantiation. Therefore from now on we will refer to the object sought by SDS as the target.
Let the search space size be N (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 suboptimal 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 M agents. The state of the search in the n^{th} 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 nonactive agents will be equal to the difference between the total number of agents and the 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 we have finite number of discrete states each characterised by the pair of two natural numbers. Stochastic Diffusion Search changes its state in a random manner and the possible future evolution of the 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 we will construct the transition matrix.

Let the state of the search in the n^{th} step, denoted X_n, be specified by a pair of integers *(a,w),* where *a* denotes a number of active agents pointing towards the target and *w* - number of active agents pointing towards false positives.
If in the n^{th} 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.

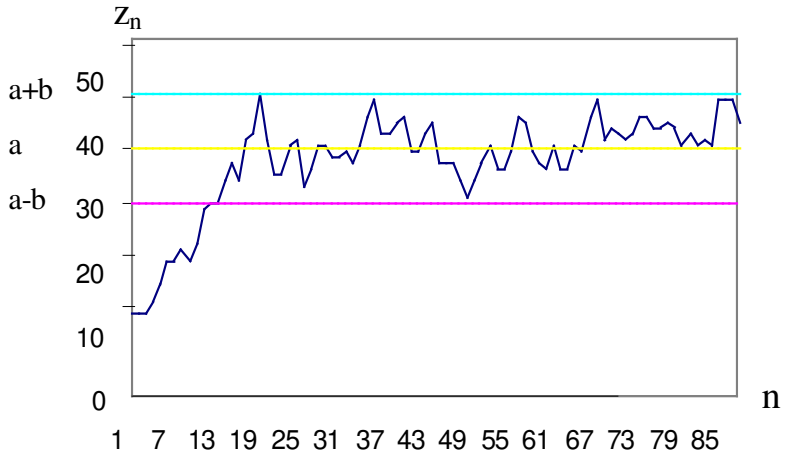

Figure 2. Hypothetical evolution of stochastic search depicting stabilisation as defined by the strong halting criterion. *n* denotes the number of iterations and z_n the maximal number of active agents pointing to the same position in the search space.

The one step evolution of the nonactive agent is determined first by the outcome of the diffusion phase and then by the testing phase. We will describe here one of its possible evolutions. During the diffusion phase a nonactive agent will choose an active agent pointing towards the target with probability *a/M* and then will remain active with probability $1\text{-}p^-$. The other possibilities follow in an analogous way and are best summarised in Figure 3.
It is apparent that transition from one state to another can take place in many different ways depending on the performance of all agents (e.g. number of nonactive 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 4). The exact formula is given below:

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

where

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

$$Bin(k_1, p^+) = \binom{b}{k_1} (p^+)^{k1} (1 - p^+)^{b-k1},$$

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

and

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

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

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

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

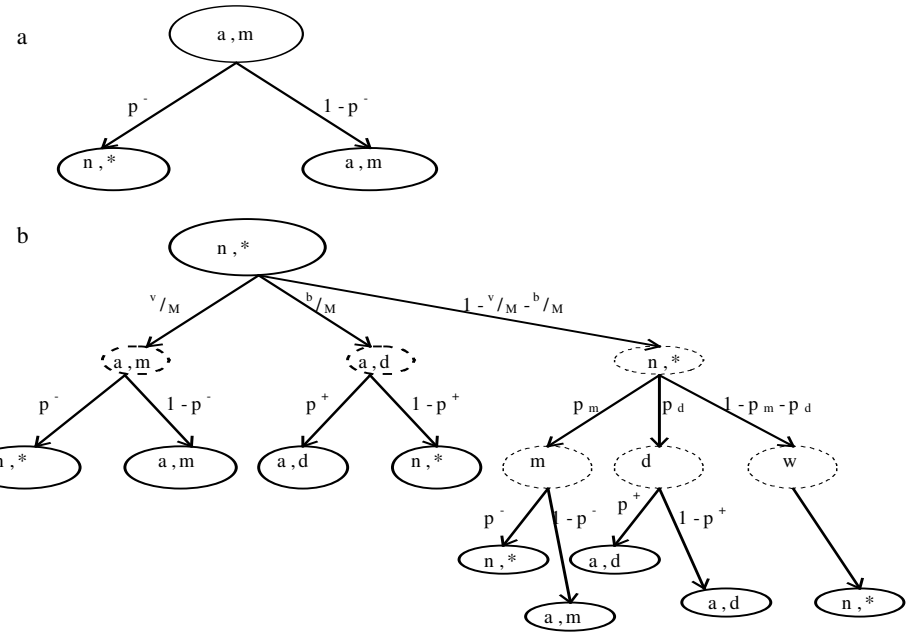

Figure 3. One step evolutions of an active agent pointing to the correct position (a) and of nonactive agent (b). Probabilities of changing states are given next to appropriate arrows. Intermediate steps are marked by dashed ovals. One step evolution of an active agent pointing to a disturbance is analogous to (a).

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 nonactive. The term $Mult(k_1, k_2, r, a, v, b)$ expresses the probability of $r-k_2$ out of $M-v-b$ inactive agents starting

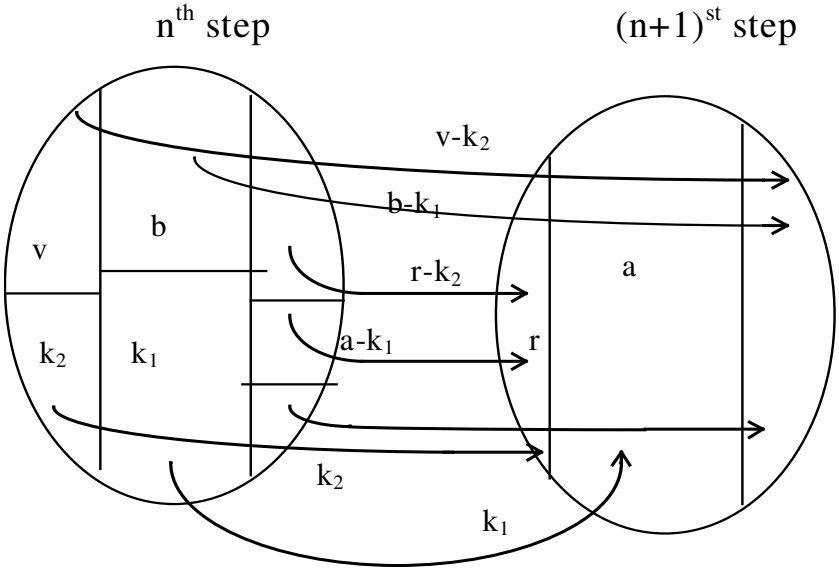

Figure 4. Diagrammatic illustration of one step evolution of the Stochastic Diffusion Search. In the n^{th} step there are v active agents pointing to the correct solution and b active agents pointing to the disturbances. In the $(n+1)^{st}$ iteration these numbers change to r and a respectively.

to point towards the target and passing the test phase, $a\text{-}k_1$ out of $M\text{-}v\text{-}b\text{-}r\text{+}k_2$ 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 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: $\sum_{s\in S} f_n^s \leq M$, where M 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 [5], the definition of convergence of stochastic diffusion search has the following formulation:

Definition 1. Strong halting criterion.
We say that stochastic diffusion search has reached an equilibrium, if

$$\exists_{a,b>0}\,(2b < M \wedge b + a \leq M \wedge a - b \geq 0)\,\exists_{n_0}\,\forall_{n>n_0}\left(|z_n - a| < b\right),$$

and the solution is the position pointed at by z_n.
Thus stochastic diffusion 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 cooperation process will lead to the allocation of most agents to the best fit position.

Note also, that the above definition does not require convergence of the process to a fixed point. 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 we would ask only, that $0 \leq z_n \leq M$.
Figure 2 shows a hypothetical evolution of the stochastic diffusion search stabilising in the sense of the strong halting criterion. From the graph we see that the search stabilises around the value a=40 and fluctuates within the band $2b$=20.
It will be shown that in the case of ideal instantiation of the target in the search space these two parameters do not play a critical role.
In the opposite case we are faced with the difficult problem. 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 this is not known in advance in the most general case. The possible solution is for the user of the SDS 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.

4. Convergence of Stochastic Diffusion Search.

We will analyse the convergence of SDS in two separate cases. First we will concentrate on the case when there exist the ideal instantiation of the target in the search space.
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 fail the test).
In what follows we will use the notation introduced in section 2.

Let the position of the model in the search space be denoted as s^m. Recall that in our Markov chain model of stochastic diffusion the presence of the object in the search space is equivalent to setting p^- to zero.
Proposition 1. *If $p^-=0$, then*

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

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

Proof. From transition probability matrix and from $p^-=0$ it follows that,

$$P\{(M,0)|(M,0)\} = \left(1 - p_{ab} - p_{af}\right)^{M-M} = 1$$

and $0 \leq P\{(v,b)|(v,b)\} < 1$, *i.e.* the only diagonal element equal to unity is *(M,0)*.
This means that our model is an absorbing Markov chain and *(M,0)* is the only absorbing state. Stochastic search will therefore eventually reach the state *(M,0)* in finite time and then will stay in this state forever. All other states are transient. The rate of convergence is geometric for some constant c, $0<c<1$.
■

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 we 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.

Proposition 2. Given $p^- \neq 0$ the strong convergence criterion does not hold in the stochastic

diffusion search.

Proof.
We will prove the above assertion by showing that a less restrictive property, of which strong convergence criterion is a special subclass, is not fulfilled either. We will show by contradiction, that

$$\exists_{a,b>0}(2b < M \wedge b + a \leq M \wedge a - b \geq 0) \lim_{n\to+\infty} P\{|z_n - a| < b\} = 1$$

is not true.
Suppose the above assertion holds. It is equivalent to

$$\exists_{a,b>0}(2b < M \wedge b + a \leq M \wedge a - b \geq 0) \lim_{n\to+\infty} P\{|z_n - a| \geq b\} = 0 \cdot$$

Let $p^- \neq 0$. In the case of $p^+ = p_d = 0$ the ephemeral states with a nonzero amount of noise are excluded from consideration. From the probability transition matrix it follows that for any state $(i,j) \in S$,

$$P\{S_{n+1}=(i,j) | S_n=(0,0)\} > 0$$

and

$$P\{ S_{n+1}=(0,0) | S_n=(i,j)\} > 0,$$

i.e. the first row and first column of the transition probability matrix **P** are strictly positive. It follows that any entry of $\mathbf{P}^2$ is positive, hence **P** is primitive. From the Perron-Frobenius theorem it follows, that there exists over states in S a limit probability distribution p_∞, such that $p_\infty > 0$. This implies that in a steady state all of the states occur with probability strictly greater than one, i.e. infinitely often and $z_n = max_{s\in S}(f_n^s)$ takes all possible values from the set $\{0,..,M\}$ with positive probability. This contradicts our assumption. ■

From the above proof it follows, that in the case of $p^- \neq 0$ the model of stochastic diffusion search is an ergodic Markov chain [11]. Therefore it is easy to see that stochastic diffusion fulfils another, weaker convergence property, namely;

Proposition 3. Given $p^- \neq 0$, stochastic diffusion search converges in a weak sense, i.e.

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

Proof. Follows immediately from the ergodicity property of stochastic search and observation that z_n is a random variable defined on the probability space $(S, \sigma(S), p_n)$, where $\sigma(S)$ is a σ-algebra of all subsets of S. ■

The above characterisations show that in the most general case Stochastic Diffusion 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 suboptimal 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.

Numerical Simulations

Simulations of SDS were performed in order to assess the robustness of SDS and rate of convergence for different search space sizes. The task was to locate in an input string of digits a given string or if it did not exist - its best instantiation. Experiments were aimed at characterising the relation between the convergence time of SDS and the search space size for varying amounts of disturbances (corrupted versions of the target). They consisted of 50 runs of SDS, for search space sizes in the range 150..300. In all tests the target was imperfectly instantiated within the search space, with errors varying from 30% to 80% of the search space size. The object to be found, the best instantiation of the target, had a 70% chance of being accepted as the correct solution (i.e. 30% out of a total number of defining digits were replaced by randomly generated ones). Noise, or disturbances were realised as partial copies of the target and were inserted into the search space in random positions. They could cause false positive recognition with a given probability p_d, where p_d is a controlled parameter in the experiment.

The experiments were carried in two groups of three experiments each. In the first group of experiments the number of disturbances in the search space was 30%, 50% and 80% of the search space size respectively. The probability of accepting a disturbance as a false positive solution in the first group was fixed to $p^+ = 50\%$ and the threshold value, $\boldsymbol{a}$**,** for recognising the equilibrium state was set to 50% of all agents. In the second group the probability of accepting the disturbance as the solution was increased to $p^+ = 60\%$ and the threshold value a was decreased to 20% of all agents. This is because we expect the equilibrium necessary for the termination of the SDS to be lower in the search space with noise increased. All graphs were obtained from averaging the convergence time over all 50 runs for each experimental condition. Graphs also show the result of performing linear regression on the raw experimental data.

The results of experiments from the first group are shown in Figure 5 and from the second group in Figure 6. From all the graphs it follows that in all cases there is a slow, approximately linear growth of the convergence time with the search space size regardless of the amount of noise or a probability of its acceptance as a correct solution. In the worst case the correlation coefficient ρ between the linear regression fit and data averages is equal to 0.78, in most cases it fluctuates around 0.9.

From Figure 5 it follows that the 60% increase of the amount of noise in the search space (from 30% to 50% of the search space size and from 50% to 80% of the search space size) causes an increase in the convergence time of no more than 15%. However, the overall linear relationship of convergence time to search space size seems to be preserved.

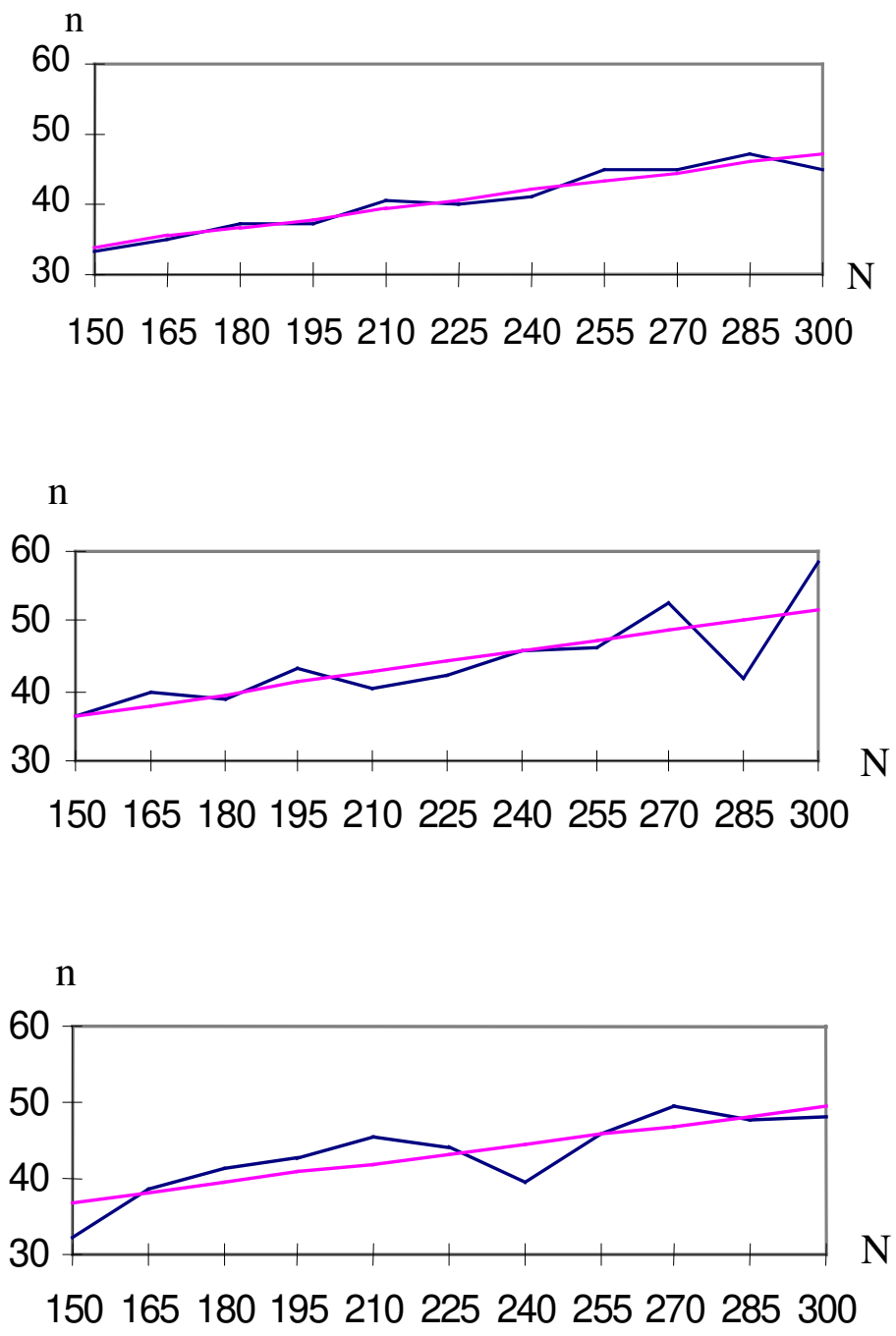

Figure 5. Convergence time, *n*, versus search space size *N*. The amounts of noise are (a) 30%, (b) 50% and (c) 80% respectively and the probability of false positive is 50%. Probability of the false negative is 30%. Each graph contains a plot of average convergence times over 50 runs and a linear regression fit. Correlation coefficients from linear regression are (a) 0.967793, (b) 0.806027 and (c) 0.838003 repectively.

Figure 6 shows a similar relationship between level of noise and convergence time even though each noisy element is 10% more like the target.

However in this case the dependence of the convergence time on the amount of noise is not so

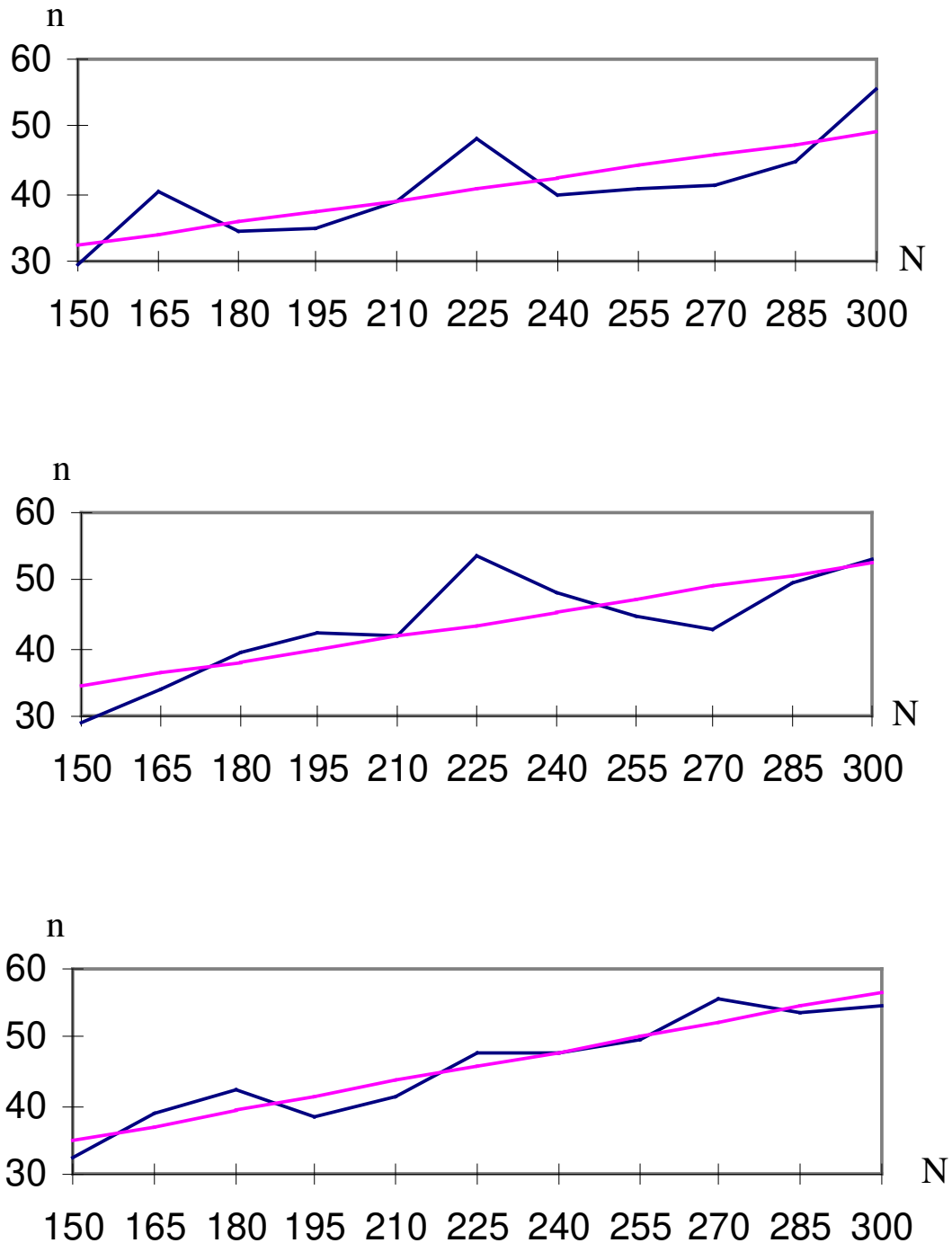

Figure 6. Convergence time, *n*, versus search space size *N*. The amounts of noise are (a) 30%, (b) 50% and (c) 80% respectively and the probability of false positive is 60%. Probability of the false negative is 30%. Each graph contains a plot of average convergence times over 50 runs and a linear regression fit. Correlation coefficients from the linear regression are (a) 0.781109, (b) 0.809235 and (c) 0.952315 respectively.

clear as there is increased variance in our results. The dependence becomes more clear if one compares graphs 6a and 6b suggesting that at least some of the variation may be caused by

averaging over a relatively small sample.

If one compares graphs from both groups with corresponding amounts of noise, one can see that decreasing the discrimination between noise and target leads to a very high increase in the convergence time. Therefore we conclude from our experiments that the convergence time of SDS is much more dependent on the discrimination between the best instantiation of the target and the background noise than on the total amount of noise in the search space.

As the difference in the probability of accepting a disturbance compared to the probability of accepting the best instantiation is reduced, it is harder for SDS to maintain an active population of agents pointing to the best instantiation of the target.
The probability of a misclassification is increased as the difference between the probability of accepting the best instantiation and the probability of accepting a disturbance is reduced. However, even in the worst case of a search space consisting of 80% of noise, with a probability of accepting each at 60%, SDS successfully converged in over 95% of cases.

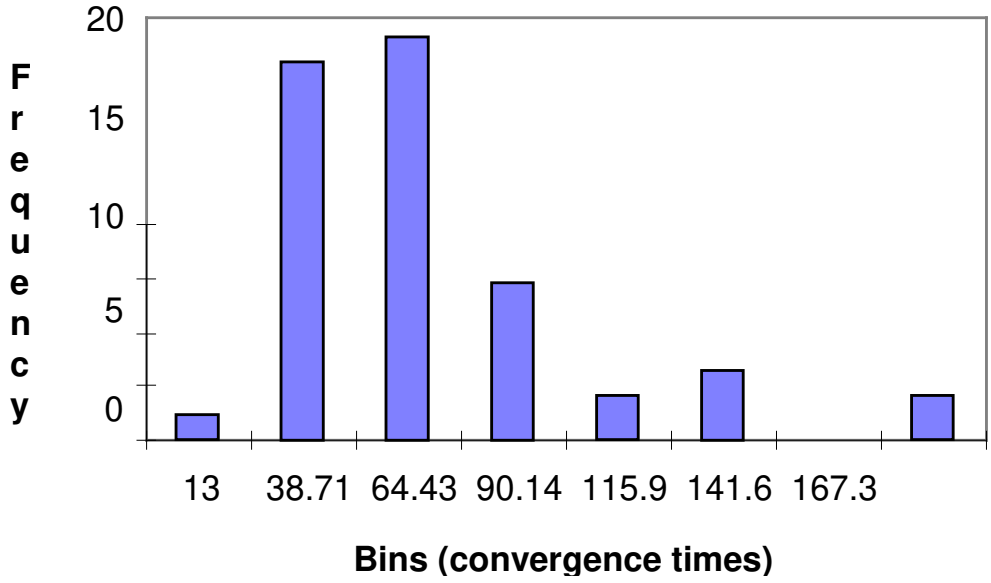

Figure 7. Histogram of the convergence time distribution for the maximal search space size, 80% of disturbances, 60% probability of false positive acceptance.

In all cases we observed a similar growth pattern of the standard deviation over trials to that of average convergence time, i.e. it grows approximately linearly with the search space time. This can be accounted for by the increase of the convergence time discrepancy in two extreme situations. First, when SDS is initialised with some agents pointing to the correct position in the search space. In this case the convergence time is very short, much shorter than the average. On the other hand in the situation when no agent point to the correct position at the beginning it may take long time before the first agent will find the best solution. Indeed from Figure 7, showing a typical result of our experiments, the histogram is calculated for the maximal search space size with 80% of noise and p^+=60%. It follows that the distribution of convergence times is skewed, showing a long tail above the average convergence time, but an actual concentration of data points around the average.

Conclusion

In this paper we presented the Stochastic Diffusion Search 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 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. 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 Stochastic Diffusion Search in solving such problems [6] follows from its adaptability, the latter being the result of the weak convergence of SDS.
From our experiments it follows that the SDS is relatively robust to the amount of noise in the search space with respect to both accuracy and convergence time. It is however sensitive to the relative discrimination between the best model instantiation and background noise. Future research will concentrate on analysing this behaviour, specifically we hope to obtain a reliable time complexity characteristic of the algorithm. We will also investigate possible ways to reduce the growth of convergence time standard deviation with the search space size.

Acknowledgments

Authors would like to thank Dr P. MacOwan and Dr B. Yu of Department of Cybernetics for proofreading of the manuscript and valuable comments.

References.

[1] Hand C C and et al 1992 A neural network feature detector using multi-resolution pyramid *Neural Networks for Images, Speech and Natural Language* ed Linggard R and Myers D J and Nightingale C (New York: Chapman & Hall)
[2]. Waite J B 1992 Training multilayer perceptrons for facial feature location: a case study *Neural Networks for Images, Speech and Natural Language* ed Linggard R and Myers D J and Nightingale C (New York: Chapman & Hall)
[3] Lisboa P J G 1992 Image classification using Gabor representation with a neural net. *Neural Networks for Images, Speech and Natural Language* ed Linggard R and Myers D J and Nightingale C (New York: Chapman & Hall)
[4]. Vincent J M and et al 1992 Image feature location in multi-resolution images using a hierarchy of multilayer perceptrons. *Neural Networks for Images, Speech and Natural Language* ed Linggard R and Myers D J and Nightingale C (New York: Chapman & Hall)
[6] Grech-Cini E. 1995 Locating Facial Features. PhD Thesis, University of Reading

[7] Aleksander I and Stonham T J 1979 Guide to Pattern Recognition using Random Access Memories *Computers and Digital Techniques* **2**(1) pp 29-40
[8] Davies L ed 1987 Genetic Algorithms and Simulated Annealing
(London: Pittman)
[9] Kirkpatrick S, Gelatt C D and Vecchi M P 1983 Optimisation by
Simulated Annealing Science **220**(4598) pp 671-680
[10] Holland J H 1975 Adaptation in Natural and Artificial Systems (Ann
Arbour: The University of Michigan Press)
[11] Kemeny J G and Snell J L 1976 *Finite Markov Chains* (New York: Springer-Verlag)