Time Complexity Analysis of the Stochastic Diffusion Search

[†]Slawomir J.Nasuto, [‡]Mark J.Bishop and [$]Stanislao Lauria
Department of Cybernetics,
Neural Networks Research Group,
University of Reading
Whiteknights,
RG6 2AE, Reading,
U.K.
[†]sjn@cyber.rdg.ac.uk
[‡]J.M.Bishop@reading.ac.uk
[$]sl@cyber.rdg.ac.uk

Abstract. The Stochastic Diffusion Search algorithm -an integral part of Stochastic Search Networks is investigated. Stochastic Diffusion Search is an alternative solution for invariant pattern recognition and focus of attention. It has been shown that the algorithm can be modelled as an ergodic, finite state Markov Chain under some non-restrictive assumptions. Sub-linear time complexity for some settings of parameters has been formulated and proved. Some properties of the algorithm are then characterised and numerical examples illustrating some features of the algorithm are presented.

Keywords: Stochastic Diffusion, invariant pattern recognition, focus of attention, Markov Chains modelling, time complexity analysis.

1. Introduction.

A basic problem in pattern analysis is that of pattern matching. It is an area of great importance and interest in its own right but also due to its numerous applications, including: text processing, image analysis, symbolic computation and automated theorem proving amongst others. Numerous pattern matching techniques have been developed for different contexts. The majority of these use specialised heuristics, enhancing performance in their particular domain but restricting applicability over other recognition tasks.

A particularly important form of pattern matching from a practical point of view is inexact matching. It occurs when a pattern does not exist in the search space in its original form but is distorted by some means. An example of such situation appears in recognition tasks in which an object to be recognised has undergone various transforms in the search space. Such transforms may be of geometric nature, e.g. translations, rotations or scale changes or changes in the lightning conditions of the scene etc. Equally important are changes of random nature like distortions due to noise, unreliability of pattern storage, errors in pattern transmission channels etc. Thus in this case one is posed with inexact matching task in which only some of features characterising the given object may remain invariant.

Several techniques have addressed the problem of inexact matching. Li [10] approaches the problem by using Marr's [11] technique of object-centred representation for recognition. In this approach an object is represented in terms of predefined relations between its characteristic features. The relational structure obtained in this way is invariant to prespecified transforms and matching is formulated and solved as an optimisation problem by a gradient like method. This can however cause problems of convergence to local minima's. It also can cope only with fixed transforms of systematic nature. Similar approach has been proposed by Shapiro and Haralick [15] where authors define the structural description of an object and perform matching by means of a tree search.

Pattern recognition has also been investigated in neural network community in the context of modelling visual process. One of the fundamental problems of early connectionist models was stimulus invariance. There

has been no clear normalisation mechanism that would constitute an integral part of a model and would enable pattern recognition under different transformations. One solution proposed to overcome this problem was Hinton Mapping [7]. In his paper Hinton proposed a PDP model capable of recognising objects regardless of transformations by imposing a canonical frame of reference on objects. His model consists of object-based, canonical feature detectors and retinocentric ones. It is assumed that retinocentric feature detectors are rich enough to be able to fit an object regardless of its transformations. A mapping from retina- based features to object-based features is performed via mutual interaction between the two classes of detectors and a fixed set of so called mapping units, representing all possible transformations. This leads to a large number of mapping units. Another drawback stated in [14] is that Hinton's mapping allows only for serial attention modelling, whereas a system allowing a parallel focus of attention could explore simultaneous, mutual constraints between patterns that could model contextual dependencies. Also significant nonsystematic or random deformations of object can pose difficulty for this system.

The Stochastic Diffusion Network (SDN) [5] has been proposed as a connectionist pattern recognition technique able to classify patterns invariant of transformation within a search space. It can perform both serial as well as parallel focusing of attention and is robust to noise distortions and partial occlusions of the object.
The Stochastic Diffusion Network consists of a Stochastic Diffusion Search (SDS) engine [4], directing n-tuple RAMs [1] into a search space.
Transformation invariance of SDN is maintained partially due to Stochastic Diffusion Search capabilities of discovering partial matches and partially due to generalisation capabilities of n-tuple RAMs used for object recognition. In [5] an application[1] of the Stochastic Diffusion Network to invariant pattern recognition problems is described. The task for the SDN was to locate an eye feature in images of human faces. The algorithm has been able to locate all eye features accurately in all images from the training set and achieved over 60 % accuracy on the testing set. Transformation invariance of SDN is discussed also in [6].

In SDS the search is performed by competing co-operative process between simple agents which pass potential position of the target in the search space to n-tuples and diffuse information across other agents in order to explore the best intermediate solutions. It is from this competition-co-operation process that the parallel focus of attention of the Stochastic Diffusion Search emerges.

Unlike most of artificial neural networks, in Stochastic Diffusion Search agents are not connected by weighted links. They also do not calculate their response by applying a non-linear transfer function to input. Thus SDS do not perform calculations according to some learning rule defining a path in a weight space - a form of computation prevalent in standard connectionist models.
Instead information processing capability of SDS comes from profound shift in the paradigm. Classical connectionist models assume that the fundamental mode of information processing is computation. In this approach communication between computing components plays a very minor, passive role. SDS exemplifies the other extreme of possible spectrum and replaces computation by communication. Agents can communicate with each other and information processing appears as an emergent, collective property. It follows from their ability to selectively choose other agents for information transmission.
By its very nature SDS avoids pitfalls of classical connectionist models when faced with non stationary problems. Most of them need to be retrained when problem to be solved changes over time.
By contrast, in the Stochastic Diffusion Search, continuous exploration of the search space by some of the agents even after finding the best solution at a given time instant enables SDS to accommodate to environment dynamics. Thus SDS is a truly adaptive algorithm.

A Markov Chain model of Stochastic Diffusion Search has been proposed in [12]. The model parameters described a broad range of search space characteristics in terms of noise and best instantiation of the data model. Due to a general formulation, the model of SDS was generic; it was not restricted to specific features of a particular search space. In [12] convergence properties of SDS has been analysed. In particular it was shown that SDS converges with probability one, if the ideal solution exists in the search space. In the case when only partial, best fit solution exists in the search space, a weak convergence has been formulated and proved. Its practical implications are high ratio of correct solutions over wide range of parameters characterising the search conditions and adaptive character of the algorithm.

[1]This work was performed as a part of the Telecom CONNEX project.

In this paper further properties of SDS will be investigated. In particular we will concentrate on time complexity of Stochastic Diffusion Search. It will be shown that Stochastic Diffusion Search is sublinear in time in presence of no noise and the perfect match. This performance is achieved without using heuristic strategies, in contrast to best one- and two-dimensional string searching algorithms or their extensions to tree matching [8, 16, 17], which achieve time linearity. The use of application specific heuristic would yield even faster convergence. Simulations performed suggest that the time characterisation approximately holds in the presence of disturbances in the search space. The Stochastic Diffusion Search algorithm is highly parallel in nature and promises to be a good model of attentive perception.

The paper is organised as follows. In section 2 we give a concise description of Stochastic Diffusion Search. Section 3 describes the Markov Chain model and section 4 discusses the time complexity of the search. Section 5 gives some numerical examples to illustrate the accuracy of the model of Stochastic Diffusion and shows the performance of the search using a variety of parameters. The final section discusses the model complexity and the consequent difficulties in detailed analysis and suggests topics for further research.

2. Stochastic Diffusion Search.

The purpose of the Stochastic Diffusion Search is to locate a predefined data pattern within a given search space or, if it does not occur, its best instantiation. Other possible partial instantiations of the data model are treated as noise or disturbances influencing the search.
To understand better settings in which Stochastic Diffusion Search operates we will use the following example. Let the search space be a picture of a crowded street. We want to find and locate a particular person in the picture. We will define a position of this person if we can point to an arbitrarily chosen, predefined reference point (this construction is unnatural for humans as we intuitively understand what we mean by a position of a person without referring to the location of his/her centre of gravity, for instance; however it seems that if we want to implement our technique in an artificial system, we have to assign to such ambiguous definition precise, unique meaning. The picture of the person made in optimal conditions will be our data model. However in the crowd the person can appear partially occluded, rotated with respect to the position on the model picture, may not wear glasses etc. It means that the image of the person from our picture does not match perfectly with that in the scene. Moreover there may be some people in the crowd with similar body constitutions, similarly clothed etc. Due to their potential similarity to a given person they constitute partial matches. However, the task of recognising the person in such a situation is relatively easy for adults. In spite of this, it may constitute a difficult problem for an artificial visual system.
The situation described above can be generalised to other potential settings- the search can be performed in n-dimensional space, where objects are represented by collections of points representing microfeatures- basic building blocks of the space. We can also imagine that the search space is represented by a graph and we are interested in locating a pattern represented as another subgraph.

The search space as well as data model are defined by 'atomic data units' -ADU's, which represent the basic microfeatures from which they are both comprised. Thus, features of objects from a search space can be effectively defined in terms of appropriate ADU's and their relationships. In the above example the search space is a bit map and ADU's can be defined as single pixel intensities; in case of a search space being a graph, ADU's can be thought of as nodes and edges. The locations of ADU's common to the object in the search space thus constitute partial solutions to the search.

Stochastic Diffusion Search is performed in parallel by a prespecified number of elements called agents. An agent is characterised by a pointer to a position in the search space and by a binary variable called activity. It assumes value 1, if agent points to potentially correct position within a search space (agent is active), otherwise it is equal to 0 (agent is inactive).
The search can be considered as a competitive co-operative process in which all agents independently seek for partial solutions. The latter, once found, compete to allocate other agents to their positions (position of a potential solution is understood here as co-ordinates of predefined reference point in the target's description).
In this way competition transforms smoothly into co-operation, when more and more agents are attracted to explore the potential fit to the data model. This competition for co-operation assures that all potential positions of the object will be examined independently with the most promising one over a number of iterations attracting most of the computational resources. Thus the correct position of the best fit to the

data model will emerge from independent, parallel exploitation of different potential positions in the search space by gradually disregarding misleading partial matches. From this principle it follows that agents would cluster over interesting positions in the search space as soon as first agents pointing to these positions spread information to others. The mechanism responsible for this diffusion of information is probabilistic in nature and consists of assigning activity to the agents and, based on this activity, selection of agents for communication,

Initially all agents are nonactive; they are assigned to randomly chosen positions within a search space. Then each of them evaluate probabilistically its position in the search space by comparing a randomly chosen ADU from the data model with corresponding one from the search space (i.e. with the ADU in the same relative position to the reference point as in the target). If the test is successful agent becomes active, otherwise it remains inactive. This results in the role of activity label as an indicator of potentially correct solution found by corresponding agent. However, it does not exclude the possibility of false positives - signalling '1' for nontargets, nor it rules out false negatives - failing to activate on the best possible match in case when the ideal instantiation does not exist in the search space.
Next, in the diffusion phase, all of the inactive agents, and only them, individually and randomly select one agent for communication. As a result, the inactive agent is reassigned to position in the search space pointed to by chosen agent, if the latter was active, otherwise it is randomly reinitialised. All agents undergo consecutively a new testing and the whole process iterates until a termination condition based on statistical equilibrium is fulfilled. The search is terminated, if maximal number of agents pointing to the same position in the search space exceeds certain threshold and remains within specified bounds over a number of iterations [5].
From the very definition of the Stochastic Diffusion Search, it follows that we can differentiate two stages of its operation. The first one is a pure random search and the second, diffusion of activity from the agent pointing to the solution. The search engine of the Stochastic Diffusion Search is purely random, whereas diffusion enables an equilibrium state of the SDS to be achieved and is important for formulating the halting criterion. This characteristic suggests that Stochastic Diffusion Search can slow down reaching equilibrium in the case of very large spaces. Similarly, in the case of search spaces distorted heavily by noise, diffusion of activity due to disturbances will decrease an average number of inactive agents taking part in random search and in effect will increase the time needed to reach the steady state.
The probabilistic character of the algorithm enables to formulate its mathematical model in the framework of Markov Chain theory [12,2,14]. The next section will recapitulate the model of SDS, which will enable subsequent investigation of its time complexity.

3. Markov Chain Model.

Let the search space size be denoted as N, i.e. N is the number of different possible locations of the object in the search space. Let the number of agents in the Stochastic Diffusion Search be M.
Let the probability of locating the correct solution, i.e. the best instantiation of the target in the search space, in a random draw be p_m . Similarly, let the probability that, in the uniformly random draw from the search space a disturbance would be selected be p_d . Let the probability of a false positive be p^+ and the probability of false negative be p^-. The former denotes probability that the active agent pointing towards disturbance will remain active after testing and the latter denotes probability that it would become inactive after testing if it pointed to the solution.
The progress of the search is determined by the number of active agents pointing to the target's position in the search space and the latter one is influenced by the number of agents distracted by false positives. Thus the state space of a Markov Chain representing SDS can be specified as pairs of integers (a, w), where a denotes a number of active agents pointing towards the target and w - number of agents pointing towards false positives.
The one step evolution of the nonactive agent depends on the evolution phase and then on testing phase. There are different possible ways in which nonactive agent can change its state. For example, during diffusion phase it can choose, with probability a/M, an active agent pointing towards the solution and then can remain inactive with probability p^-. Other possibilities follows similarly.
The evolution of the Stochastic Diffusion Search is obviously determined by the evolution of particular agents. There are different possible ways the SDS can transfer from one state to another, e.g. increase of the overall number of active agents by one can result from one nonactive agent passing the test or two active agents becoming inactive during the test and three inactive ones becoming active and so on. One step transition probabilities for the Markov Chain model of SDS are obtained from summing up probabilities of all possible ways in which a given transition can be achieved. From the above, it follows that for $p^- \neq 0$ and

$p^+ \neq 0$ the probability of transition of the Stochastic Diffusion Search from the state *(v,b)* to *(r,a)* in one iteration is given by the formula:

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

where

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

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

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

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 $g \geq 0$.

The term $Bin(k_2,p^-)$ denotes the probability that k_2 out of v agents of the type *(a,m)* will preserve their states during testing phase and $v-k_2$ agents will evolve to the state *(n,*)*. Similarly, the term $Bin(k_1,p^+)$ denotes the probability, that k_1 out of *b* agents of the type *(a,d)* will preserve their states during testing and $b-k_1$ agents will become nonactive and switch their state to *(n,*)*. 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 evolving to the state *(a,m)*, $a-k_1$ out of $M-v-b-r+k_2$ evolving to the state *(a,d)* while those remaining ones stay 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. The one step evolution of the SDS illustrating transition probability formula for the entire search, is depicted diagramatically in graph 1.

4. Time complexity.

We will characterise the convergence rate of Stochastic Diffusion Search in the case when the target exists in the search space and there is no noise. However, the results obtained here should extend over a range of cases in which both of these assumptions are relaxed. This conclusion follows from properties of transition probability matrix of underlying Markov Chain model.
We will give time complexity estimations for synchroneous mode of operation of SDS. By this we understand dynamical mode of the Stochastic Diffusion Search in which all agents perform their tasks simultaneously.

Proposition 1.
Let $p^-=0$, $p^+=0$ and $p_d=0$. The convergence rate of Stochastic Diffusion Search is $O(-M \log(1 - \frac{1}{x}))$ for $N>N(M)$ and $O\left(-\log\left(\frac{1}{M}\left(1 - \frac{1}{x}\right)\right)\right)$ for N<N(M), where N(M) is given by

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

Proof.
For $p^-=0$, $p^+=0$ and $p_d=0$ the state transition matrix *P* is upper triangular with nonzero entries on the main diagonal (in the case of no noise, states with nonzero number of active agents pointing towards disturbances are ephemeral and therefore can be discarded from further consideration. This is equivalent to cancelling corresponding rows and columns from the transition matrix.) From spectral decomposition of *P* it follows that

$$P^n = \sum_{i=0}^{M} \lambda^n{}_i r_i l_i{}^T,$$

where l_i^T, r_i are left and right eigenvectors of *P* respectively. Matrix P is triangular so its eigenvalues lie on the main diagonal and are of the form

$$\lambda_i = p_{ii} = \left[\left(1 - \frac{i}{M}\right)\left(1 - p_m\right)\right]^{M-i},\ i=0,...,M.$$

It follows that the only eigenvalue equal to unity is λ_M, all others being strictly smaller than one. Let us define an assymptotic rate of convergence of a stochastic matrix P as

$$R_\infty = -\log(\rho(P_1)),$$

where P_1 is a matrix constructed from P by deleting a row and a column corresponding to the unit eigenvalue and ρ is a spectral radius of this matrix.
From this it follows that in order to determine the rate of convergence of Stochastic Diffusion Search, it is sufficient to investigate the behaviour of the powers of the dominant eigenvalue smaller than one. Examining the function

$$p_{M,p_m}(x) = [(1-\tfrac{x}{M})(1-p_m)]^{M-x},\ x\in[0,M-1]$$

shows, that it possesses a proper global minimum and two maxima for *x=0* and *x=M-1*. The values of maxima vary with M and p_m . Therefore in order to locate a global maximum, it is sufficient to locate regions on the plane, where the function

$$h(y,z) = p_{y,z}(0) - p_{y,z}(y-1),\ (y,z)\in(1,\infty]\times(0,1)$$

has a constant sign. For this we need to solve an equation in two variables

$$h(y,z) = 0,$$

i.e.

$$(1-z)^y = \tfrac{1}{y}(1-z).$$

Dividing both sides by (1-z) and taking logarithm yields

$$(y-1)\log(1-z) = -\log y.$$

From this we obtain

$$\log(1-z) = \log y^{\frac{-1}{y-1}},$$

and because of strict monotonicity of logarithm after rearranging we obtain finally

$$z = 1 - y^{\frac{-1}{y-1}}.$$

From the plot of z(y) we can see, that this function divides $(1,\infty]\times(0,1)$ into two regions where the function h(y,z) has a different constant sign. Going back to variables M, p_m leads to

$$\max_{i\in\{0,\ldots,M-1\}} \lambda_i = \begin{cases} (1-p_m)^M, & for\ p_m \le p_m(M) \\ \frac{1}{M}(1-p_m), & for\ p_m \ge p_M(M), \end{cases}$$

where

$$p_m(M) = 1 - M^{\frac{-1}{M-1}}.$$

So the assymptotic rate of convergence is

$$R_\infty = \begin{cases} -\log\left(\frac{1}{M}(1-\frac{1}{N}\right), & for\ N \le N(M) \\ -M\log\left(1-\frac{1}{N}\right), & for\ N \ge N(M) \end{cases}$$

■

From graph 2 we see, that the function describing the average convergence rate of the SDS versus the search space size, tends very quickly to its asymptotic assuring that estimation is very accurate for all *N>N(M).*

Also we can observe, that for a fixed N the rate is inversely proportional to the number of agents used in search. We see that from the point of view of convergence, the more agents we use the shorter the time to converge.

As we could see, the value of *N(M)* depends on M only and determines the points of change in the convergence characteristic of the SDS.

Definition 1. We will call a search space size *N(M),* *defined by*

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

a *supercritical value* for M and *N(M)* as *supercritical curve.*

In the second part of the proof we used the fact, that function g, defined as a mean convergence rate, converges to a constant for fixed M, when its argument tends to infinity, even though in this case *N<N(M)* i.e. it by no means approaches infinity. However the result is valid in the sense, that g as strictly increasing, is bounded above by its limit for all values of *N<N(M).* Conversely to the case of the first estimate, where we could consider a mean convergence time to be a good approximation of a linear function of the search space size, this is not the case for the estimation below the supercritical value of N. The definition of the subcritical value given below will allow us to refine the result of the last proposition and to determine the region in the N×M plane, in which $^1/_{logM}$ in the parallel, connectionist case ($^M/_{logM}$ in the sequential implementation) is a good approximation of the convergence time for the Stochastic Diffusion Search in the case of no noise and complete instantiation of the data model in the search space.

Let us first define a minimal *N*, for which function *g(N)* is well approximated by $^1/_{logM}$. ($^M/_{logM}$.).

Definition 2. We call N_m *minimal*, if for a given M and a tolerance $p\in(0,1)$,

$$N_m = \min\left\{N: \tfrac{1}{\log M} - g(N) \le p\tfrac{1}{\log M}\right\}.$$

Similarly we will call *N_m(M)* a minimal curve.

In the sequential implementation case we have analogously

$$N_m = \min\left\{N: \tfrac{M}{\log M} - g(N) \le p\tfrac{M}{\log M}\right\}.$$

The convergence and strict monotonicity of the function g implies, that for any M there are infinitely many values of N for which g(N) is close to the constant $^1/_{logM}$ ($^M/_{logM}$) i.e. N_m is well defined.

Let us now determine *N_m(M).* From

$$n - \tfrac{1}{\log M} = p\tfrac{1}{\log M}$$

and

$$n = \tfrac{-1}{\log\frac{1}{M}(1-\frac{1}{N})},$$

after rearranging we obtain

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

Calculation in the sequential case are analogous and lead to the same formula.

Now we can define subcritical value N_s.

Definition 3. We call N_s a *subcritical search space size* for a given M and p∈(0,1), if N_s is minimal and not greater than *N(M).* Similarly we call $N_s(M)$ a subcritical curve.

It follows from the above definition, that $N_s(M)$ is not defined for all possible *M* and coincides with $N_m(M)$ where the latter is defined.
From graph 3 one can observe, that both supercritical and subcritical curves divide the plane N×M into 3 interesting regions:

1. Region I, in which convergence time depends linearly on the search space size;
2. Region II, in which convergence time is approximately constant and independent of the search space size;
3. Region III, in which convergence time increases nonlinearly with the search space size.

Up to now we kept the assumption given in [5], that the number of agents M involved in Stochastic Diffusion Search is equal to the size of the model. However from the analysis of time complexity of SDS it follows that we could set the number of agents in the SDS to any arbitrary value. This would result in a change of the convergence time to the best solution but the overall characteristic of time complexity would remain analogous and it would not affect the quality of the eventual solution either. Of course in the most general case, for given search conditions there is a certain critical number of agents, below which also average quality of search could be affected.
It seems interesting therefore to characterise the Stochastic Diffusion Search in terms of the actual number of agents used in the search. Two values of M play an important role in this characterisation. Let us define a number of agents M_0, for which the supercritical curve intersects subcritical one, as *critical number* for a given tolerance $p \in (0,1)$, and let us denote by M_1, a number of agents for which subcritical N equals two.

From straightforward calculations it follows, that

$$M_0 = \frac{1}{p}, \text{ and } M_1 = 2^{\frac{1}{p}-1}.$$

Now for each $M > M_0$, there exist range of search space sizes, for which convergence time is approximately constant and for $M > M_1$, time for convergence is either constant or increases linearly with *N*. Graph 4 shows average convergence time as a surface over N×M plane and graphs 5 and 6 show crossections of this surface together with appropriate asymptotics for different values of M.

5. Numerical examples.

In this section we will present and discuss results of some simulations performed with Stochastic Diffusion Search for different values of search parameters. These simulations will illustrate some key points of the theoretical discussion carried in the previous section. Experiments were performed on a toy, artificial task, which nevertheless can be generalised to account for a wide range of interesting practical problems. The goal of Stochastic Diffusion Search was to locate, in a string of characters from a finite alphabet, a given string or if it has not appeared - its best instantiation. The size of the search space was therefore given by the number of ways we could choose a string of given length from the long string representing the search space. We introduced noise to the search space by inserting at random positions substrings of a given data model. We implemented the algorithm in Turbo Pascal and run experiments on IBM PC 486 DX with 8 Mb RAM. Hardware and compiler constraints imposed quite severe restrictions on the size of the problems considered. Namely the maximal number of agents used in the search was restricted to 70 and the maximal search space size to 300. A random number generator from [13] was used instead of the standard routine provided by Turbo Pascal. The random number generator was reported in [13] as giving in practical applications infinite sequences of pseudo -random numbers and performed much better than original one in statistical tests we performed. In the simulations the following parameters describe the operating conditions for the Stochastic Diffusion Search:
- N - the size of the search space;
- l - the size of the model;
- M - the number of agents;
- p_d- the probability of choosing a disturbance in a uniformly random draw from the search space;
- p^+- the probability of a disturbance being improperly classified;
- p^- - the probability of incorrect classification of the best instantiation of the model.

5.1. Noise free search space.

These experiments aim to illustrate different performance characteristics of the Stochastic Diffusion

Search for different regions of N×M plane. Here the search space is free of disturbances and the target is present in the search space. From the Markov chain model we know that SDS will converge to the correct position and once at this position it will point there constantly. We are therefore interested in the number of iterations necessary for the Stochastic Diffusion Search to point to the correct position for the first time. The first experiment is performed with number of agents far exceeding the size of the model. The search space size varies from the size of the model to N(M). The values of parameters are following: M=70, l=3; N varies from 4 to 24. We run the Stochastic Diffusion Search 100 times for each search space size and register number of iterations necessary for convergence. The result of the experiment - averages from 100 runs for each search space size together with two standard deviations band are shown in graph 7.

In the graph we can see that, according to the results from the previous section, convergence time depends in a non-linear way on the search space size. From 2 standard deviations band it follows, that results from all runs are very well concentrated around averages. It also can be seen from the graph that in spite of the fact that initially the model occupies 75% of the search space it takes on average about 5 iterations for the Stochastic Diffusion Search to converge. This may be a result of time needed to diffuse the correct position among all agents, but also suggest that the positioning-testing mechanism should be inspected closely. One could naturally expect that in this case, i.e. when the model constitutes a significant part of the search space it should affect the convergence time resulting in quick convergence. However we see that in the present definition of the Stochastic Diffusion Search the definition of the positioning-testing mechanism is independent of the model size.

Graph 8 displaying the skewness of the sample distributions suggests that there is no pattern in the distributions of convergence times around respective averages.

In fact the graph 7 displays the Stochastic Diffusion Search performance for two regions in the N×M plane: the region of non-linear growth with the search space size and the region of the approximate saturation. However, it is difficult to distinguish in the graph a saturation region, where according to the proposition Stochastic Diffusion Search should converge approximately independently of the search space size. It is because of the limitations on the maximal number of agents we could use in our simulations. This number further restricts search space sizes to 24. This causes that only slow non-linear growth of convergence time with the search space size can be observed.

The aim of the next two experiments is twofold. They show Stochastic Diffusion Search convergence time for N>N(M) and also illustrate the effect of performing search with the number of agents equal to and different from the target size. Therefore in the first of the two experiments we use 70 agents in the SDS, set the target size to 3 and vary the search space size from 250 to 300. In the second experiment the number of agents is equal to the size of the target - M=l=5, and the search space sizes vary as in the first experiment. In these and all next experiments we run SDS 500 times for each search space size.

Graphs 9, and 10 show average convergence times with two standard deviations band, and skewness respectively for the first experiment. Corresponding quantities for the second experiment are displayed in graphs 11, and 12.

From graphs 9 and 11 we see that in both cases the average number of iterations needed by the Stochastic Diffusion Search to converge follows straight line. However, in case of number of agents exceeding the target size the average convergence time is much shorter. Accordingly, in this case standard deviation is much smaller than in the second case. This suggests, in agreement with intuition, that the more agents we use in the search, the faster and more consistent the convergence.

It follows also that, as we deduced from the construction of the model, there is no reason to restrict the number of agents by the model size.

5.2. Experimenting with a noisy search space.

The last set of experiments aims at providing some insight into influence of noise on the convergence of the algorithm. In these experiments therefore the model also exists in the search space and we are again interested in the convergence time of SDS. In the light of the conclusions concerning the number of agents in the Stochastic Diffusion Search we performed our experiments with M=70 and l=3. The search space was distorted by a moderate amount of noise characterised by the following values of parameters: p_d=0.1, p^+=0.2. Two experiments were performed to check the behaviour of the Stochastic Diffusion Search in the main regions of the N×M plane. In the first experiment the search space size varied from 4 to 24 and in the second one from 250 to 300. The results for the case N<N(M) are displayed in graphs 13 and 14 and for the case N>N(M) in graphs 15 and 16 respectively.

From all 4 graphs it follows that in the presence of moderate amount of noise the Stochastic Diffusion Search characteristic found theoretically and confirmed in previous experiments remains invariant.

In the case of N<N(M) results show non-linear dependence on the search space size well concentrated around averages, whereas for N>N(M) linear dependence is preserved together with higher dispersion of samples around averages. Also skewness different characteristics for both cases remain the same. The above observations suggest, that the magnitude of dispersion as well as skewness of samples are characteristic features of two main components of the Stochastic Diffusion Search, namely random search engine and diffusion of positions across agents. Graphs 13 and 15 reveal that there is no significant difference between average convergence times in the presence of disturbances and in the noise-free search space. This means that Stochastic Diffusion Search is quite robust to the noise in the search space.

6. Conclusions and further research.

In this article we have discussed time complexity of the Stochastic Diffusion and have given a basic analysis of it. Our analysis was based on the theory of Markov Chains. We have identified two principal components of the Stochastic Diffusion Search: the random search engine and the diffusion of activation among the agents. We also have given a detailed characterisation of the convergence time of the algorithm in terms of the search space size and the number of agents used in the search.

It has been shown, that in the presence of the data model in the search space and no noise the SDS is time sub-linear with the search space size for spaces greater than a supercritical space size and roughly time constant for spaces smaller than supercritical. From the definition of the Stochastic Diffusion Search it follows that it is quite independent of the search space characteristic. The time characterisation obtained can be therefore attributed to the search itself. It also ensures that exploring properties characteristic to a given problem could speed up the convergence of the search to the optimal solution. These properties can be extracted from the problem prior to an execution of the search and implemented in the form of sampling from different than uniform probability distribution over the search space. This would correspond to knowledge insertion technique used in connectionist models to enhance and speed up learning. Such methods work fine, but have a limited use in the situation when there is no previous knowledge of the problem or its analysis is too difficult.

Stochastic Diffusion Search can also offer an interesting solution for such hard problems. Starting from the uniform probability distribution corresponding to having no previous knowledge about the search space it could adaptively estimate in each time step a true, unknown probability distribution based on the past history. In other words this would correspond to process of knowledge acquisition and learning incorporated to enhance search. Such changes from uniform probability distribution would also provide one possible way of focusing the search on more promising areas of the search space.

The possible multimodality of this distribution would cause parallel focus of attention on probabilistic basins of attraction whereas dynamical adaptation of this distribution would encompass sequential attention and attention switching. One can also go a step further and imagine the situation of dynamically evolving search space. This would mean that the true probability distribution is not static but changes in time. Again this situation could be captured by the adaptive probability estimation of the search space.
The experiments performed highlighted some of the features of this search mechanism. They also suggest that the time characteristic extends also for the spaces without perfect instantiation of the data model and distorted by noise.
Our model has however several limitations. First of all, modelling disturbances by means of two parameters only, suggests that results concerning steady state distribution should be considered as rough estimations of the actual equilibrium probabilities. Secondly, in the most general case our model is an ergodic, Markov chain with (m+1)*(m+2)/2 states, where m is the number of agents. Therefore even though we have given explicit formulas for transition probabilities, analytic calculations for nontrivial number of agents used in search are next to impossible due to explosion of the Markov chain's state space. The intricate transition probability formula makes it difficult to take noise parameters into account in the estimation of time complexity.
In the future research we would like to concentrate on the further analysis of the algorithm, including noise coefficients into consideration as well as providing some bounds on the search performance probabilities. We will also try to improve overall algorithm's performance, i.e. convergence time and dispersion of results from several runs by

References:

[1] I. Aleksander and T. J. Stonham, "Guide to Pattern Recognition using Random Access Memories," *Computers & Digital Techniques*, vol. 2, no 1, pp.29-40, 1979.
[2] T. J. Bailey, *The elements of Stochastic Processes*. New York: John Wiley & Sons, Inc., 1964.
[3] R.S. Bird, "Two dimensional Pattern Matching," *Information Processing Letters*, vol. 6, no. 5, pp. 168-170, 1977.
[4] J. M. Bishop, "Stochastic Searching Networks," *Proceedings of the 1st IEE Conf. on ANN*, pp 329-331, 1989.
[5] J.M. Bishop and P. Torr, "The Stochastic Search Network," in *Neural Networks for Images, Speech and Natural Language*, R. Linggard, D.J. Myers & C Nightingale Eds. New York: Chapman & Hall, 1992.
[6] E. Grech-Cini, Motion Tracking in Video Films. PhD Thesis, University of Reading, 1995.
[7] G. E. Hinton, "A parallel computation that assignes canonical object-based frames of reference," *Proceedings of 7th International Joint Conference on Artificial Intelligence,* 1981.
[8] Ch. M Hoffman and J. O'Donnell, "Pattern Matching in Trees," *Journal of ACM*, vol. 29, no. 1, pp. 68-95, 1982.
[9] R. Horn, and Ch. Johnson, *Matrix Analysis*. Cambridge: Cambridge University Press, 1994.
[10] S. Z. Li, "Matching: Invariant to Translations, Rotations and Scale Changes," *Pattern Recognition,* vol. 25, no. 6, pp. 583-594, 1992.
[11] D. Marr, *Vision*. San Francisco: W.H. Freeman and Co, 1982.
[12] S. Nasuto and J. M. Bishop, "Convergence Analysis of the Stochastic Diffusion Search," submitted for publication.
[13] W. H. Press et. al., *Numerical Recipes in Pascal*. Cambridge: Cambridge University Press, 1992.
[14] E. Seneta, *Nonnegative Matrices*. New York: Springer, 1981.
[15] L. G. Shapiro and R. M. Haralick, "Structural Description and Inexact Matching," *IEEE Transations on Pattern Analysis and Machine Intelligence,* vol. 3, pp. 504-519, 1981.
[16] J. Tarhio, "A Sublinear Algorithm for Two-Dimensional String Matching," *Pattern Recognition Letters*, vol. 17, no. 8, pp. 833-838, 1996.
[17] J. van Leeuwen (Ed.), *Handbook of Theoretical Computer Science. Algorithms and Complexity.* Amsterdam: Elsevier, 1990.

Let us first consider the case, when $p_m > p_m(M)$. Denoting by $p \in (0,1)$ a desired error bound , from

$$p = (1 - p_m)^{Mn}$$

it easily follows, that

$$n = \frac{a}{M \ln(1-p_m)},$$

where $a=-\ln p > 0$.
Noticing that p_m is proportional to the reciprocal of the search space size, $p_m \sim 1/N$, and taking into account, that $N(M)$ has to be an integer it follows, that estimation of n in terms of the search space size N, such that

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

is given by

$$n = \frac{-a}{M \ln(1-\frac{1}{N})}$$

From the above it follows, that for $N > N(M)$ we can analyse dependence of n on N by examining the asymptotic behaviour of the function

$$f(x) = \frac{-a}{M \ln\left(1-\frac{1}{x}\right)},$$

when x tends to infinity. Thus, we have

$$\lim_{x \to +\infty} \frac{f(x)}{x} = \frac{-a}{M} \lim_{x \to +\infty} \frac{1}{x \ln\left(1-\frac{1}{x}\right)} =$$

$$= \frac{-a}{M} \lim_{x \to +\infty} \frac{1}{\ln\left(1-\frac{1}{x}\right)^x} = \frac{a}{M}.$$

Similarly

$$\lim_{x \to +\infty} \left(f(x) - \frac{a}{M} x\right) = \frac{-a}{M} \lim_{x \to +\infty} \left(\frac{1}{\ln(1-\frac{1}{x})} + x\right) = \frac{-a}{2M}$$

The last equality in the above equation can be obtained by expanding the rational expression in the parenthesis in Taylor series.
We see that $f(x)$ approaches a linear function

$$f(x) - (cx + d) \xrightarrow{x \to +\infty} 0,$$

where

$$c = \frac{a}{M},$$

$$d = \frac{-a}{2M}.$$

So in terms of the search space size N, for $N > N(M)$ we can write

$$n \approx \frac{a}{M} N - \frac{a}{2M},$$

i.e. algorithm is probabilistically bounded by $O\left(\frac{a}{M} N - \frac{a}{2M}\right)$.

For the case of $N \leq N(M)$ we obtain

$$p = \left[\frac{1}{M}\left(1 - \frac{1}{N}\right)\right]^n.$$

Taking logarithm of both sides and rearranging leads to

$$n = \frac{-a}{\log \frac{1}{M}(1-\frac{1}{N})}.$$

Let consider behaviour of the function

$$g(x) = \frac{-a}{\log \frac{1}{M}(1-\frac{1}{x})}.$$

We obtain

$$\lim_{x\to+\infty}\frac{g(x)}{x}=\lim_{x\to+\infty}\frac{1}{x}\frac{-a}{\log\frac{1}{M}(1-\frac{1}{x})}=$$

$$=\lim_{x\to+\infty}\frac{-a}{\log(1-\frac{1}{x})^{x}-\log M^{x}}=0$$

and

$$\lim_{x\to+\infty}g(x)=\lim_{x\to+\infty}\frac{-a}{\log\frac{1}{M}(1-\frac{1}{x})}=$$

$$=\lim_{x\to+\infty}\frac{-a}{\log(1-\frac{1}{x})-\log M}=\frac{a}{\log M}.$$

Therefore in terms of the size of the search space

$$n\longrightarrow\frac{a}{\log M},\ when\quad N\to\infty.$$

■

Captions:

Graph 1. 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.

Graph 2. Plot of number of iterations to converge, n ,versus the search space size, *N*, *N>N(M).* Connectionist case; *p=0.01, M=5.*

Graph 3. Partition of *N×M* plane into regions of different convergence time characteristic. Area I corresponds to linear time complexity, area II - to convergence independent of the search space size and area III to saturation of the SDS. Connectionist case.

Graph 4. Average convergence time, n, as a function of the number of agents M, and the search space size N.

Graph 5. Plot of number of iterations to converge, n, versus the search space size together with asymptotics. Crossection along the constant $M<M_0$. Connectionist case.

Graph 6. Plot of number of iterations to converge, n, versus the search space size together with asymptotics. Crossection along the constant $M>M_0$. Connectionist case.

Graph 7. Average convergence time for search space sizes smaller than the supercritical value, N<N(M) together with two standard deviations band. Sequential implementation. Each point on the curve corresponds to average from 100 runs; M=70, l=3; N varies from 4 to 24.

Graph 8. Skewness as a function of the search space size N, for N<N(M). Sequential implementation. Each point on the curve corresponds to average from 100 runs; M=70, l=3; N varies from 4 to 24.

Graph 9. Average convergence time and two standard deviations band for the Stochastic Diffusion Search in the case of N>N(M) and for relatively large amount of agents. Sequential case. Each data point correspond to an average of 500 runs; M=70, l=3, N vary from 250 to 300, $p_d=p^+=0$.

Graph 10. Skewness of the Stochastic diffusion Search in the case of N>N(M) and for relatively large amount of agents. Sequential case. Each data point correspond to an average of 500 runs; M=70, l=3, N vary from 250 to 300, $p_d=p^+=0$.

Graph 11. Average convergence time and two standard deviations band for the Stochastic Diffusion Search in the case of N>N(M) and for relatively small amount of agents. Sequential case. Each data point correspond to an average of 500 runs; M=5, l=5, N vary from 250 to 300, $p_d=p^+=0$.

Graph 12. Skewness of the Stochastic Diffusion Search in the case of N>N(M) and for relatively small amount of agents. Sequential case. Each data point correspond to an average of 500 runs; M=5, l=5, N vary from 250 to 300, $p_d=p^+=0$.

Graph 13. Average convergence time together with two standard deviation band for the Stochastic Diffusion Search in the presence of noise for N<N(M); Sequential case; M=70, l=3, $p_d=0.1$,$p^+=0.2$, N varies from 4 to 24.

Graph 14. Skewness of variation of the Stochastic diffusion Search in the presence of noise for N<N(M); Sequential case; M=70, l=3, $p_d=0.1$, $p^+=0.2$, N varies from 4 to 24.

Graph 15. Average convergence time together with two standard deviation band for the Stochastic diffusion Search in the presence of noise for N>N(M); Sequential case; M=70, l=3, $p_d=0.1$, $p^+=0.2$, N varies from 4 to 24.

Graph 16. Skewness of the Stochastic Diffusion Search in the presence of noise for N>N(M); Sequential case; M=70, l=3, $p_d=0.1$, $p^+=0.2$, N varies from 4 to 24.