Minimum Stable Convergence Criteria for Stochastic Diffusion Search

D.R.Myatt, J.M.Bishop and S.J. Nasuto

November 19, 2003

D.R. Myatt[1] and S.J. Nasuto (*Department of Cybernetics, The University of Reading, Whiteknights, Reading RG6 2AY, United Kingdom*)

J.M. Bishop (*Department of Computing, Goldsmiths College, University of London, New Cross, London SE14 6NW, United Kingdom*)

[1]d.r.myatt@reading.ac.uk

Abstract

An analysis of Stochastic Diffusion Search, a novel and efficient optimisation and search algorithm, is presented, resulting in a derivation of the minimum acceptable match resulting in a stable convergence within a noisy search space. The applicability of SDS can therefore be assessed for a given problem.

Introduction: Stochastic Diffusion Search (SDS), first described in [1], is an efficient probabilitistic multi-agent global search and optimisation technique that has been applied to diverse problems such as site selection for wireless networks [2], mobile robot self-localisation [3], object recognition [4] and text search [5]. Additionally, a hybrid SDS and n-tuple RAM [6] technique has been used to track eyes in video sequences [7]. Previous analysis of SDS has investigated its convergence [8] and resource allocation [5] using Markov Chains and Ehrenfest Urn models under a variety of noise conditions. This paper will outline the derivation of a simpler convergence condition, (underpinning that suggested in [10]), and illustrates its implications through appropriate numerical simulations.

Stochastic Diffusion Search: SDS can be used for pattern search and matching. Such problems can be cast in terms of optimisation by defining the objective function, $F(\mathbf{x})$, for a hypothesis $\mathbf{x}$ about the location of the solution, as the similarity between the target pattern and the corresponding region at $\mathbf{x}$ in the search space and finding $\mathbf{x}$ such that $F(\mathbf{x})$ is maximised. In general SDS can most easily be applied to optimisation problems where the objective function is decomposable into components that can be evaluated independently:

$$F(\mathbf{x}) = \sum_{i=1}^{n} F_i(\mathbf{x}), \tag{1}$$

where $F_i(\mathbf{x})$ is defined as the i^{th} partial evaluation of $F(\mathbf{x})$.

In order to locate the optima of a given objective function SDS employs a population of k agents, each of which maintains a hypothesis about the optima. In operation the algorithm entails the iteration of Test and Diffusion Phases until agents converge upon the optimum hypothesis.

Initialisation: Typically the initial hypothesis of each agent is selected uniformly randomly over the search space. If information about probable solutions is available *a priori* this can be used to bias the initial selection of hypotheses.

Test Function: The boolean Test Function returns whether a randomly selected partial evaluation of the objective function is indicative of a 'good' hypothesis. E.g. In pattern matching the Test Function may return True if the i^{th}

sub-feature of the target pattern is present at position $(\mathbf{x}, i)$ in the search space.

The *Test Score* for a given hypothesis, $\mathbf{x}$, is the probability that the Test Function will return true, and is hence representative of the value of $F(\mathbf{x})$.

Test Phase: Each agent applies the Test Function to its current hypothesis. If the Test Function returns true the agent becomes *active* and otherwise becomes *inactive*.

Diffusion Phase: Each *inactive* agent, A, select another agent B at random. If B is active then the hypothesis of B is copied to A. Conversely, if B is also inactive then A selects another hypothesis randomly over the entire search space.

Convergence: As iterations progress clusters of agents with the same hypothesis form. At convergence the largest cluster of agents defines the optimal solution.

Expected cluster size formulation of SDS*:* In this section the minimum Test Score, $\alpha_{\min}$, for which a stable cluster of agents can form will be derived. A simplifying assumption is that, by considering only the mean transition of agents between different clusters, rather than the full probability distribution (as investigated in [5]), a sufficiently accurate model of SDS may be produced. The noise model that will be assumed is that of 'homogeneous background noise', where every non-optimal hypothesis corresponds to a distractor of Test Score β. It is also assumed that there is a single optimal solution with Test Score α that has a negligible probability of being selected. Let $\bar{c}_i$ be the mean number of active agents with the optimal solution as a proportion of the total population.

During the diffusion phase, the mean number of inactive agents selecting an agent within the optimal cluster is given by $g(\bar{c}_i, \alpha, \beta)\bar{c}_i$, where g yields the number of inactive agents for a given iteration. From Figure 1 g can be immediately written as

$$g(\bar{c}_i, \alpha, \beta) = \frac{1-\alpha}{\alpha}\bar{c}_i + (1-\beta)\left(1 - \frac{\bar{c}_i}{\alpha}\right). \tag{2}$$

Therefore, the function f that defines the mean 1-step optimal cluster size evolution is

$$\bar{c}_{i+1} = f(\bar{c}_i, \alpha, \beta) = \alpha\left(\bar{c}_i + g(\bar{c}_i, \alpha, \beta)\bar{c}_i\right). \tag{3}$$

Figure 2 considers (3) as a 1 dimensional iterated map, and graphically it can be seen that for a non-zero attractor to exist the condition

$$\frac{df}{d\bar{c}_i} > 1 \tag{4}$$

must hold for $\bar{c}_i = 0$. Differentiating (3) wrt $\bar{c}_i$ yields

$$\frac{df}{d\bar{c}_i} = \alpha\,(2-\beta) - 2\bar{c}_i\,(\alpha-\beta) \tag{5}$$

and it follows that the minimum value of α for which the constraint in (4) holds is

$$\alpha_{\min} = \frac{1}{2-\beta}. \tag{6}$$

Hence, for any $\alpha < \alpha_{\min}$ the size of the cluster will tend to zero for *any* initial cluster size and the search will fail.

Numerical Results: SDS was performed on two simulated search spaces, one with an optimal solution $\alpha = \alpha_{\min} + 0.01$ and the other $\alpha = \alpha_{\min} - 0.01$, while homogeneous background noise was varied between 0 and 0.99 in increments of 0.01. Initially *all* agents were associated with the optimal solution but the probability of an agent subsequently selecting it from the search space was zero. 10000 agents were used in each case and the cluster size of the optimal solution was measured after 5000 iterations.

Figure 3 shows that after 5000 iterations the cluster size of the search with $\alpha < \alpha_{\min}$ had returned to zero (as expected), while SDS successfully retained a cluster for $\alpha > \alpha_{\min}$. The experimental value of $\alpha_{\min}$ has therefore been shown to be ± 0.01 of the theoretical, validating the theory.

Conclusion: This paper has described a novel formulation for the SDS algorithm that allows the calculation of the minimum match, $\alpha_{\min}$, in a given search space that will guarantee stable convergence of SDS. In practical situations when the simplifying assumptions are violated (such as the background noise showing deviation from homogeneity) this value will still provide a useful estimate.

References

[1] J.M. Bishop.: 'Stochastic Searching Networks'. Proc. 1st IEE Conf. on Artifical Neural Networks, 1989, (London), pp. 329-331

[2] R.M. Whitaker, S. Hurley.: 'An agent based approach to site selection for wireless networks'. Proc ACM Symposium on Applied Computing (Madrid), March 2002, pp. 574-577

[3] N.I. Katevas, N.M. Sgouros, S.G. Tzafestas, G. Papakonstantinou, P. Beattie, J.M. Bishop, P. Tsanakas, D. Koutsouris.: 'The Autonomous Mobile Robot SENARIO: A Sensor-Aided Intelligent Navigation System for Powered Wheelchairs'. IEEE Robotics and Automation 4(4) 1997, pp. 60-70

[4] H.J.Grech-Cini, G.T.McKee.: 'Locating the Mouth Region in Images of Human Faces'. In P.S.Schenker (Ed.), Proceedings of SPIE - The International Society for Optical Engineering, Sensor Fusion VI 2059, (Massachusetts), 1993

[5] S.J. Nasuto.: 'Resource allocation of the Stochastic Diffusion Search'. PhD Thesis, 1999, (University of Reading)

[6] I. Aleksander and T. J. Stonham. 'A guide to pattern recognition using random-access memories'. IEE Proceedings-E Computers and Digital Techniques 2(1), 1979, pp. 29-40.

[7] J.M. Bishop, P. Torr.: 'The Stochastic Search Network'. In R. Linggard, D.J. Myers, C. Nightingale (Eds.), Neural Networks for Images, Speech and Natural Language, 1992, New York, Chapman and Hall

[8] S.J. Nasuto, J.M. Bishop.: 'Convergence Analysis of Stochastic Diffusion Search'. Journal of Parallel Algorithms and Applications 14:2, 1999, pp. 89-107

[9] S.J. Nasuto, J.M. Bishop, S. Lauria.: 'Time Complexity of Stochastic Diffusion Search'. Neural Computation '98, 1998, (Vienna)

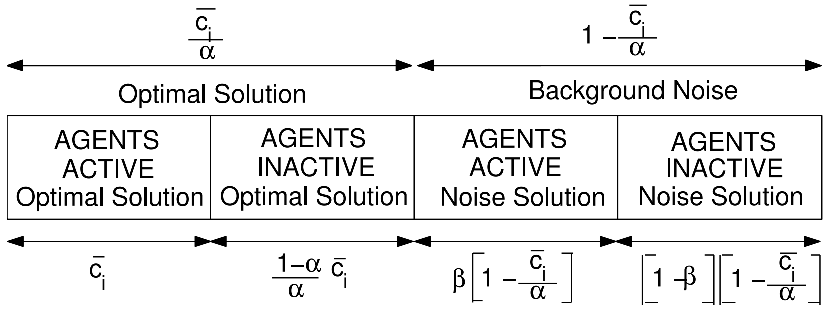
[10] H.J. Grech-Cini.: 'Locating Facial Features'. PhD Thesis, 1995, (University of Reading)

List of captions and subcaptions for figures

Figure 1. Illustration of the current state of the agent population in iteration i in terms of $\bar{c}_i$, α and β.

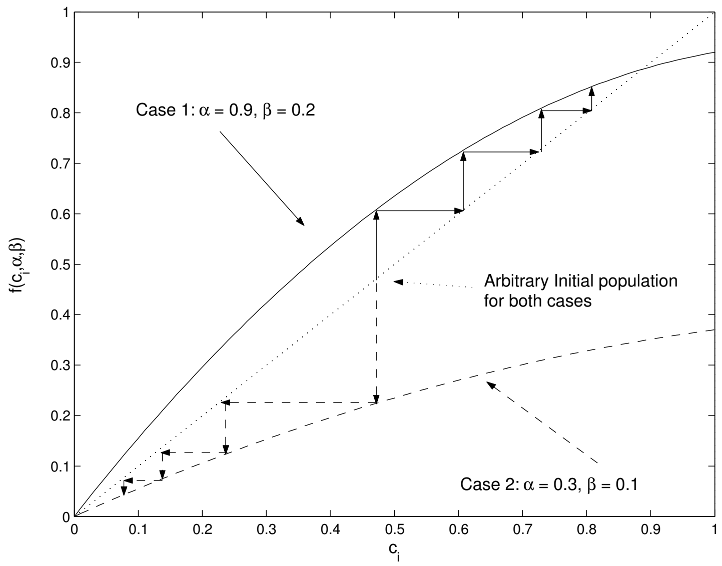
Figure 2. A 1 dimensional iterated map showing how different values of α and β can either result in a stable optimal cluster or a return to zero.

Figure 3. A graph showing the cluster size after 5000 iterations for values of $\alpha = \alpha_{\min} \pm 0.01$. Insets show the evolution of the optimal cluster size for $\beta = 0.8, \alpha = \alpha_{\min} \pm 0.01$.



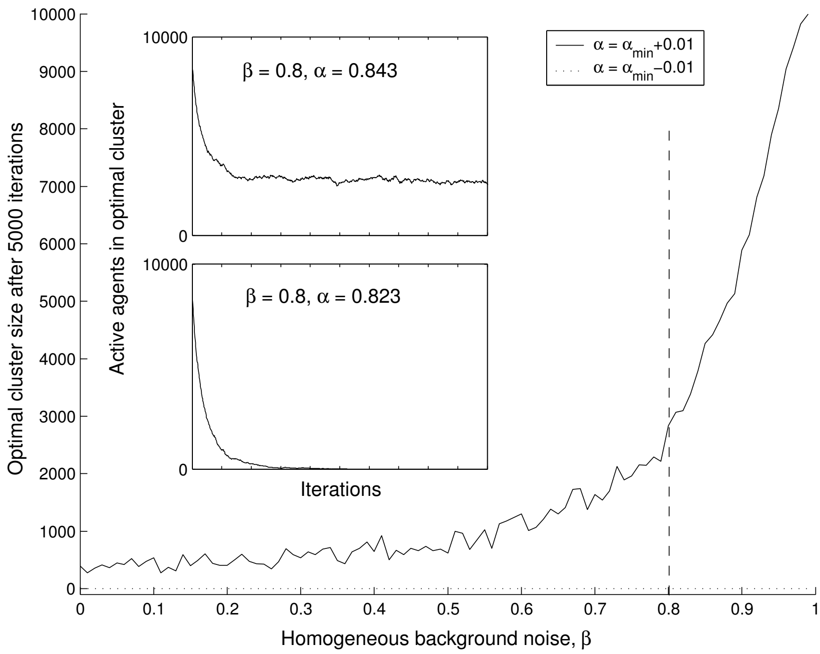
$\frac{\bar{c}_i}{\alpha}$
$1 - \frac{\bar{c}_i}{\alpha}$
Optimal Solution
Background Noise
AGENTS ACTIVE Optimal Solution
AGENTS INACTIVE Optimal Solution
AGENTS ACTIVE Noise Solution
AGENTS INACTIVE Noise Solution
$\bar{c}_i$
$\frac{1-\alpha}{\alpha} \bar{c}_i$
$\beta \left[1 - \frac{\bar{c}_i}{\alpha}\right]$
$\left[1 - \beta\right]\left[1 - \frac{\bar{c}_i}{\alpha}\right]$

Figure 1:



Case 1: α = 0.9, β = 0.2
Arbitrary Initial population
for both cases
Case 2: α = 0.3, β = 0.1
f(c_i,α,β)
c_i

Figure 2:



10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Optimal cluster size after 5000 iterations
Homogeneous background noise, β
Active agents in optimal cluster
β = 0.8, α = 0.843
β = 0.8, α = 0.823
Iterations
α = α_min+0.01
α = α_min−0.01

Figure 3: