Stabilizing swarm intelligence search via positive feedback resource allocation

Slawomir Nasuto[1] and Mark Bishop[2]

[1] Slawek Nasuto, Dept. Cybernetics, University of Reading, UK.
s.j.nasuto@reading.ac.uk
[2] Mark Bishop, Dept. Computing, Goldsmiths, University of London, UK.
m.bishop@gold.ac.uk

Summary. A novel Swarm Intelligence method for best-fit search, Stochastic Diffusion Search, is presented capable of rapid location of the optimal solution in the search space. Population based search mechanisms employed by Swarm Intelligence methods can suffer lack of convergence resulting in ill defined stopping criteria and loss of the best solution. Conversely, as a result of its resource allocation mechanism, the solutions SDS discovers enjoy excellent stability.

1 Introduction

In recent years there has been growing interest in swarm intelligence, a distributed mode of computation utilising interaction between simple agents [16]. Such systems have often been inspired by observing interactions between social insects: ants, bees, birds (cf. Ant Algorithms and Particle Swarm Optimisers) see Bonabeau [8] for a comprehensive review. Swarm Intelligence algorithms also include methods inspired by natural evolution such as Genetic Algorithms [12, 14] or indeed Evolutionary Algorithms [3]. The problem solving ability of Swarm Intelligence methods, emerges from positive feedback reinforcing potentially good solutions and the spatial/temporal characteristics of their agent interactions.

Independently of these algorithms, Stochastic Diffusion Search (SDS), was first described in 1989 as a population-based, pattern-matching algorithm [5]. Unlike stigmergetic communication employed in Ant Algorithms, which is based on modification of the physical properties of a simulated environment, SDS uses a form of direct communication between the agents similar to the tandem calling mechanism employed by one species of ants, Leptothorax Acervorum, [17].

SDS is an efficient probabilistic multi-agent global search and optimisation technique [11] that has been applied to diverse problems such as site selection for wireless networks [23], mobile robot self-localisation [4], object recognition [6] and text search [5]. Additionally, a hybrid SDS and n-tuple RAM [1] technique has been used to track facial features in video sequences [6, 13].

Previous analysis of SDS has investigated its global convergence [19], linear time complexity [20] and resource allocation [18] under a variety of search conditions.

1.1 Global optimisation

Global optimisation algorithms have been recently classified in terms of their theoretical foundations into four distinct classes [22]:

- incomplete methods - heuristic searches with no safeguards against trapping in a local minimum;
- asymptotically complete - methods reaching the global optimum with probability one if allowed to run indefinitely long without means to ascertain when the global optimum has been found;
- complete - methods reaching the global optimum with probability one in infinite time that know after a finite time that an approximate solution has been found to within prescribed tolerances;
- rigorous - methods typically reaching the global solution with certainty and within given tolerances.

In heuristic multi-agent systems the above characterisation is related to the concept of the stability of intermediate solutions, because the probability that any single agent will loose the best solution is often greater than zero. This may result in a lack of stability of the found solutions or in the worst case non-convergence of the algorithm. Thus for multi-agent systems it is desirable to characterise the stability of the discovered solutions. For example it is known that many variants of Genetic Algorithms do not converge and so the optimal solution may disappear from the next population. Consequently, in practice either an elitist selection mechanism is utilised or the information about the best solution has to be maintained throughout the entire evolution process.

In this paper we demonstrate that the solutions discovered by SDS are exceptionally stable and we illustrate some implications of the resource allocation mechanisms employed by SDS concerning the stability and robustness of the algorithm as well as its convergence behaviour.

In the next section we will introduce Stochastic Diffusion Search and we will briefly introduce the model of SDS on which the results of this paper are based. The following section will discuss the stability of SDS and contrast it with naive parallel random search. The final section will include discussion and conclusions.

2 Stochastic Diffusion Search

SDS is based on distributed computation, in which the operations of simple computational units, or agents are inherently probabilistic. Agents collectively construct the solution by performing independent searches followed by diffusion of information through the population. Positive feedback promotes better solutions by allocating to them more agents for their exploration. Limited resources induce strong competition from which the largest population of agents corresponding to the best-fit solution rapidly emerges.

In many search problems the solution can be thought of as composed of many subparts and in contrast to most Swarm Intelligence methods SDS explicitly utilises such decomposition to increase the search efficiency of individual agents. In SDS each agent poses a hypothesis about the possible solution and evaluates it partially. Successful agents repeatedly test their hypothesis while recruiting unsuccessful agents by direct communication. This creates a positive feedback mechanism ensuring rapid convergence of agents onto promising solutions in the space of all solutions. Regions of the solution space labelled by the presence of agent clusters can be interpreted as good candidate solutions. A global solution is thus constructed from the interaction of many simple, locally operating agents forming the largest cluster. Such a cluster is dynamic in nature, yet stable, analogous to, "*a forest whose contours do not change but whose individual trees do*", [2, 7]. The search mechanism is illustrated in the following analogy.

2.1 The restaurant game analogy

A group of delegates attends a long conference in an unfamiliar town. Each night they have to find somewhere to dine. There is a large choice of restaurants, each of which offers a large variety of meals. The problem the group faces is to find the best restaurant, that is the restaurant where the maximum number of delegates would enjoy dining. Even a parallel exhaustive search through the restaurant and meal combinations would take too long to accomplish. To solve the problem delegates decide to employ a Stochastic Diffusion Search.[3]

Each delegate acts as an agent maintaining a hypothesis identifying the best restaurant in town. Each night each delegate tests his hypothesis by dining there and randomly selecting one of the meals on offer. The next morning at breakfast every delegate who did not enjoy his meal the previous night, asks one randomly selected colleague to share his dinner impressions. If the experience was good, he also adopts this restaurant as his choice. Otherwise

[3] It should be emphasised that this analogy is provided simply to illustrate the communication and feedback mechanisms at the heart of a stochastic diffusion search, and **not** as a heuristic to be employed by a group of hungry delegates attending a midweek conference in Sicily.

he simply selects another restaurant at random from those listed in 'Yellow Pages'.

Using this strategy it is found that very rapidly significant number of delegates congregate around the best restaurant in town.

Abstracting from the above algorithmic process:

Initialisation phase
  whereby all agents (delegates) generate
  an initial hypothesis (restaurant)
loop
  Test phase
    Each agent evaluates evidence for its hypothesis
    (meal degustation). Agents divide into active
    (happy diners) and inactive (disgruntled diners).
  Diffusion phase
    Inactive agents adopt a new hypothesis by either
    communication with another agent (delegate) or,
    if the selected agent is also inactive, there is no
    information flow between the agents; instead the
    selecting agent must adopt a new hypothesis
    (restaurant) at random.
endloop

By iterating through test and diffusion phases agents stochastically explore the solution space. However, since tests succeed more often on good candidate solutions than in regions with irrelevant information, an individual agent will spend more time examining good regions, at the same time recruiting other agents, which in turn recruit even more agents. Candidate solutions are thus identified by concentrations of a substantial population of agents.

Central to the power of SDS is its ability to escape local minima. This is achieved by the probabilistic outcome of the partial hypothesis evaluation in combination with reallocation of resources (agents) via stochastic recruitment mechanisms. Partial hypothesis evaluation allows an agent to quickly form its opinion on the quality of the investigated solution without exhaustive testing (e.g. it can find the best restaurant in town without having to try all the meals available in each).

2.2 The model of SDS

In [18] a model of SDS was introduced which allowed for an analysis of the steady state behaviour of the system. The model captures the behaviour of individual agent as a Markov Chain evolving on a finite state space. The evolution of the entire system is then equivalent to the behaviour of an ensemble of coupled Markov Chains. It is possible to derive the steady state probability distribution of the entire ensemble [18] and the stability of the resource allocation presented in the next section will be based on this result.

For simplicity the discussion in this section is coached in terms of a simple template based search. However, the search and allocation mechanisms are generic and independent of the particular properties of the search space.

Consider a noiseless search space in which there exists a unique object with a non-zero overlap with the template - a desired solution. Assume that there is a total number of N agents and that upon a random choice of a feature for testing a hypothesis about the solution location, an agent may fail to recognise the best solution (a false negative) with a probability $p^- > 0$. In addition, a probability of locating the desired solution in a uniform random draw from the search space is p_m.

As an agent in SDS can be in one of two states, i.e. either active upon successful testing, or inactive otherwise, it may thus be represented as a Markov Chain, X_n, with a finite state space {*active, inactive*} with the transition probability matrix

$$P_n = \begin{pmatrix} 1-p^- & p^- \\ p_1^n & 1-p_1^n \end{pmatrix} \tag{1}$$

where

$$p_1^n = \frac{m}{N}(1-p^-) + (1-\frac{m}{N})p_m(1-p^-) \tag{2}$$

and m denotes number of active agents in the population at iteration n. The term p_1^n in (1) encapsulates the total probability that an inactive agent may become active either due to diffusion of information or due to random sampling of a new solution from the search space. Its dependence on the state of other chains in the population implies that this Markov Chain is not homogenous. As each of the agents can be modelled by such a Markov Chain, the entire SDS corresponds to the ensemble of such Markov Chains. The analysis of this ensemble Markov Chain model presented in [18] ascertains that the *steady state* probability distribution of the number of active agents, n, pointing to the solution is binomial,

$$\pi(n) = \binom{N}{n} \pi_1^n \pi_2^{N-n} \tag{3}$$

where

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

and $\pi_2 = 1 - \pi_1$.

From equations (3), (4) we can immediately estimate the steady state mean and variance of the number of active agents concentrated on the solution

$$E[n] = N\pi_1, Var[n] = N\pi_1\pi_2 \tag{5}$$

The characterisation of these quantities in terms of the search conditions can be found in [18]. In the following section we will address the implications of the resource allocation for the stability and convergence behaviour of SDS.

3 Resource Allocation

Because SDS behaves in the limit as an ensemble of identical ergodic Markov Chains [18] we can characterise its stability in terms of their steady state probability distribution. As the operation of agents is probabilistic in nature we expect that the activity of SDS will fluctuate around the steady state. However, the positive feedback utilised during diffusion in forming a cluster of agents corresponding to the best-fit solution will bring back the system from downwards fluctuations towards the equilibrium. This will not be possible however, if the number of agents pointing to the solution drops to zero. Thus, to characterise search stability we can use the probability of occurrence of the state with all agents being inactive.

For the ergodic Markov Chain, the mean return time to a state j is inversely proportional to the equilibrium probability of this state, [15]

$$m_j = \frac{1}{\pi(j)} \tag{6}$$

Consider Stochastic Diffusion Search with $N = 1000$ agents, a probability of locating the best instantiation of the target $p_m = 0.001$ and a probability of failing the test while pointing to the desired solution (false negative) $p^- = 0.2$. Equations (2)-(3) imply that $\pi_1 \approx 0.75$, $\pi_2 \approx 0.25$, $E[n] \approx 750$, $\sigma \approx 13.7$. Therefore, in the steady state the number of active agents pointing to the desired solution will be about 750. The fluctuations of the agent's number are well concentrated around the mean value with the standard deviation not exceeding 1.5% of the total number of agents. In the example considered, the mean return time to the state in which all agents are inactive is therefore

$$m_0 = \frac{1}{\pi(0)} \approx \frac{1}{(0.25)^{1000}} \propto 10^{602} \tag{7}$$

Thus indeed this state is on average visited extremely rarely and the search is very stable.

Consider the behaviour of Stochastic Diffusion Search with the same parameters characterising the search space but with smaller number of agents. Let $N = 10$, then $E[n] \approx 7.5$ and $\sigma \approx 1.7$, thus the variability of the number of active agents around the quasi equilibrium raised to about 17%. Also the mean return time to the state with all agents inactive decreased in this case

$$m_0 = \frac{1}{\pi(0)} \approx \frac{1}{(0.25)^{10}} \propto 10^{6} \tag{8}$$

Even though the number of agents participating in the search is rather small, the mean return to 'all-inactive' state is still relatively large. Also it is interesting to note that the stability of the search depends exponentially on the number of agents in SDS. In general, denoting the log of the mean return time

$$\tau = \log m_0 \tag{9}$$

and considering the ratio, r, of this quantity for searches using different numbers of agents, N_1, N_2

$$r = \frac{\tau_0^1}{\tau_0^2} = \frac{log\pi^1(0)}{log\pi^2(0)} = \frac{N_1 log\pi_2}{N_2 log\pi_2} = \frac{N_1}{N_2} \tag{10}$$

leads to the relation

$$N_1 = rN_2 \tag{11}$$

Thus, the ratio of logarithms of mean return times to the 'all-inactive' state is equal to the ratio of the numbers of agents involved in the search and in the above example $r = 100$. The above relationship between the size of the agent population and the return to the 'all inactive' state may allow for an efficient control of stability of solutions in practical applications.

To see the effect of positive feedback employed in formation of the largest active agents group consider for comparison a parallel random search without diffusion of information. Assume that one performs N independent uniformly random draws of potential solution positions from the search space and, upon each draw, checks a random feature at that position. The probability of recognising the best solution k times in such a procedure is given by

$$p[X = k] = \binom{N}{k} [p_m(1 - p^-)]^k [1 - p_m(1 - p^-)]^{N-k} \tag{12}$$

where the standard notation from SDS model is used.

For $N = 1000$, $p_m = 0.001$ and $p^- = 0.2$, one has

$$p[X = 0] \approx 0.449, \;\; p[X = 750] \propto 10^{-2081}$$

and for $N = 10$,

$$p[X = 0] \approx 0.992, \;\; p[X = 7] \propto 10^{-20}$$

Thus, in both cases, in completely random draws the probability of being completely unsuccessful is orders of magnitude higher than in SDS. This indicates that the diffusion of information in SDS changes radically the way the algorithm performs the search. The probability of the random parallel search being completely unsuccessful is very large whereas for SDS it is extremely small. On the contrary, the probability of obtaining a number of successful draws comparable to the largest cluster of active agents in SDS is very small, indicating that the solution proposed by Stochastic Diffusion Search is highly reliable.

4 Conclusions

SDS is a novel distributed probabilistic algorithm, performing the best-fit search. It is known to converge rapidly to the global optimum [19, 20]. We have demonstrated here that it also forms very robust representation of the desired solution. Although the algorithm operates by means of many simple agents posing independent hypotheses about the presence of the solution in the search space, the positive feedback resulting from the diffusion of information about potentially interesting positions very rapidly forms extremely stable but dynamic representations [7].

The neural architecture implementing standard SDS was proposed based on biologically inspired, novel model neurons operating as filters on the information encoded in the temporal structure of the spike trains [21]. Such network requires full connectivity between neurons as well as their synchronous operation. We have also investigated other architectures corresponding to versions of SDS relaxing these assumptions. In [10] we investigate variants of SDS based on introducing neighbourhood structure limiting interneuron connectivity. Asynchronous SDS, in which each neuron undergoes a cycle of operation independently of others was proposed in [9]. In both cases we observed analogous convergence properties as well as stability of solution representations. It follows that neither limited communication nor asynchronous operation impede the behaviour of SDS. Thus, these results, together with the theoretical analysis, suggest that SDS represents a very simple yet rapid and robust mode of information processing.

References

1. Aleksander I, Stonham TJ (1979) Computers & Digitial Techniques 2(1): 29-40
2. Arthur WB (1994) Amer Econ Rev (Papers and Proceedings) 84: 406
3. Back T (1996) Evolutionary Algorithms in Theory and Practice. Oxford University Press
4. Beattie PD, Bishop JM (1998) Journal of Intelligent and Robotic Systems 22: 255-267
5. Bishop JM (1989) Stochastic Searching Networks. In: IEE Conference Publication No. 313 Proc 1st IEE Int Conf Artificial Neural Networks. London
6. Bishop JM, Torr PH (1992) The Stochastic Search Network. In: Linggard R, Myers DJ, Nightingale C (eds) Neural Networks for Images, Speech and Natural Language. Chapman Hall, New York.
7. Bishop JM, Nasuto SJ, De Meyer K (2002) Knowledge Representation in Connectionist Systems. In: Dorronsoro JR (ed) Lecture Notes in Computer Science 2415, Springer, Berlin Heidelberg New York
8. Bonabeau E, Dorigo M, Theraulaz G (1999) Swarm Intelligence: from Natural to Artificial Systems. Oxford University Press, Oxford UK
9. De Meyer K (2000) Explorations in Stochastic Diffusion Search: soft- and hardware implementations of biologically inspired Spiking Neuron Stochastic Diffusion Networks. Technical Report KDM/JMB/2000-1, University of Reading, UK

10. De Meyer K, Bishop JM, Nasuto SJ (2002) Small World Network behaviour of Stochastic Diffusion Search. In: Dorronsoro JR (ed) Lecture Notes in Computer Science 2415, Springer, Berlin Heidelberg New York
11. De Meyer K, Nasuto SJ, Bishop, JM (2006) Stochastic Diffusion Optimisation: the application of partial function evaluation and stochastic recruitment in Swarm Intelligence optimisation, In: Abraham A, Grosam C, Ramos V (eds) Studies in Computational Intelligence (31): Stigmergic Optimization, Springer
12. Goldberg D (1989) Genetic Algorithms in search, optimization and machine learning. Addison Wesley, Reading MA
13. Grech-Cini E (1995) Locating facial features. PhD Thesis, University of Reading, Reading UK
14. Holland JH (1975) Adaptation in natural and artificial systems. The University of Michigan Press, Ann Arbor
15. Iosifescu M (1980) Finite Markov processes and their applications. Wiley, Chichester
16. Kennedy J, Eberhart RC, Shi Y (2001) Swarm Intelligence. Morgan Kauffman, San Francisco
17. Moglich, M., Maschwitz, U., Holldobler, B., (1974). Science 186 (4168): 1046-1047
18. Nasuto SJ (1999) Analysis of Resource Allocation of Stochastic Diffusion Search. PhD Thesis, University of Reading, Reading UK
19. Nasuto SJ, Bishop JM (1999) Journal of Parallel Algorithms and Applications 14: 89-107
20. Nasuto SJ, Bishop JM, Lauria S (1998) Time Complexity of Stochastic Diffusion Search. In: Heiss M (ed) Proceedings of the International ICSC / IFAC Symposium on Neural Computation. Vienna Austria
21. Nasuto SJ, Dautenhahn K, Bishop JM (1999) Communication as an Emergent Methaphor for Neuronal Operation. In: Nehaniv C (ed) Lecture Notes in Artificial Intelligence 1562. Springer, New York
22. Neumaier A (2004) Complete search in continuous global optimization and constraint satisfaction. In: Isereles A (ed) Acta Numerica 2004. Cambridge University Press, Cambridge UK
23. Whitaker RM, Hurley S (2002) An agent based approach to site selection for wireless networks. In: ACM Press Proc ACM Symposium on Applied Computing. Madrid Spain