23

Stochastic Diffusion Search: Modifications and Application

Mohammad Majid al-Rifaie

School of Computing and Mathematical Sciences, University of Greenwich
Old Royal Naval College, London, United Kingdom

J. Mark Bishop

Department of Computing
Goldsmiths, University of London, United Kingdom

CONTENTS

23.1 Introduction

Stochastic Diffusion Search (SDS) [1] introduced a new probabilistic approach for solving best-fit pattern recognition and matching problems. SDS, as a

multi-agent population-based global search and optimisation algorithm proposed in 1989, is a distributed mode of computation utilising interaction between simple agents [2].

SDS has been applied to a wide range of applications and problems, including but not limited to computer vision and medical imaging [3, 32, 5, 6, 7, 8] optimisation [9, 10, 11, 12] machine learning [13, 14, 15, 16, 17, 18] computational creativity [19, 20, 21, 22, 23] digital arts [24, 25, 26, 27, 28] and other theoretical or practical domains [29, 30, 31, 32, 33, 34, 35, 36, 37].

This paper first provides a brief description of the algorithm, and then a set of suggested variations and modifications is presented. These include an introduction to various recruitment strategies which govern the diffusion of information amongst the population. In addition to providing details about the initialisation of agents' hypotheses and their criteria for the algorithm termination, the concept of partial function evaluation is also expanded. The paper is then concluded with a real-world application of SDS for identifying metastasis in bone scans and how the algorithm's features are adopted to deal with this particular problem.

23.2 SDS algorithm

SDS algorithm commences a search or optimisation by initialising its population (e.g. agents). In any SDS search, each agent maintains a hypothesis, h, defining a possible problem solution. After initialisation two phases are followed (see the high-level description of SDS in Algorithm 25):

- Test Phase
- Diffusion Phase

In the Test Phase, SDS checks whether the agent hypothesis is successful or not by performing a *partial function evaluation, pFE*, which is some function of the agent's hypothesis; $pFE = f(h)$; subsequently, this evaluation returns a domain independent boolean value. Later in the iteration, contingent on the strategy employed, successful hypotheses diffuse across the population and in this way information on potentially good solutions spreads throughout the entire population of agents.

In the Diffusion Phase, each agent recruits another agent for interaction and potential communication of the hypothesis. Various flavours of the communication strategies which are deployed in the Diffusion Phase are explained in the next section.

Algorithm 25 SDS Algorithm.

```
Initialising agents()
While (stopping condition is not met)
  Testing hypotheses()
    Determining agents' activities (active/inactive)
  Diffusing hypotheses()
    Exchanging of information
End While
```

23.3 Further modifications and adjustments

In this section, further relevant details related to SDS are discussed, including different recruitment strategies, initialisation of agents, termination criteria and partial function evaluation.

In SDS, similar to other optimisation algorithms, the goal is finding the best solution based on the criteria specified in the objective function. The collection of all candidate solutions (hypotheses) forms the search space and each point in the search space is represented by an objective value, from which the objective function is formed. In the minimisation mode, for example, the lower the objective value is, the better the result.

One of the issues related to SDS is the mechanism behind allocating resources to ensure that while potential areas of the problem space are exploited, exploration is not ignored. For this purpose, different recruitment methods, where one agent recruits another one, are investigated:

23.3.1 Recruitment Strategies

Three recruitment strategies are proposed in [38]: active, passive and dual. These strategies are used in the Diffusion Phase of SDS. Each agent can be in either one of the following states: It is *active* if the agent is successful in the Test Phase; an agent is *inactive* if it is not successful; and it is *engaged* if it is invloved in a communication with another agent.

The standard SDS algorithm [1] uses the passive recruitment mode, which will be described next along with other recruitment modes.

23.3.1.1 Passive recruitment mode

In the *passive recruitment mode*, if the agent is not active, another agent is randomly selected and if the randomly selected agent is active, the hypothesis of the active agent is communicated (or *diffused*) to the inactive one. Otherwise a new random hypothesis is generated for the inactive agent and there will be no flow of information (see Algorithm 26).

Algorithm 26 Passive Recruitment Mode.

```
For ag = 1 to No_of_agents
   If ( !ag.activity() )
      r_ag = pick a random agent()
      If ( r_ag.activity() )
         ag.setHypothesis( r_ag.getHypothesis() )
      Else
         ag.setHypothesis( randomHypothesis() )
      End If/Else
   End If
End For
```

23.3.1.2 Active recruitment mode

In the *active recruitment mode*, active agents are in charge of communication with other agents. An active agent randomly selects another agent. If the randomly selected agent is neither active nor engaged in communication with another active agent, then the hypothesis of the active agent is communicated to the inactive one and the agent is flagged as engaged. The same process is repeated for the rest of the active agents. However if an agent is neither active nor engaged, a new random hypothesis is generated for it (see Algorithm 27).

Algorithm 27 Active Recruitment Mode.

```
For ag = 1 to No_of_agents
   If ( ag.activity() )
      r_ag = pick a random agent()
      If ( !r_ag.activity() AND !r_ag.getEngaged() )
         r_ag.setHypothesis( ag.getHypothesis() )
         r_ag.setEngaged(true)
      End If
   End If
End For

For ag = 1 to No_of_agents
   If ( !ag.activity() AND !ag.getEngaged() )
      ag.setHypothesis( randomHypothesis() )
   End If
End For
```

23.3.1.3 Dual recruitment mode

In *dual recruitment mode*, both active and inactive agents randomly select other agents. When an agent is active, another agent is randomly selected. If the randomly selected agent is neither active nor engaged, then the hypothesis of the active agent is shared with the inactive one and the inactive agent is flagged as engaged. Also, if there is an agent which is neither active nor engaged, it selects another agent randomly. If the newly selected agent is active, there will be a flow of information from the active agent to the inactive one and the inactive agent is flagged as engaged. Nevertheless, if there remains

an agent that is neither active nor engaged, a new random hypothesis is chosen for it.

Algorithm 28 Dual Recruitment Mode.

```
For ag = 1 to No_of_agents
   If ( ag.activity() )
      r_ag = pick a random agent()
      If ( !r_ag.activity() AND !r_ag.getEngaged() )
         r_ag.setHypothesis( ag.getHypothesis() )
         r_ag.setEngaged(true)
      End If
   Else
      r_ag = pick a random agent ()
      If ( r_ag . activity () AND ! ag . getEngaged () )
         ag . setHypothesis ( r_ag . getHypothesis () )
         ag . setEngaged ( true )
      End If
   End If/Else
End For

For ag = 1 to No_of_agents
   If ( !ag.activity() AND !ag.getEngaged() )
      ag.setHypothesis( randomHypothesis() )
   End If
End For
```

23.3.1.4 Context sensitive mechanism

Comparing the above-mentioned recruitment modes, it is theoretically determined in [38] that robustness and greediness decrease in the active recruitment mode. Conversely, these two properties are increased in dual recruitment strategy. Although, the greediness of dual recruitment mode results in decreasing the robustness of the algorithm, the use of *Context Sensitive Mechanism* limits this decrease [38, 39]. In other words, the use of context sensitive mechanism biases the search towards global exploration. In the context sensitive mechanism if an active agent randomly chooses another active agent that maintains the same hypothesis, the selecting agent is set inactive and adopts a random hypothesis. This mechanism frees up some of the resources in order to have a wider exploration throughout the search space as well as preventing cluster size from overgrowing, while ensuring the formation of large clusters in case there exists a perfect match or good sub-optimal solutions (see Algorithm 29).

Algorithm 29 Context Sensitive Mechanism.

```
If ( ag.activity() )
   r_ag = pick a random agent ()
   If ( r_ag.activity() AND
        ag.getHypothsis() == r_ag.getHypothsis() )
      ag.setActivity ( false )
      ag.setHypotheis ( randomHypothsis() )
   End If
End If
```

23.3.1.5 Context free mechanism

In *Context Free Mechanism,* which is another recruitment mechanism, the performance is similar to context sensitive mechanism, where each active agent randomly chooses another agent. However, if the selected agent is active (irrespective of having the same hypothesis or not), the selecting agent becomes inactive and picks a new random hypothesis. By the same token, this mechanism ensures that even if one or more good solutions exist, about half of the agents explore the problem space and investigate other possible solutions (see Algorithm 30).

Algorithm 30 Context Free Mechanism.

```
If ( ag.activity() )
   r_ag = pick a random agent ()
   If ( r_ag.activity() )
      ag.setActivity ( false )
      ag.setHypotheis ( randomHypothsis() )
   End If
End If
```

23.3.2 Initialisation and termination

Although normally agents are uniformly distributed throughout the search space, if the search space is of a specific type, or knowledge exists about it *a priori*, it is possible to use a more intelligent (than uniform random distribution) startup by biasing the initialisation of the agents.

If there is a pre-defined pattern to find in the search space, the goal will be locating the best match or, if it does not exist, its best instantiation in the search space [40]. Similarly, in a situation which lacks a pre-defined pattern, the goal will be finding the best pattern in accordance with the objective function.

In both cases, it is necessary to have a termination strategy. In one method[1], SDS terminates the process when a statistical equilibrium state is reached, which means that the threshold of the number of active agents is exceeded and the population maintained the same state for a specified number of iterations. In [41], four broad types of halting criteria are introduced:

1. No stopping criterion, where the user interrupts the course of action of the search or optimisation and is usually preferred when dealing with *dynamically changing* problem spaces or when there is no predefined pattern to look for
2. Time-based criterion, in which passing a pre-set duration of time is the termination point of the algorithm
3. Activity-based criterion, which is a problem-dependent halting criterion, and it is the most prevalent form in the SDS algorithm.

[1] Ibid

The termination of the process is decided upon through monitoring the overall activity of the agents (e.g. reaching a certain user defined activity level, reaching a stable population state after a sudden increase in their activities)

4. Cluster-based criterion that keeps track of the formation of stable clusters.

Determining the termination criteria without having a fixed a priori threshold as a prerequisite is a possible approach (e.g. Quorum sensing, which is a system of stimulus and response correlated to population density, is used in some social insects in search for nest sites or many species of bacteria to coordinate gene expression based on the density of their local population [42]). By the same analogy and as stated in [41], it is possible to implement the termination criterion as a random sampling process: for example, a cluster-based termination procedure may consider monitoring the hypotheses of a subset of the population until the same hypothesis is encountered more than once. This provides partial evidence of the formation of a cluster. The size of the sample taken from the population can then be increased.

The two most common termination strategies in SDS (introduced in [40]) are the following:

- Weak halting criterion is the ratio of the active agents to the total number of agents. In this criterion, cluster sizes are not the main concern.
- Strong halting criterion investigates the number of active agents that form the largest cluster of agents all adopting the same hypothesis.

Therefore, the choice of the halting mechanism is based on whether to favour the active agents in the whole of the agent populations (weak halting mechanism), which is similar to the activity-based criterion, or to consider the largest cluster of active agents (strong halting mechanism), which is similar to the cluster-based criterion.

23.3.3 Partial function evaluation

One of the concerns associated with many optimisation algorithms (e.g. Genetic Algorithm [43], Particle Swarm Optimisation [44] etc.) is the repetitive evaluation of computationally expensive fitness functions. In some applications, such as tracking a rapidly moving object, the repetitive function evaluation significantly increases the computational cost of the algorithm. Therefore, in addition to reducing the number of function evaluations, other measures should be taken in order to reduce the computations carried out during the evaluation of each possible solution as part of the optimisation or search processes.

The commonly used benchmarks for evaluating the performance of swarm intelligence algorithms are typically small in terms of their objective functions computational costs [45, 46], which is often not the case in real-world

applications. Examples of costly evaluation functions are seismic data interpretation [46], selection of sites for the transmission infrastructure of wireless communication networks and radio wave propagation calculations of one site [37] etc.

Costly functions have been investigated under different conditions [47] and the following two broad approaches have been proposed to reduce the cost of function evaluations:

- The first is to estimate the fitness by taking into account the fitness of the neighbouring elements, the former generations or the fitness of the same element through statistical techniques introduced in [48, 49].
- In the second approach, the costly fitness function is substituted with a cheaper, approximate fitness function.

When agents are about to converge, the original fitness function can be used for evaluation to check the validity of the convergence [47].

Many fitness functions are decomposable to components that can be evaluated separately. In partial evaluation of the fitness function in SDS, the evaluation of one or more of the components may provide partial information and means for guiding the optimisation.

The partial function evaluation of SDS allows the algorithm to absorb certain types of noise in the objective function without affecting the convergence time or the size and stability of the clusters.

Additionally, noise, which does not alter the averaged probabilities of the test score (probability of producing active agents during the test phase, averaged over all component functions) but increases the variance in the evaluation of component functions, has no effect on the resource allocation process of SDS [50]. However, if the value of the test score changes as a result of noise presence, the resource allocation process may be influenced either:

- positively if the value of the test score increases.
- or negatively if the value of the test score decreases.

23.4 Application: Identifying metastasis in bone scans

This section discusses the application of SDS as a tool to identify metastasis in bone scans. This work has been originally published in [6, 7]. SDS is adapted for this particular purpose and its performance is investigated by running SDS agents on sample bone scans (see Fig. 23.1-Top) whose statuses have been determined by the experts.

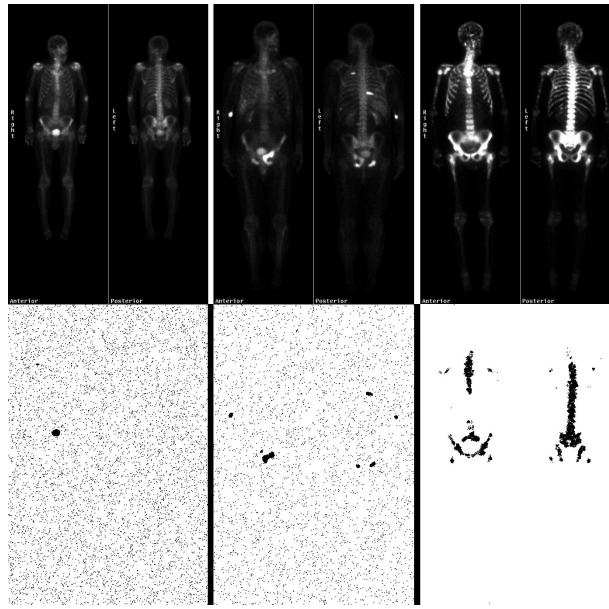

FIGURE 23.1
Bone Scans. Top: Typically 2-6 hours after intravenous administration of technetium-99m-labelled diphosphonates. Brighter areas indicate a higher radiotracer uptake. Bottom: The scans are processed using Stochastic Diffusion Search algorithm. Left: Healthy; middle: partially affected; right: metastatic disease spread.

23.4.1 Experiment setup

Each scan, as shown in Fig. 23.1 (Top) are processed by the SDS agents which are responsible for locating the affected area(s). According to the description provided by the experts, Fig. 23.1 (Top-middle and right) are the areas of metastasis.

The reproducibility and the accuracy of the SDS algorithm can be utilised in developing a standardised system to interpret the bone scans preventing operator errors and discrepancies. This technology can be employed as an adjunct by radiologists to assess the various parts of the bone scan making the diagnosis of the lesions more thorough and less time consuming. Additionally this technique can be effectively used to develop programs for teaching and training medical students and junior doctors.

The number of agents used in this experiment is 10,000 and the algorithm is run for 10 iterations (i.e. 10 cycles of test and diffusion phases). At the beginning of the process, all the agents are initialised randomly throughout the search space. In other words, each agent randomly picks a pixel from the image of the bone scan (i.e. one pixel from 460×690). During the test phase of SDS

o	o	o
o	x	o
o	o	o

FIGURE 23.2
Agent's Neighbours in Test Phase. The symbol x represents the position of the agent and the o's represent the neighbours used during the test phase.

o	o	o	o	o
o	o	o	o	o
o	o	x	o	o
o	o	o	o	o
o	o	o	o	o

FIGURE 23.3
Diffusion Area. The symbol x represents the position of the active agent and the o's represent the accessible places during the diffusion phase.

algorithm, each agent's status should be determined. The method used here to set the activity of the agents is to find the average of the colour intensity[2] ($avgIn$) of each agent and its neighbours (see Fig. 23.2). If $avgIn > 180$ the agent is flagged active, otherwise inactive[3].

Note that in this application, standard SDS which uses passive recruitment mode, is employed.

During the diffusion phase, each inactive agent randomly selects another agent from the population; if the selected agent is active, the selecting agent adopts the hypothesis (i.e. location) of the active agent and the information sharing takes place. The strategy used for information sharing is to randomly pick an area surrounding the active agent (see Fig. 23.3). Active agents also check their position by continuously picking a random pixel in the neighbourhood; this way, an area which does not have a good enough potential is discarded from one iteration to the next.

23.4.2 Results

As shown in Fig. 23.1 (Bottom), areas with higher potential of metastasis are identified. Other than urinary bladder activity, faint renal activity, and minimal soft-tissue activity which are normally present in the scan (Fig. 23.1 Bottom-left), the existence of multiple, randomly distributed areas of increased uptake of varying size, shape, and intensity are highly suggestive

[2] Colour intensity signifies the brightness of each pixel, which is a spectrum between 0 and 255.

[3] The value of 180 is problem-dependent and could be adjusted to increase or decrease the sensitivity of the system.

of bone metastases (Fig. 23.1 Bottom-middle). Additionally as stated before, when the metastatic process is distributed, almost all of the radiotracer congregates in the skeleton, with little or no activity in the soft tissues or urinary tract (Fig. 23.1 Bottom-right).

In order to show the behaviour of the algorithm in each iteration, the position of each agent during the process of metastasis identification is illustrated in Fig. 23.4. In this figure, the three original bone scans referred to earlier are used as input to the system, and as the figure shows, successful agents diffuse their positions across the population and this way, information on potentially good solutions spreads throughout the entire population of agents. This process is caused through the recruitment strategy, where each agent recruits another agent for interaction and potential communication of the promising areas.

In order to decide whether the activity of the agents when presented with different types of bone scans (e.g. not affected, affected and highly affected), would be a distinctive indicator, a statisical analysis is performed. TukeyHSD Test [51] is used to highlight whether there is a significant difference between the activity of the agents when processing the bone scans. Table 23.1 (a) shows the activity rate of the populations over each iteration. Three different samples are used for this analysis: Samples 1, 2 and 3 refer to the scans

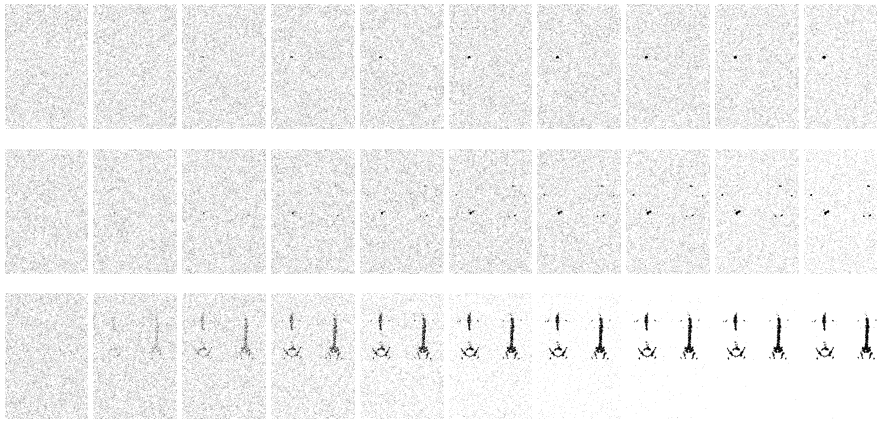

FIGURE 23.4
SDS algorithm processing bone scans in 10 iterations. Each row shows the behaviour of the agents when presented with one bone scan. Each bone scan is processed by 10,000 agents (illustrated as black dots) and through communication, agents explore different areas of the bone scan to identify potential areas of metastasis. The leftmost figures in each row show the location of the agents on the first iteration, and the rightmost ones represent the last iteration. Top: Healthy; middle: partially affected; bottom: metastatic disease spread.

TABLE 23.1
Activity Status of Agents.

Itr	Sample 1	Sample 2	Sample 3
0	0±0	0±0	0±0
1	5±2	17±4	277±16
2	15±4	47±9	763±37
3	33±8	100±18	1602±76
4	66±18	201±31	2991±137
5	129±33	379±51	4992±188
6	245±62	697±84	7260±198
7	461±110	1250±141	8947±123
8	852±201	2201±230	9583±51
9	1557±351	3650±330	9708±22

(a) Mean ±standard deviation of the number of active agents in each iteration is shown (rounded to the nearest number).

Itr	s1 − s2	s1 − s3	s2 − s3
0	–	–	–
1	o – X	o – X	o – X
2	o – X	o – X	o – X
3	o – X	o – X	o – X
4	o – X	o – X	o – X
5	o – X	o – X	o – X
6	o – X	o – X	o – X
7	o – X	o – X	o – X
8	o – X	o – X	o – X
9	o – X	o – X	o – X

(b) Based on TukeyHSD Test, if the difference between each pair of samples is significant, the pairs are marked (o – X shows that the right sample has significantly more active agents than the left one). This test uses 95% family-wise confidence level. The aim is to show that agents dealing with scans which have different levels of metastasis exhibit significantly different behaviour.

in Fig. 23.1 (left to right). Table 23.1 (b) shows that other than the first iteration where the agents are just initialised, different bone scans would result in significantly different activity rates. This could be used as an indicator, highlighting the difference between various scans and whether they are healthy, partially affected or the metastasis is spread.

23.4.3 Concluding remarks

The results of applying SDS to detect areas of metastasis in this experiment demonstrate the suitability of the approach. A statistical analysis further

investigates the behaviour of the agents in the population and the outcome highlights that the algorithm exhibits a statistically significant difference when applied to bone scans for healthy, partially affected or heavily affected individuals. Finally, it is important to note that the presented technique could be effectively utilised as an adjunct to the experts' eyes of a specialist.

23.5 Conclusion

This paper provides the principal topics on the standard SDS algorithm followed by details of various flavours to SDS; these include, recruitment strategies, initialisation of agents and termination of the process as well as partial function evaluation. The paper then provides a real-world application of SDS where it is used to identify metastasis in bone scans. In this medical imaging problem, a set of clinically pre-determined bone scans is presented to SDS algorithm which are then processed and the outcomes are analysed. While this application is designed to be an adjunct to a clinician, the process can be expanded and further complexity can be added to identify different types of tissues and affected organs.

References

1. J.M. Bishop. "Stochastic Searching Networks" in *Proc. of 1st IEE Conf. on Artificial Neural Networks*, pp. 329-331, 1989.
2. K. de-Meyer, J.M. Bishop, S.J. Nasuto. *Stochastic diffusion: Using recruitment for search* in Proc. of Symposium on Evolvability and Interaction, pp. 60-65, The University of London, UK, 2003.
3. J.M. Bishop, P. Torr. "The Stochastic Search Network" in *Neural Networks for Images, Speech and Natural Language*, pp. 370-387, Chapman & Hall, New York, 1992.
4. E. Grech-Cini. "Locating Facial Features". PhD Thesis, University of Reading, Reading, UK, 1995.
5. M.M. al-Rifaie, A. Aber, D.J. Hemanth. "Deploying swarm intelligence in medical imaging identifying metastasis, micro-calcifications and brain image segmentation". *Systems Biology, IET*, vol. 9(6), pp. 234-244, 2015.
6. M.M. al-Rifaie, A. Aber. "Identifying Metastasis in Bone Scans with Stochastic Diffusion Search" in *Information Technology in Medicine and Education (ITME)*, 2012.

7. M.M. al-Rifaie, A. Aber, A.M. Oudah. "Utilising Stochastic Diffusion Search to identify metastasis in bone scans and microcalcications on mammographs" in *Proc. of IEEE International Conference on Bioinformatics and Biomedicine Workshops*, pp. 280-287, 2012.
8. M.M. al-Rifaie, A. Aber, A.M. Oudah. "Ants intelligence framework; identifying traces of cancer" in The House of Commons, UK Parliment. SET for BRITAIN 2013. Poster exhibitions in Biological and Biomedical Science, 2013.
9. M.M. al-Rifaie, M.J. Bishop, T. Blackwell. "An investigation into the merger of stochastic diffusion search and particle swarm optimisation" in *Proc. of the 13th Annual Conference on Genetic and Evolutionary Computation, GECCO 2011*, pp. 37-44, 2011.
10. M.M. al-Rifaie, J.M. Bishop, T. Blackwell. "Information sharing impact of stochastic diffusion search on differential evolution algorithm". *Memetic Computing*, vol. 4(4), pp. 327-338, 2012.
11. M.M. al-Rifaie, M. Bishop, T. Blackwell. "Resource Allocation and Dispensation Impact of Stochastic Diffusion Search on Differential Evolution Algorithm" in *Nature Inspired Cooperative Strategies for Optimization* (NICSO 2011). Studies in Computational Intelligence, vol. 387, pp. 21-40, Springer, 2012.
12. M.G.H. Omran, A. Salman. "Probabilistic stochastic diffusion search" in *Swarm Intelligence*, pp. 300-307, Springer, 2012.
13. M.M. al-Rifaie, M. Y.-K. Matthew and M. d'Inverno. "Investigating Swarm Intelligence for Performance Prediction" in *Proc. of the 9th International Conference on Educational Data Mining*, pp. 264-269, 2016.
14. H. Alhakbani, M.M. al-Rifaie. "Feature Selection Using Stochastic Diffusion Search" in *Proc. of the Genetic and Evolutionary Computation Conference*, pp. 385-392, 2017.
15. M.M. al-Rifaie, D. Joyce, S. Shergill, M. Bishop. "Investigating stochastic diffusion search in data clustering" in *SAI Intelligent Systems Conference (IntelliSys)*, pp. 187-194, 2015.
16. H.A. Alhakbani, M.M. al-Rifaie. "Exploring Feature-Level Duplications on Imbalanced Data Using Stochastic Diffusion Search" in *Multi-Agent Systems and Agreement Technologies*, pp. 305-313, Springer, 2016.
17. I. Aleksander, T.J. Stonham. "Computers and Digital Techniques 2(1)" in *Lecture Notes in Artificial Intelligence*, vol. 1562, pp. 29-40, Springer, 1979.
18. H.A. Alhakbani, M.M. al-Rifaie. "A Swarm Intelligence Approach in Undersampling Majority Class" in *Proc. of 10th International Conference, ANTS 2016*, pp. 225-232, 2016.

19. M.M. al-Rifaie, J.M. Bishop, S. Caines. "Creativity and autonomy in swarm intelligence systems". *Cognitive Computation*, vol. 4(3), pp. 320-331, 2012.
20. M.M. al-Rifaie, J.M. Bishop. "Weak and Strong Computational Creativity" in *Computational Creativity Research: Towards Creative Machines. Atlantis Thinking Machines*, vol. 7, pp. 37-49, Atlantis Press, 2015.
21. M.M. al-Rifaie, F.F. Leymarie, W. Latham, M. Bishop. "Swarmic autopoiesis and computational creativity". *Connection Science*, vol. 29(4), pp. 276-294, 2017.
22. J.M. Bishop, M.M. al-Rifaie. "Autopoiesis, creativity and dance". *Connection Science*, vol. 29(1), pp. 21-35, 2017.
23. M.M. al-Rifaie, A. Cropley, D. Cropley, M. Bishop. "On evil and computational creativity". *Connection Science*, vol. 28(1), pp. 171-193, 2016.
24. M.M. al-Rifaie, A. Aber, M. Bishop. "Cooperation of Nature and Physiologically Inspired Mechanisms in Visualisation" in *Biologically-Inspired Computing for the Arts: Scientific Data through Graphics*, IGI Global, United States, 2012.
25. M.M. al-Rifaie, M. Bishop. "Swarmic Paintings and Colour Attention". *Lecture Notes in Computer Science*, vol. 7834, pp. 97-108, Springer, 2013.
26. M.M. al-Rifaie, M. Bishop. "Swarmic Sketches and Attention Mechanism". *Lecture Notes in Computer Science*, vol. 7834, pp. 85-96, Springer, 2013.
27. A.M. al-Rifaie, M.M. al-Rifaie. "Generative Music with Stochastic Diffusion Search". *Lecture Notes in Computer Science*, vol. 9027, pp. 1-14, Springer, 2015.
28. M.M. al-Rifaie, W. Latham, F.F. Leymarie, M. Bishop. "Swarmic Autopoiesis: Decoding de Kooning" in *Proc. of 3rd Symposium on Computational Creativity, AISB 2016*, 2016.
29. M.A.J. Javid, M.M. al-Rifaie, R. Zimmer. "Detecting Symmetry in Cellular Automata Generated Patterns Using Swarm Intelligence" in *Proc. of 3rd International Conference on the Theory and Practice of Natural Computing (TPNC 2014)*, pp. 83-94, 2014.
30. F.M. al-Rifaie, M.M. al-Rifaie. "Investigating Stochastic Diffusion Search in DNA Sequence Assembly Problem" in *Proc. of SAI Intelligent Systems Conference (IntelliSys)*, pp. 625-631, 2015.
31. M.A.J. Javid, W. Alghamdi, A. Ursyn, R. Zimmer, M.M. al-Rifaie. "Swarmic approach for symmetry detection of cellular automata behaviour". *Soft Computing*, vol. 21(19), pp. 5585-5599, 2017.

32. E. Grech-Cini. "Locating Facial Features". PhD Thesis, University of Reading, Reading, UK, 1995.

33. P.D. Beattie, J.M. Bishop. "Self-localisation in the SENARIO autonomous wheelchair". *Journal of Intellingent and Robotic Systems*, vol. 22, pp. 255-267, 1998.

34. A.K. Nircan. "Stochastic Diffusion Search and Voting Methods". PhD Thesis, Bogaziki University, 2006.

35. K. de-Meyer, M. Bishop, S. Nasuto. "Small World Effects in Lattice Stochastic Diffusion Search". *Lecture Notes in Computer Science*, vol. 2415, pp. 147-152, Springer, 2002.

36. T. Tanay, J.M. Bishop, S.J. Nasuto, E.B. Roesch, M.C. Spencer. "Stochastic Diffusion Search Applied to Trees: a Swarm Intelligence Heuristic Performing Monte-Carlo Tree Search" in *Proc. of AISB 2013*, 2013.

37. R.M. Whitaker, S. Hurley. "An Agent Based Approach to Site Selection for Wireless Networks" in *Proc. of 1st IEE Conf. on Artificial Neural Networks*, 2002.

38. D.R. Myatt, S.J. Nasuto, J.M. Bishop. *Artificial Life X Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems*, Bloomington, USA, 2006.

39. S.J. Nasuto. "Resource Allocation Analysis of the Stochastic Diffusion Search". PhD Thesis, University of Reading, Reading, UK.

40. S.J. Nasuto, J.M. Bishop. "Convergence analysis of stochastic diffusion search". *Parallel Algorithms and Applications*, vol. 14(2), 1999.

41. K. de-Meyer. "Foundations of Stochastic Diffusion Search". PhD Thesis, University of Reading, Reading, UK, 2003.

42. M.B. Miller, B.L. Bassler. "Quorum sensing in bacteria". *Annual Reviews in Microbiology*, vol. 55(1), pp. 165-199, 2001.

43. D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Longman Publishing Co., 1989.

44. J. Kennedy, R.C. Eberhart. "Particle swarm optimization" in *Proc. of the IEEE International Conference on Neural Networks*, pp. 1942-1948, 1995.

45. J. Digalakis, K. Margaritis. "An experimental study of benchmarking functions for evolutionary algorithms". *Int. J. Comput. Math.* vol. 79, pp. 403-416, 2002.

46. D. Whitley, S. Rana, J. Dzubera, K.E. Mathias. "Evaluating evolutionary algorithms". *Artificial Intelligence*, vol. 85(1-2), pp. 245-276, 1996.

47. Y. Jin. "A comprehensive survey of fitness approximation in evolutionary computation". *Soft Computing*, vol. 9, pp. 3-12, 2005.

48. J. Branke, C. Schmidt, H. Schmeck. "Efficient Fitness Estimation in Noisy Environments" in *Proc. of Genetic and Evolutionary Computation Conference*, 2001.

49. M.A. el-Beltagy, A. J. Keane. "Evolutionary Optimization for Computationally Expensive Problems using Gaussian Processes" in *Proc. of Int. Conf. on Artificial Intelligence'01*, pp. 708-714, 2001.

50. K. de-Meyer, S.J. Nasuto, J.M. Bishop. "Stochastic Diffusion Optimisation: the Application of Partial Function Evaluation and Stochastic recruitment in Swarm Intelligence Optimisation" in *Swarm Intelligence and Data Mining*, Springer, 2006.

51. R.G. Miller. *Simultaneous Statistical Inference*, Springer, 1981.