Application of Chaos in Stochastic Diffusion Search

Santhanakrishnan Ramani

***Abstract*— Stochastic Diffusion Search(SDS) is one of the robust and popular heuristic algorithms used for global optimization. It is an agent-based probabilistic global search and optimization technique best suited to problems where the objective function can be decomposed into multiple independent partial-functions. In this paper, I would like to explore the effect of chaos theory in SDS. The effect is studied by using three different applications where SDS is used namely text search, global numerical optimization, and detection of metastasis in the bone scan. The results prove that use of chaos in SDS has some improvement over the existing algorithm and will be an interesting area to explore more in the near future.**

I. INTRODUCTION

Global optimization is a branch of applied mathematics and numerical analysis that deals with the global optimization of a function or a set of functions according to some criteria. In order to search for the global optimum various Swarm Intelligence (collective behavior of decentralized, self-organized systems, natural or artificial) heuristic algorithms namely Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Stochastic Diffusion Search (SDS) are used. In recent years, chaos theory has been used in conjunction with many of these heuristics to prevent them from getting stuck at the local optima and to speed up the rate of convergence to the global optimum [5], [6].

II. STOCHASTIC DIFFUSION SEARCH

SDS came into existence in 1989 as a population-based, pattern-matching algorithm [1]. Unlike stigmergic communication employed in Ant Colony Optimization, which is based on modification of the physical properties of a simulated environment, SDS uses a form of direct (one-to-one) communication between the agents similar to the tandem calling mechanism employed by one species of ants, *Leptothorax Acervorum* [7].

A. Algorithm

It consists of four phases namely initialize, test, diffusion, and convergence. In the initialization phase, each agent is assigned a possible hypothesis and stochastically employed in the search space. In the test phase, all agents evaluate the partial objective function according to their current hypothesis and the result of this evaluation is binary making them active or inactive. In the diffusion phase, the inactive agent's, A randomly picks another agent, B if he is active, A copies B's hypothesis, else randomly selects another hypothesis over the entire search space. The test and diffusion phase keep repeating until the termination criteria are reached.

B. Analogy - The Restaurant Game

A group of delegates arrive in a foreign town and wants to find the best place where everyone would enjoy dining. The search space is set of all restaurants. Even a parallel exhaustive search through the restaurant and meal combinations would take a long time to accomplish. So, they decide to employ SDS. The hypothesis is to find the best restaurant in the town. Each night a delegate chooses a random restaurant and tests his hypothesis by selecting randomly 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, he simply selects another restaurant at random from those listed in 'Yellow Pages'. Using this strategy it is found that very rapidly a significant number of delegates congregate around the 'best' restaurant in town.

C. Applications

SDS has been applied to many diverse search and optimization problems such as site selection for wireless networks [4], sequence detection in Bioinformatics, mobile robot self-localization, object recognition, eye tracking, lip tracking and text search [1]. In recent years, SDS has been used in integration with other swarm intelligence techniques and has provided statistically significant result compared to the technique alone.

III. WHY CHAOS ?

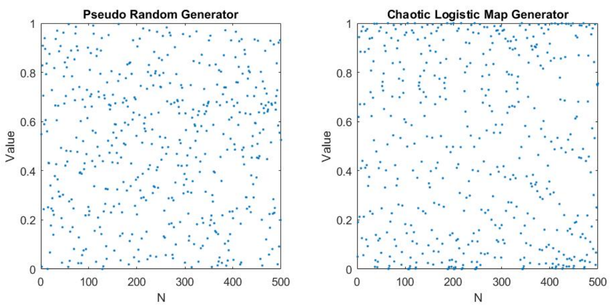

Fig. 1. Chaos-Logistic Vs Pseudo random number sequence

The figure 1 shows the plot of 500 random numbers generated using chaotic logistic map and pseudo random generator. The chaotic Logistic map was generated by setting $R = 4$ and some random value between [0,1] for x_0 in the equation shown below.

$$x_{n+1} = Rx_n(1 - x_n)$$

It can be clearly seen that the value generated by pseudo random generator lies between [0.3,0.8], and doesn't cover the whole value domain, whereas the one generated using Chaotic Logistic Map performs better.

A. Properties

The properties of chaotic sequence like Ergodicity, Sensitive Dependence on Initial Conditions (SDOIC), Non-Periodic, Non-Converging, Bounded makes it a more interesting candidate. Ergodicity ensures chaotic variables to traverse all state non-repeatedly within a certain range according to its own laws. SDOIC ensures that there are not two identical new populations and these populations are deterministic and reproducible.

B. Inspiration

Recently, chaotic sequences have been adopted instead of random sequences and very interesting and somewhat good results have been shown in many applications like Secure Transmission, Nonlinear Circuits, DNA Computing, Image Processing [5].

IV. EXPERIMENT SETUP

Tested the effect of chaos on SDS by implementing the following three applications which make use of Stochastic Diffusion Search. The places wherever random generation was used, I used the Chaotic Logistic Map stated above to generate values.

A. Text Search

The main objective is to find whether a string exists in the text. A quick demonstration of how to apply SDS for finding the string as explained in [11]. Let us assume we need to find a 3 letter pattern in a string of length 16. An example Model and Search Space is shown below in figure 2. The number of agents used was 4.

Model

Index:	0	1	2
Model:	S	I	B

Search Space

Index:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Search Space	X	Z	A	V	M	Z	S	I	B	V	G	O	L	B	E	H

Fig. 2. Model and Search Space

In the initialization phase the agents are assigned position in the text chaotically. In the test phase the agents hypothesis will be the set of 3 letters starting from the position they where initialized and they test by comparing the letter at one of the three positions with corresponding letter present in that position in model. If both the letters are same then the agent is made active and inactive otherwise. In the diffusion phase, if the inactive agent, A communicates with a active agent, B then they copy the hypothesis of the active agent, else a new hypothesis is assigned chaotically. The figure 3 below gives a picture what happens during iteration 1.

Agent No:	1	2	3	4
Hypothesis position:	12 L-B-E	9 V-G-O	2 A-V-M	3 V-M-Z
Letter picked:	1^{st}	2^{nd}	1^{st}	3^{rd}
Status:	×	×	×	×

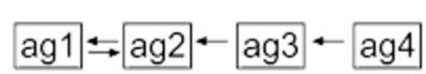

Fig. 3. Iteration 1

The process continues until all agents become active. The figures 4, 5, 6 represents what happens during the respective intervals until convergence.

Agent No:	1	2	3	4
Hypothesis position:	6 S-I-B	10 G-O-L	0 X-Z-A	5 Z-S-I
Letter picked:	2^{nd}	3^{rd}	1^{st}	1^{st}
Status:	√	×	×	×

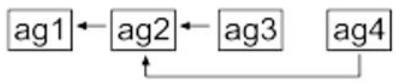

Fig. 4. Iteration 2

Agent No:	1	2	3	4
Hypothesis position:	6 S-I-B	6 S-I-B	1 Z-A-V	4 M-Z-S
Letter picked:	3^{rd}	1^{st}	2^{nd}	3^{rd}
Status:	√	√	×	×

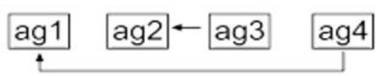

Fig. 5. Iteration 3

Agent No:	1	2	3	4
Hypothesis position:	6 S-I-B	6 S-I-B	6 S-I-B	6 S-I-B
Letter picked:	1^{st}	2^{nd}	3^{rd}	1^{st}
Status:	√	√	√	√

Fig. 6. Iteration 4

Ran the same experiment using a model of length 5, search space of length 445 with 5 agents.

B. Bone Metastasis Detection

The main objective is to find the areas of metastasis in bone. Bone metastasis occurs when cancer cells from the primary tumor, such as prostate, breast, and lung cancer, spread to the bone. For experimenting I took the images of healthy, mildly affected and metastasis affected bone scans which were taken typically 2-6 hours after intravenous administration of technetium-99m-labeled diphosphonates, where brighter areas in the bone scan indicated metastasis [8]. Then applied Chaotic Diffusion Search using 10,000 agents and ran the test and diffusion phase for 10 iterations.

In the initialization phase, the agents were assigned a pixel position chaotically. In the test phase, if the average intensity of the agents neighbor pixel values is >180, the agents were set active and inactive otherwise. And in the diffusion phase, if an inactive agent communicates with an active agent, it takes the value of one of the neighbor pixel as its new position, else chaotically assigned a new position and the next iteration starts.

C. Global Numerical Optimization

The main objective is to find the global optimal value (min or max) for the given function using SDS. The main reason for using chaos in SDS is to prevent the algorithm from getting stuck in the local optima and speed up the convergence rate when there are numerous local optima. In order to experiment the effect of chaos on finding the global optimum value, used the Rosenbrock function shown below in figure 7 as the benchmark function.

$$f(x_1 \cdots x_n) = \sum_{i=1}^{n-1} (100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2)$$

$$-2.048 \leq x_i \leq 2.048$$

$$\text{minimum at } f(1, 1, \cdots, 1) = 0$$

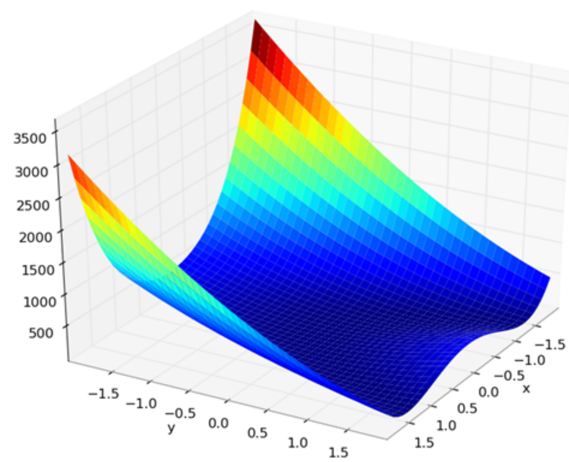

Fig. 7. 2D Rosenbrock Function

In the initialization phase, agents are allocated chaotically in the search space, and in the test phase agents evaluate the function using their current hypothesis and are set active, if the function value is less than the function value of an agent selected chaotically, else inactive. In the diffusion phase inactive agents, A pick random agents, B and share their hypothesis if they are active, else are chaotically assigned a new hypothesis. The process continues until 90% of the agents become active and cluster around an area and a local optimization can be used to get a better accuracy in that area.

V. RESULT

A. *Text Search*

The table I shows the time taken and the number of iterations for the SDS to converge before the pattern is found in the given string averaged over 100 iterations. We could see that using Chaos instead of random in SDS gives better results.

TABLE I
RESULTS OF TEXT SEARCH USING SDS

Method	Time Taken (in secs)
Random	0.01767
Chaotic	0.01656

Method	No. of iterations
Random	249
Chaotic	201

B. *Bone Metastasis Detection*

The table II shows the number of active agents every iteration for all the three bone scans, and it can be clearly seen that the number of agents active is statistically significant as it is low for Healthy, somewhat high for Mildly affected and very high for the one with Metastasis, which one could leverage as a measure to identify the ones that have Metastasis. The Figure 8, 9, 10 shows the clustering of active agents around the affected areas for the three different bone scans after 10 iterations of SDS.

TABLE II
RESULTS OF BONE METASTASIS DETECTION

No of Active Agents			
Iteration	Healthy	Mild	Metastasis
0	0	0	0
1	5	14	258
2	14	40	706
3	26	83	1538
4	52	144	2749
5	83	245	4500
6	142	408	6629
7	276	687	8473
8	478	1164	9520
9	806	1849	9881

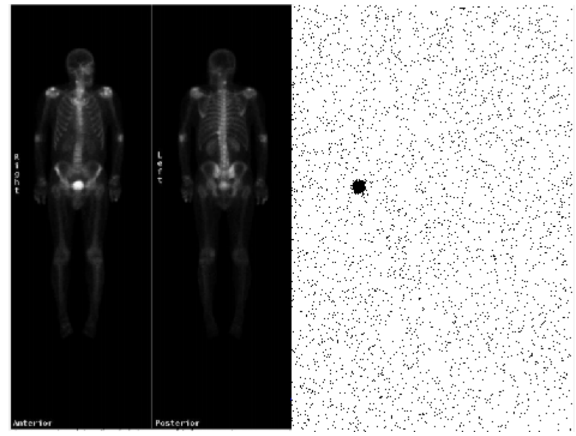

Fig. 8. Healthy Bone Scan

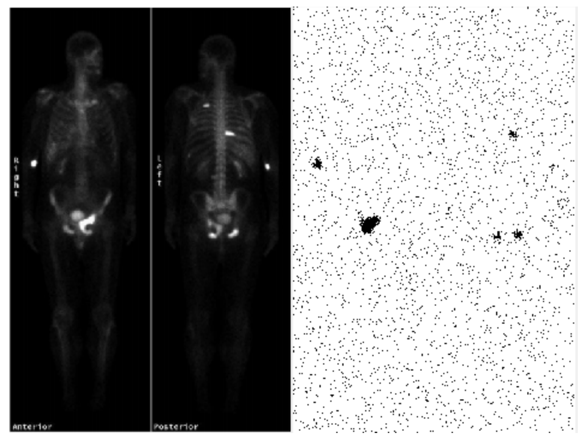

Fig. 9. Mildly Affected Scan

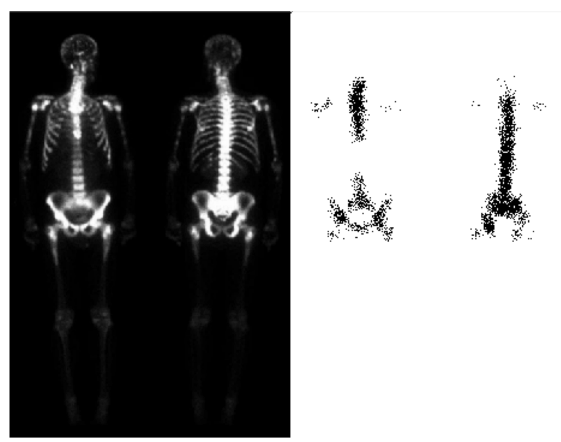

Fig. 10. Bone Metastasis Scan

C. *Global Numerical Optimization*

The table III shows us the time taken to converge to the area possibly where the global optimum is present using SDS, and from here on any local optimization techniques could be used to find the exact global optima. We could conclude from the results that using chaos in SDS has some positive effect in finding the global optima (max or min). The figure 11, 12 represents the position of the agents initially allocated chaotically in the search space and clustering of agents around potential areas at the time of convergence respectively.

TABLE III
RESULTS OF GLOBAL OPTIMIZATION USING SDS

Method	Time Taken (in secs)
Random	1.43
Chaotic	0.89

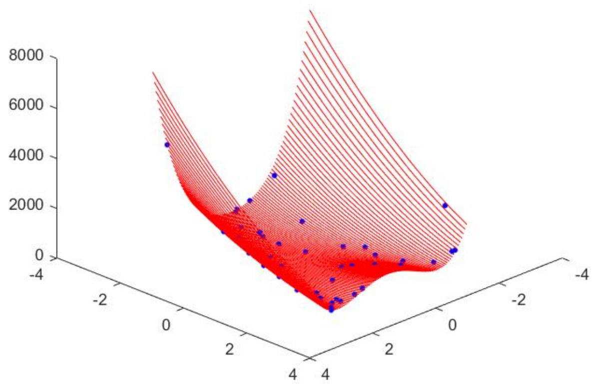

Fig. 11. Initialization Phase

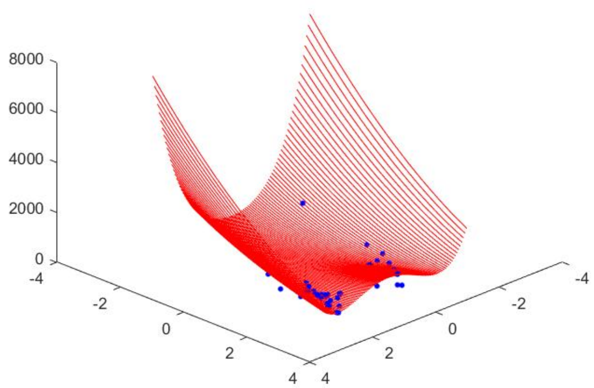

Fig. 12. Convergence Phase

VI. CONCLUSIONS AND FUTURE WORK

The results of applying chaos to SDS has proven to show improvement over the original algorithm in all the three applications used for the experiment, which instigates us to test its effect on many other applications using SDS. Since SDS can be run in a distributed environment and can be processed in parallel, as future work it could be applied in areas of Bioinformatics where there is a huge need to find patterns from a huge volume of data as quickly as possible, similarly the detection of bone metastasis application can be used as adjunct to the doctors expert view, and it would be an efficient solution to prevent patients from doctors doing malpractices, and recently SDS has been used in conjunction with many other swarm intelligence heuristics, replacing with this novel algorithm could produce better results and would be an interesting area to look into.

REFERENCES

[1] Bishop, J. M. "Stochastic searching networks." Proc. 1st IEE Conf. on Artifical neural networks. 1989.

[2] Nasuto, Slawomir J., J. Mark Bishop, and Stanislao Lauria. "Time Complexity Analysis of the Stochastic Diffusion Search." NC. 1998.

[3] Nasuto, Slawomir, and M. Bishop. "Convergence analysis of stochastic diffusion search." PARALLEL ALGORITHMS AND APPLICATION 14.2 (1999): 89-107.

[4] Hurley, Steve, and Roger M. Whitaker. "An agent based approach to site selection for wireless networks." Proceedings of the 2002 ACM symposium on Applied computing. ACM, 2002.

[5] Alatas, Bilal. "Chaotic bee colony algorithms for global numerical optimization." Expert Systems with Applications 37.8 (2010): 5682-5687.

[6] dos Santos Coelho, Leandro, and Viviana Cocco Mariani. "Use of chaotic sequences in a biologically inspired algorithm for engineering design optimization." Expert Systems with Applications 34.3 (2008): 1905-1913.

[7] Moglich, M., U. Maschwitz, and B. Holldobler. "Tandem calling: a new kind of signal in ant communication." Science 186.4168 (1974): 1046-1047.

[8] Al-Rifaie, Mohammad Majid, and Ahmed Aber. "Identifying metastasis in bone scans with stochastic diffusion search." Information Technology in Medicine and Education (ITME), 2012 International Symposium on. Vol. 1. IEEE, 2012.

[9] Caponetto, Riccardo, et al. "Chaotic sequences to improve the performance of evolutionary algorithms." Evolutionary Computation, IEEE Transactions on 7.3 (2003): 289-304.

[10] Omran, Mahamed GH, and Ayed Salman. "Probabilistic stochastic diffusion search." Swarm Intelligence. Springer Berlin Heidelberg, 2012. 300-307.

[11] al-Rifaie, Mohammad Majid, and John Mark Bishop. "Stochastic diffusion search review." Paladyn, Journal of Behavioral Robotics 4.3 (2013): 155-173.