Chaotic sequences to improve genetic algorithms performances

Riccardo Caponetto*, Paolo Arena, Stefano Fazzino and Luigi Fortuna †

Abstract — **This paper proposes a numerical analysis of Genetic Algorithms (GAs) convergence. Based on experimental tests it is investigated the effect of introducing chaotic dynamics during evolution process instead of random ones. The approach is based on the substitution of the random numbers generator with chaotic sequences. The obtained results show that genetic algorithm are extremely sensitive to different random number generators whereas some particular chaotic sequences are always able to increase algorithm genetic features.**

1 Introduction

It is known that genetic algorithms convergence is strictly connected to the random sequences applied on operators during algorithm running. The experience shows that when two genetic optimizations start using different random sequences the final results could be very close but not equal, and the related optimization procedure is strongly time dependent. The appellation *random guided research* well describe the smartness of a method that is essentially random based. Even if random procedures - usually adopted both in commercial and home-made GAs - have passed statistical tests, they can reach neither the global minimum of the considered function nor a short or fixed time convergence of the algorithm. Recently, in many applications like secure transmission [1], natural phenomena modelling [2] and non linear circuit [3], chaotic sequences have been adopted instead of random, showing interesting results. The choice of chaotic sequences is theoretically justified by their unpredictability or, using engineering language, by their spread spectrum characteristic. As a consequence the interesting issue consists in the investigation of the use of chaotic sequences instead of random during GAs evolution. In this paper, by using experimental tests based on De Jong functions [4], it is shown that GAs convergence could be enhanced using particular chaotic series. A comparison between standard Random Number Generators (RNG) and chaotic number generators is made starting from the same starting conditions, showing the existence of particular chaotic sequences always able to increase the algorithm exploitation capability. It means that this approach allows to perform a deeper search of solutions in particular more promising subregions of the problem domain. This communication is structured as follows: in section 2 an introduction on random number generators is reported and the three procedures used during the tests are given; in section 3, after a short introduction on chaotic dynamics, the systems generating the chaotic time series used for the tests are described; in section 4 the results of the test, carried out using the six De Jong functions are reported and some conclusion are given.

*STMicroelectronics, Soft Computing group, Stradale Primosole 50, 95121 Catania Italy, E-mail:riccardo.caponetto@st.com

†Universita' di Catania, Facolta' di Ingegneria, D.E.E.S., Viale A. Doria 6, 95125 Catania Italy, E-mail:lfortuna@dees.unict.it

2 Random number generator and chaotic systems

The minimal standard RNG proposed in [5] represents the core of two of the three RNG adopted in our study. The first RNG that we will use is based on the minimal standard algorithm for the random values, but it shuffles the output to remove low-order serial correlations. The shuffling algorithm is due to Nays and Durhan as described in [7]. In the following this algorithm will be used and labeled as *rand1*. In order to increase the period of the previous generator, L'Ecuyer in [8] proposed a new algorithm with a period of 10^{18}. This commonly used procedure is very powerful and will be applied in the following with the label *rand2*. The last RNG used in our work, and labelled with *rand3*, is based on a *subtractive method* as proposed in [7] by Knuth. The chaotic generators adopted during the tests are the logistic map with parameters $x_0 = 0.2027$ and $a = 4$, the sinusoidal iterator with $a = 2.3$ and $x_0 = 0.7$, the Gauss map, the Lozi map with $a = 1.7$ and $b = 0.5$ and finally the Chua's oscillator with $\alpha = 9$, $\beta = 14.286$, $\gamma = 0$, $m_0 = -1/7$ and $m_1 = 2/7$.

3 Results

In order to compare random and chaotic sequences a software tool has been developed.

It is composed as follow: a *Matlab-Matcom-Libs* that allows to call from the main routine all the functions and toolboxes available with Matlab; a group of *Shared Libraries*, implemented using DLL

(Dynamic Loadable Libraries), that allows to define and to link dynamically the test functions to the main program; a user-friendly *Java Interface* to set the parameters of the GAs; a *Genetic Algorithm*, based on the shareware software Galib 2.4.2 and a *Chaotic Engine*, written in C language that implements all the chaotic dynamic generators. All the *test runs* have the following common parameters, generation number=400, Population size=30, Number of subpopulation=10, Mutation probability=0.001, Crossover probability=0.9, Convergence percentage=0.99, Replacement percentage=0.25, Replacement number=5 and the test functions taken into account are always maximized. Furthermore, SteadyStateGas, binary strings, elitism and single crossover have been adopted.

Standard number generators have been used starting from fixed seeds having the following values: 1, 2, 100, 200, 1000, 2000, 100000, 200000, 1000000, 2000000 (in the following tables from Seed1 to Seed12), while for the chaotic systems the parameters are the same introduced in the previous section. The number of cross-over, the number of mutations, the number of genoma evaluation, the maximumscore and minimumscore, on-line performance, off-line min and max performance and finally the best solution of the maximization have been monitored.

%beginequation

In particular the on-line and on-line performance indexes represent respectively the mean value of the fitness among the population elements at a given generation T, whilst the off-line index represents the mean of the fitness of the best members calculated among all generations. We define *best solution* the gene vector of best fitted element of the population at the latest step of the evolution process. In the following the results concerning the considered six De Joung functions, *f1-f6* are reported. For each function it is given a table showing the results obtained with the five chaotic systems and with the *RNG* (*rand1*, *rand2* and *rand3*).

4 Remarks and conclusion

Taking into account the reported tables the following conclusion can be done. For function from *f1* to *f4* it is possible to note that the number of crossovers, mutations and genome evaluations are greater if chaotic dynamics, instead of RNG, are used. In particular, the number of mutations increase by using sinusoidal and logistic maps. This clearly enhances the exploitation capability of the genetic search. The off-line performance index gets better using both sinusoidal and logistic maps. As a consequence this allows to find more accurate solutions in less generations. Furthermore the performance of the GAs is strongly effected by changes on the initial seed, although using the same RNG. Regarding functions *f5* and *f6*, even if the exploitation capability is maintained, the offline index is slightly better if RNG are used. In particular, for function *f5* and *f6*, *rand1* and *rand2* respectively give better results.

References

[1] R. Caponetto, M. Criscione, L. Fortuna, D. Occhipinti and L.Occhipinti *Synthesis of a Programmable Chaos Generator, based on CNN Architectures, with Applications in Chaotic Communication"*, CNNA '98, London UK, 14-17 April 1998.

[2] M. Bucolo, R. Caponetto, L. Fortuna and M.G. Xibilia *How the Chua Circuit Allows to Modeling Population Dynamics*, NOLTA '98, La Regent, Crans-Montana, Switzerland, September 14-17 1998.

[3] P.Arena, R. Caponetto, L. Fortuna, A. Rizzo and M. La Rosa *Self Organization in non recurent complex system*, To appear on International Journal on Bifurcation and Chaos.

[4] D.E. Goldberg *Genetic Algorithm in Search Optimization and Machine Learning*, Addison Wesley, 1989.

[5] W. Press, S. Teukolsky, W. Vetterling and B. Flannery, *Numerical recipes in C*, in cambridge University Press, 1992.

[6] S. Park and K. Miller *Communication of ACM*, vol. 31 pp 1192-1201, 1988.

[7] D. Knuth *Seminumerical algortihms 2nd ed.*, in The art of computer programming vol.2, Addison Wesley 1981.

[8] P. L'Ecuyer *Communication of ACM*, vol. 31 pp 742-774, 1988.

[9] T.S. Parker and L.O. Chua *Practical Numerical Algorithms for Chaotic System*, Springer Verlag, 1989.

[10] H. Peitgen, H. Jurgens and D. Saupe *Chaos and fractals*, Springer-Verlag, 1992.

f1	*Rand1*	*Rand2*	*Rand3*	*Logistic*	*Chua*	*Sinusoid.*	*Misi.-Lozi*	*Gauss*
Crossover	5365	5440	5419	4691	5341	5170	5570	5578
Mutation	280	279	339	6692	460	22851	0.518	419
Genome eval.	5428	5490	5482	5501	5411	5739	5644	5635
Max Score	78.6432	78.6432	78.6416	78.6432	78.6432	78.6432	78.6432	78.6432
Min Score	2.68719	2.68719	2.68719	2.0465	2.0465	2.0465	2.0465	2.0465
On-line	76.6385	76.8958	76.581	77.9697	78.1127	78.1837	77.5732	77.8274
Off-line max	76.9306	77.233	76.581	78.3193	78.3758	**78.3864**	78.0442	78.0379
Off-line min	76.4572	76.6984	76.3878	77.7008	77.9284	77.9971	77.3344	77.6571
Best solution	-5.12 -5.12 5.12	-5.12 5.12 5.12	5.11 5.12 -5.12	-5.12 5.12 5.12	5.12 -5.12 5.12	-5.12 -5.12 -5.12	-5.12 5.12 5.12	5.12 5.12 5.12

Table 1: Performance with random and chaotic sequences for function *f1*.

f2	*Rand1*	*Rand2*	*Rand3*	*Logistic*	*Chua*	*Sinusoid*	*Misi.-Lozi*	*Gauss*
Crossover	5461	5348	5460	5107	5271	5245	5586	5575
Mutation	193	168	228	4519	292	15356	354	285
Genome eval.	5503	5398	5507	5437	5338	5652	5652	5630
Max Score	3905.93	3905.93	3897.74	3897.74	3905.93	3905.93	3897.74	3897.74
Min Score	0.159313	0.159313	0.159313	0.595894	0.595894	0.595894	0.595894	0.595894
on-line	3855.22	3720.31	3855.83	3862.15	3871.3	3878.97	3852.03	3876.72
off-line max	3876.06	3743.45	3876.87	3881.24	3892 18	**3895.49**	3884.59	3892.33
off-line min	3841.84	3706.42	3843.52	3848.12	3856.92	3863.37	3836.24	3864.63
Best solution	-2.048 -2.048	-2.048 -2.048	2.048 -2.048	2.048 -2.048	-2.048 -2.048	-2.048 -2.048	2.048 -2.048	2.048 -2.048

Table 2: Performance with random and chaotic sequences for function *f2*.

f3	*Rand1*	*Rand2*	*Rand3*	*Logistic*	*Chua*	*Sinusoid.*	*Misi.-Lozi*	*Gauss*
Crossover	5423	5441	5382	4628	5293	4955	5597	5449
Mutation	491	476	562	11149	765	38165	868	937
Genome eval.	5505	5521	5469	5699	5424	5860	5687	5549
Max Score	54	53	55	55	53	55	54	54
Min Score	14	14	14	11	11	11	11	11
on line	51.5298	52.081	53.825	54.0854	51.5502	54.3484	53.3169	53.6489
off-line max	5.1715	52.265	53.995	54.2525	51.715	**54.5325**	53.565	53.7975
off-line min	51.4225	51.9825	53.7225	53.9625	51.4475	54.22	53.2075	53.555
Best solution	-5.00156 4.58187 -5.01765 -5.07937 -5.02437	-5.10062 -4.04951 -4.11639 -5.01562 -5.11516	-5.02594 -5.08062 -5.11766 -5.07937 -5.08203	-5.07453 -502.578 -5.11578 -5.04359 -5.11578	-4.30811 -5.09453 -4.20483 4.81234 -4.60671	-5.05234 -5.07391 -5.04484 -5.08234 -5.08672	-4.30764 -5.1025 -5.09984 -5.05234 -5.086	-5.001 -5.105 -5.117 -4.812 -5.086

Table 3: Performance with random and chaotic sequences for function *f3*.

f4	*Rand1*	*Rand2*	*Rand3*	*Logistic*	*Chua*	*Sinusoid.*	*Misi-Lozi*	*Gauss*
Crossover	5410	5393	5421	4655	5321	5309	5566	5484
Mutation	2942	2849	2902	66879	4637	229180	5310	4210
Genome eval.	5671	5635	5668	6030	5749	6030	5864	5765
Max Score	1064.35	1031.98	1020.14	1038.46	1058.65	1001.74	1120.36	1024.99
Min Score	130.885	130.885	130.885	718.679	718.679	718.679	718.679	718.679
on-line	896.535	873.426	880.359	939.404	933.956	885.926	944.637	920.849
off-line max	907.566	884.39	890.892	**963.642**	944.915	909.212	959.979	929.868
off-line min	891.251	868.296	875.24	928.384	928.585	874.509	938.448	916.311

Table 4: Performance with random and chaotic sequences for function *f4*.

f5	*Rand1*	*Rand2*	*Rand3*	*Logistic*	*Chua*	*Sinusoid.*	*Misi-Lozi*	*Gauss*
Crossover	5461	5348	5460	5107	5271	5245	5596	5575
Mutataion	193	168	228	4519	292	15356	369	285
Genome eval	5503	5398	5507	5437	5338	5652	5656	5630
Max Score	489.233	494.068	489.237	499.002	499.002	499.002	497.018	498.005
Min Score	0.0004894	0.0004894	0.0004894	0.0002565	0.0002565	0.0002565	0.0002565	0.0002565
on-line	486.099	489.463	486.478	494.826	495.015	496.315	489.228	495.519
off-line max	489.208	494.02	489.18	498.292	498.496	**498.877**	495.896	497.977
off-line min	484.362	487.622	484.353	492.469	493.311	494.194	486.849	493.428
Best Solution	-317.435	-317.435	-321.195	-318.595	-318.695	-318.935	0.00100	-163.893
	163.853	-163.853	159.392	-320.755	-317.435	-320.695	-319.375	-320.035

Table 5: Performance with random and chaotic sequences for function *f5*.

f6	*Rand1*	*Rand2*	*Rand3*	*Logistic*	*Chua*	*Sinusoid.*	*Misi-Lozi*	*Gauss*
Crossover	5378	5366	5402	3936	5338	4724	5607	5373
Mutation	990	929	1030	22319	1540	76514	1779	1419
Genome eval.	5491	5479	5525	6030	5524	6030	5733	5448
Max Score	3415.46	3587.34	3527.25	4155.22	3459.45	3718.57	3711.95	3164.39
Min Score	-1104.02	-1104.02	-1104.02	-1095.07	-1095.07	-1095.07	-1095.07	-1095.07
on-line	3054.7	3402.22	3440.98	3890.43	3291.26	3512.21	3519.93	3105.6
off-line max	3076.84	3424.34	3366.18	**3941.63**	3315.97	3551.74	3561.37	3127.59
off-line' min	3042.2	3388.79	3327.63	3862.89	3278.36	3486.56	3503.02	3094.54

Table 6: Performance with random and chaotic sequences for function *f6*.