Turkish Journal of Science & Technology
16(1), 145-153, 2021

Performance of Chaotic Mapping Multi-Objective Optimization Algorithms

Eyup EROZ[1*], Erkan TANYILDIZI[2]
[1] Software Engineering, Technology Faculty, Firat University, Elazig, Turkey
[2] Software Engineering, Technology Faculty, Firat University, Elazig, Turkey
[*1] eeroz@firat.edu.tr, [2] etanyildizi@firat.edu.tr

(Geliş/Received: 07/02/2021; Kabul/Accepted: 19/02/2021)

Abstract: Multi-objective optimization is defined as the process of producing suitable solutions to problems with multiple objectives. The randomly generated string of numbers is of great importance in achieving solutions close to the global optimum in intuitive multi- objective optimization. Collecting the randomly generated string of numbers in a certain area increases the risk of moving away from the global optimum. Chaotic maps are used to reduce this risk it is not periodic as the variety of numbers produced in chaotic maps is high. For this reason, chaotic maps are used in the random number generation part of optimization algorithms. Chaos-based algorithms have become an important field of study because they are flexible and can escape from local minimums. In this study, the effects of chaotic maps on the new and successful Multi-objective Gold Sine Algorithm (MOGoldSA) were compared with the Multi-Objectıve Ant Lion Optimization (MOALO) algorithm.

Key words: Chaotic Map, Multi-Objective Optimization, Chaos-Based Algorithms, Metaheuristic Methods

Kaotik Haritalı Çok Amaçlı Optimizasyon Algoritmalarının Performansı

Öz: Çok amaçlı optimizasyon birden fazla amacı olan problemlere uygun çözümler üretme işlemidir. Sezgisel çok amaçlı optimizasyonda global optimuma yakın çözümler elde etmede rastgele üretilen sayı dizesi büyük öneme sahiptir. Rasgele üretilen sayı dizesinin belli bir alanda toplanması, global optimumdan uzaklaşma riskini arttırmaktadır. Bu riski azaltmak için kaotik haritalar kullanılmıştır. Kaotik haritalarda üretilen sayıların çeşitliliği yüksek olduğu için periyodik değildir. Kaotik haritalarda üretilen sayıların çeşitliliği yüksek olduğu için periyodik değildir. Bu nedenle optimizasyon algoritmalarının rasgele sayı üretim kısmında kaotik haritalar kullanılmıştır. Kaos temelli algoritmalar esnek ve lokal minimumdan kaçabilme özelliklerine sahip oldukları için önemli bir çalışma alanı haline gelmiştir. Bu çalışmada kaotik haritaların yeni ve başarılı Çok Amaçlı Altın Sinüs Algoritması (MOGoldSA) üzerinde etkileri, literatürde yer edinmiş Çok Amaçlı Karınca Aslanı Optimizasyonu (MOALO) algoritması ile karşılaştırılarak incelenmiştir.

Anahtar kelimeler: Kaotik Harita, Çok Amaçlı Optimizasyon, Kaos Temelli Algoritmalar, Metasezgisel Yöntemler

1. Introduction

Metaheuristic optimization algorithms are a very popular field that is easy to transform, has high solution capacity, and has wide usage. Metaheuristic optimization algorithms have been developed inspired by nature. They solve problems based on all physical or biological events that exist in nature [1]. Metaheuristic optimization is the process of generating suitable solutions for optimization problems by starting from a randomly generated number string. Heuristic optimization, starting from a random string of numbers during the optimal solution to the problem or achieve convergence iterations are targeted. Therefore, the most important criteria is to determine the random string of numbers to reach the optimum solution. Generating the same values in random number strings affects the performance of the algorithm. The main goal of heuristic optimization is to reach a global optimum. Accordingly, chaotic maps are used in random number generation to increase the performance of heuristic optimization algorithms [2].

Chaos theory is the study of chaotic dynamic systems sensitive to initial conditions. In the chaos theory, the theorem that the smallest changes that may occur in the initial conditions will have a large effect on the results. The "butterfly effect" is the most popular example of chaos theory. In this direction, it has been predicted that heuristic optimization algorithms can be used to generate random number generation with chaotic maps and to produce appropriate solutions to problems [3].

* Corresponding author: eeroz@firat.edu.tr. ORCID Number of authors: [1] 0000-0003-2670-0606, [2] 0000-0003-2973-9389

2. Chaotic Multi-Objective Gold Sine Algorithm

MOGoldSA is the applicable version of Gold-SA, a single-objective optimization algorithm, for multi-objective problems. Gold-SA is a math-based metaheuristic single-objective optimization algorithm developed from the sine function. Sinus function is a periodic function that repeats values at regular intervals. The duration of the sine function is 2π and all unit circles with sine values can be scanned as shown in Figure 1 [12].

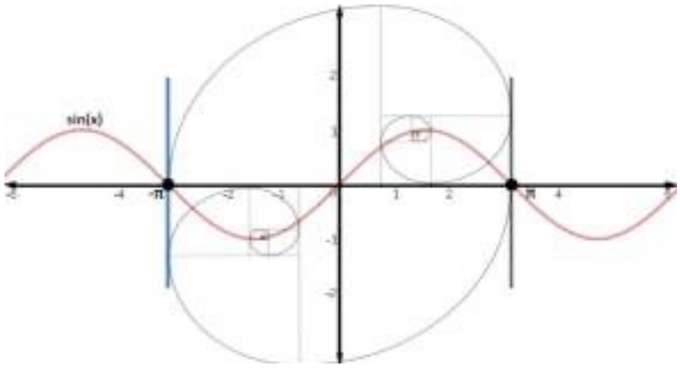

Figure 1. GoldSA scanning process [12]

Equation 1 is used for GoldSA to give the best result:

$$V(i,j) = V(i,j) * |\sin r_1| - p * \sin r_1 * |x_1 * D(j) - x_2 * V(i,j)| \tag{1}$$

Where, $V\ (i,j)\ i$ is the value of the current solution in size. $D(j)$ determined target value is a random number in the range r_1, $[0,2\pi]$ and the interval p $[\,0,\pi]$. x_1 ve $x_2$are coefficients obtained by the golden division method. These coefficients narrow the search area and allow the current value to approach the target value.

MOGoldSA was developed based on the hypothesis that Gold-SA can show the same success on multi-objective optimization problems, based on its features such as wide scanning of the search space, producing near-optimum results and fast operation while producing solutions to problems. All parameters of MOGoldSA except archive size are the same as Gold-SA [13].

In the MOGoldSA, p parameter is randomly determined in the range of $[0,\pi]$ for each search agent. These randomly generated values prevent a large search of the search space. To eliminate this problem, Mirjalili and Amir H. Gandomi [14] achieved successful results by normalizing the chaotic maps with Equation 2 and Equation 3. In this study, it is aimed to scan as much of the search space as possible in order to find better solutions to multi-objective optimization problems by using the r_1, pattern shown in Equation 1, which is adaptively decreasing by integrating with normalized chaotic maps.

$$V(t) = Max - \frac{t}{T}(Max - Min) \tag{2}$$

$$Normalize\ (K_i(t);\ [a,b]'dan\ [0,V(t)]'ye):\ K_i^{norm}(t) = \frac{(K_i(t)-a)\times(V(t)-0)}{(b-a)} + 0 = \frac{(K_i(t)-a)\times(V(t))}{(b-a)} \tag{3}$$

Where, K: Chaotic map, i: Index of the chaotic map, t: Current iteration, T: Maximum number of iterations, $[Max - Min]$: Represents the adaptive range. $[a,b]$: Shows the range of chaotic maps. $V\ (t)$ is reduced by each iteration, while at each iteration $[a,b]$ matches $[0,V\ (t)]$. This situation refers to the adaptive normalization process.

As a result, the value of the coefficient p is updated using Equation 4.

$$p(t) = K_i^{norm}(t) + c - t \times ((c)/T) \tag{4}$$

Adding chaotic maps to the randomly generated sequence contributes to the random behavior of both adaptive p and chaotic maps at the same time. Chaotic maps provide to have a wide spectrum by generating different random sequences each time. By changing the value of p abruptly and helping to get rid of the local minimum, it provides better convergence speed. Adaptive normalization, on the other hand, slows down the transition of the chaos-based MOGoldSA from the exploration phase to the exploitation phase.

3. Chaotic Multi-objective Ant Lion Optimizer

ALO algorithm was developed to imitate the hunting mechanisms of ant lions. The general steps of the algorithm are as follows [15]:

a. Ants cluster is started with random values and are the main search agents in ALO.
b. The fitness value of each ant is calculated.
c. The ants make random walks around the ant lions and move on the search area.
d. Antlion population is never evaluated.
e. There is an antlion assigned to each ant.
f. There is an elite antlion that affects the movement of ants.
g. If any antlion gets better than elite, it will be replaced by elite.
h. Steps b to g are repeatedly executed until an end criterion is met.
i. The position and fitness value of the elite antlion is returned as the best estimate for the global optimum.

The main task of the ants is to explore the search area. They move in random walks in the search area. Ant lions maintain the best position achieved by the ants and are directed towards promising locations for ant search. To solve optimization problems, ALO simulates the random roaming of ants, clinging to an ant lion pit, building a pit, sliding towards ant lions, hunting and rebuilding the pit, and elite ant lion selection. The mathematical model established for each of the steps specified is presented in the next paragraphs.

The formula for the random walk of ants in the ALO algorithm is given in Equation 5:

$$X(t) = [0, cumsum(2r(t_1) - 1),\ cumsum(2r(t_2) - 1), \dots, cumsum(2r(t_n) -)] \tag{5}$$

Where, $cumsum$ calculates the cumulative sum, n is the maximum number of iterations, t represents the random walk step

To keep the random walk within the search area and prevent the ants from overflowing, random walks are normalized using Equation 6 below:

$$X_i^t = \frac{(X_i^t - a_i) \times (d_i^t - c_i^t)}{(b_i - a_i)} + c_i^t \tag{6}$$

Where, $c_i^t\, t.$ minimum variable i in iteration, $d_i^t\, t.$ maximum variable i in iteration, $a_i\ i.$ is the random walking minimum of the variable and $b_i\ i.$ is the random walking maximum of the variable

ALO simulates the ants squeezing into the ant lion's pit by random walks and this process is given in Equation 7 and Equation 8.

$$c_i^t = Antlion_j^t + c^t \tag{7}$$
$$d_i^t = Antlion_j^t + d^t \tag{8}$$

Where,c^t the minimum of all variables in $t.$ and d^t. Shows the vector containing the maximum of all variables in $t.$ iteration.c_i^t, $i.$ the minimum of all variables for ant, d_i^t, shows the maximum of all variables for the ant. $Antlion_j^t$, shows the position of selected $j.$ ant lions in $t.$ iteration.

Large ant lions create larger pits to increase their survival time. To simulate this process, ALO uses the roulette wheel operator who selects ant lions based on their suitability values. The roulette wheel helps ant lions striving to attract more ants. Using the formulas in Equation 9 and Equation 10, the limits of random walks should be reduced adaptively to simulate sliding ants from ant lions:

$$c^t = \frac{c^t}{I} \tag{9}$$

$$d^t = \frac{d^t}{I} \tag{10}$$

Where, I is ratio, c^t is the minimum of all variables in $t.$ iteration, d^t, Shows the vector containing the maximum of all variables in $t.$ iteration

The final step in ALO is to catch the ants and rebuild the pit. It is simulated as in Equation 11:

$$Antlion_j^t = Ant_i^t \quad if\ f(Ant_i^t) < \left(Antlion_j^t\right) \tag{11}$$

Where, t current number of iterations, $Antlion_j^t$, shows the position of selected $j.$ ant lions in $t.$ iteration, $Ant_i^t\ i.$ indicates the ant's position in t.iteration.

The last operator in ALO is the elite, where the most suitable ant lion is stored during optimization. Elite is the only ant lion that can infect all ants. This means that random walks to ant lions are drawn towards a chosen ant lion or elite ant lion. The formula is given in Equation 12 to consider both cases.

$$Ant_i^t = \frac{R_A^t + R_E^t}{2} \tag{12}$$

Where, R_A^t is the random walk around the antlion selected using a roulette wheel and R_E^t is the random walk around the elite antlion.

The ALO algorithm created with the specified mathematical methods has been developed for multi-objective problems and the MOALO algorithm has been brought to the literature [16]. The optimum solutions found by using archive method were stored in MOALO, which tries to produce optimal solutions by using the Pareto method. The convergence of the MOALO algorithm is inherited from the ALO algorithm. If a solution is chosen from the archive, the ALO algorithm can improve its quality. However, a limit must be found for the archive, and solutions must be selected from the archive to improve distribution. In this approach, the perimeter of each solution is analyzed by considering a predefined radius. The number of solutions in the environment counts and is considered the measure of dispersion. Improving the distribution of solutions in the archive is carried out in two stages.

Equation 11 and Equation 13 must be substituted for the nature of multi-objective objects to require ALO to solve multi-objective problems.

$$Antlion_j^t = Ant_i^t \quad if\ f(Ant_i^t) \prec f\left(Antlion_j^t\right) \tag{13}$$

Where, t current number of iterations, $Antlion_j^t$, shows the position of selected $j.$ ant lions in $t.$ iteration, $Ant_i^t\ i.$ indicates the ant's position in t.iteration.

The rest of the operators in MOALO are the same as in ALO. MOALO produces effective solutions to many constrained and unconstrained optimization problems in the literature. In order to increase this performance, it is aimed to produce more effective solutions by using chaotic maps during the generation of random number strings. Thus, Chaotic Ant Lion Optimizer (CMOALO) was developed.

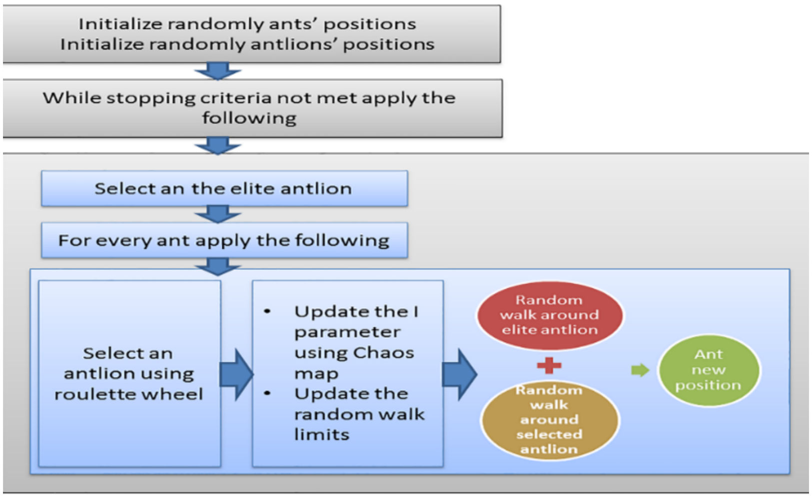

Figure 2. CMOALO [15]

4. Chaos Maps

Chaos literally means great disorder or confusion. Chaotic systems are dynamic systems that depend on initial conditions. Chaos theory studies the behavior of unpredictable systems. In order for a system to be chaotic, it must not be sensitive to initial conditions, topologically complex, periodic, ergodic and stochastic. Due to the dynamic properties of chaos variables, chaos search has the ability to escape from local optima according to random search. It has therefore been applied for optimization. Chaotic map is a map that exhibits a kind of chaotic behavior [17, 18]. In this study, 5 chaotic maps (Logical, Singer, Sinusoidal, Piecewise and Tent) were used for the comparison of multi-objective optimization algorithms. The mathematical expressions of the maps are as follows:

Table 1. Chaotic Maps

	Chaotic Maps	Function	
C1	Piecewise	$x_{i+1} = \begin{cases} \frac{x_i}{P} & 0 \le x_i < P \\ \frac{x_i - P}{0.5 - P} & P \le x_i < 0.5, \\ \frac{1 - P - x_i}{0.5 - P} & 0.5 \le x_i < 1 - P, \\ \frac{1 - x_i}{P} & 1 - P \le x_i < 1 \end{cases}$	(14)
C2	Singer	$x_{i+1} = \mu(7.86x_i - 23.31x_i^2 + 28.75x_i^3 - 13.302875x_i^4), \mu = 2.3$	(15)
C3	Sinusoidal	$x_{i+1} = ax_i^2 \sin(\pi x_i), a = 2.3$	(16)
C4	Tent	$x_{i+1} = \begin{cases} \frac{x_i}{0.7} & x_i < 0.7 \\ \frac{10}{3}(1 - x_i) & x_i \ge 0.7 \end{cases}$	(17)
C5	Logical	$x_{i+1} = ax_i(1 - x_i), a = 4$	(18)

5. Experimental Results

In this study, comparisons were made over 13 different comparison functions (constrained and unconstrained). In our studies using 11 different chaotic maps on MOGoldSA, 5 most effective chaotic maps were used [19]. Using these chaotic maps on the MOALO, two multi-objective optimization algorithms that produce successful results are compared. In the study, each algorithm was run 20,000 iterations and 100 optimum solutions were obtained. The mathematical expressions of some of the functions used are shown in Table 2.

Table 2. Mathematical model of some comparison functions used

F	Variable Limit	Objective Functions
FON	$x_i \in [-4,4]$ $i = 1, \ldots, n$ $n = 10$	$f_1(x) = 1 - exp\left(-\sum_1^3 \left(x_i - \frac{1}{\sqrt{3}}\right)^2\right)$
KUR	$x_i \in [-5,5]$ $i = 1, \ldots, n$ $n = 3$	$f_1(x) = \sum_{i=1}^{n-1}\left(-10exp\left(-0.2\sqrt{x_i^2} + \sqrt{x_{i+1}^2}\right)\right)$ $f_2(x) = \sum_{i=1}^{n}(\|x_i\|^{0.8} + 5\sin x_i^3)$
CNEX	$x_1 \in [0,11]$ $x_2 \in [0,5]$	$f_1(x) = x_1$ $f_2(x) = (1 + x_2)/x_1$ $g_1(x) = 9x_1 + x_2 \geq 6$ $g_2(x) = -x_2 + 9x_1 \geq 1$
SRN	$x_1 \in [-20,20]$ $x_2 \in [-20,20]$	$f_1(x) = 2 + (x_1 - 2)^2 + (x_2 - 2)^2$ $f_2(x) = 9x_1 - (x_2 - 1)$ $g_1(x) = {x_1}^2 + {x_2}^2 \leq 225$ $g_2(x) = x_1 - 3x_2 + 10 \leq 0$

The criteria used to evaluate the success of the developed algorithms are called performance criteria. In our study, the performance criteria that have been included in the literature and frequently used in the comparison of multi-objective optimization algorithms were used. These; General Distance (GD) and Reversal General Distance (RGD)

General Distance (GD):

GD was proposed by Veldhuizen in 1998 and the solutions obtained are calculated by measuring the distances between the Pareto-optimal fronts. Whether a solution belongs to the optimal solution set is calculated by the distance of the obtained solution to the pareto optimal set. This criterion called General Distance is calculated with Equation 19. [20,21]:

$$GD = \frac{\left(\sum_{i=1}^{|Q|} d_i^p\right)^{1/p}}{|Q|} \tag{19}$$

Where, d_i parameters expression in Equation 19 is the Euclidean distance and it is calculated as shown in Equation 20.

$$d_i = \min_{k=1} |P| \sqrt{\sum_{m=1}^{M} (f_m^i - f_m^k)^2} \tag{20}$$

Where, f_m^k, P 'nin k. of the element m. objective function value.

Reverse General Distance (RGD):

It is an improved version of the GD criterion. Unlike the GD criterion, Pareto optimally calculates the distance of the points on the front to the obtained solution set. The mathematical expression of the RGD criterion is included in Equation 21

$$RGD = \frac{\left(\sum_{i=1}^{|P|} d_i^Q\right)^{1/q}}{|P|} \tag{21}$$

The biggest advantage of RGD is that it can measure both convergence and diversity of solutions simultaneously. Similar to GD, RGD is that as the number of targets increases, it becomes exponentially costly. [21].

Table 3 shows the statistical results of MOGoldSA according to GD criteria. According to these results, the Singer map was the most successful chaotic map with a success rate of 6/13, while the Logical map was 4/13, Piecewise map 2/13, Tent and Sinusoidal maps 1/13.

Table 3. CMOGoldSA: Statistical results according to GD criterion

		BEAM	BNH2	CNEX	CTP8	FON1	KUR1	SRN1	VIE1	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
Singer	Best	0,0196	**5,3389**	**0,1011**	**0,1158**	**0,0088**	14,8731	3,6958	**0,0942**	0,0134	0,0161	0,0174	**0,4558**	0,0098
	Mean	0,0249	8,5491	0,2333	0,2248	0,0134	15,0780	8,1060	0,2083	1,7700	3,0605	1,4542	0,8867	0,3403
	Worst	0,0380	8,6710	0,5421	0,8421	0,0253	15,4752	21,9288	0,5057	2,7173	4,1907	2,9696	0,9127	1,0561
	Std	0,0040	0,6067	0,0912	0,1309	0,0033	0,1854	4,9218	0,0990	1,0313	1,5669	1,0643	0,0851	0,3768
Sinusoidal	Best	0,0177	8,6188	0,1379	0,1293	0,0090	14,8748	3,1611	0,0991	0,0127	0,0145	0,0189	0,7854	**0,0081**
	Mean	0,0243	8,6602	0,2782	0,2446	0,0133	15,0907	7,7150	0,1825	1,3320	3,6167	1,3280	0,9084	0,3382
	Worst	0,0331	8,6705	0,8724	0,4753	0,0262	15,5087	14,2698	0,3196	2,8237	4,4080	2,7000	0,9127	1,0071
	Std	0,0039	0,0115	0,1753	0,0927	0,0039	0,2083	3,3208	0,0640	1,2426	0,9187	0,9787	0,0232	0,3959
Tent	Best	0,0164	8,5120	0,1278	0,1283	0,0120	14,8614	3,7117	0,0945	0,0137	0,0124	**0,0140**	0,5226	0,0089
	Mean	0,0242	8,6513	0,2172	0,2302	0,0149	15,1675	7,6690	0,2207	1,5216	3,2706	1,3654	0,8915	0,4319
	Worst	0,0318	8,6695	0,3728	0,5669	0,0235	15,5589	16,7396	0,4826	2,8675	4,2219	2,4975	0,9127	0,9719
	Std	0,0046	0,0324	0,0653	0,0995	0,0030	0,2229	2,9734	0,0877	1,1665	1,4843	0,8870	0,0766	0,3830
Piecewise	Best	0,0169	8,6199	0,1466	0,1159	0,0102	14,8678	3,1961	0,1070	**0,0123**	0,0127	0,0166	0,8137	**0,0081**
	Mean	0,0255	8,6613	0,2410	0,2168	0,0152	15,1163	6,8933	0,2253	1,1708	3,0995	1,2688	0,9065	0,2760
	Worst	0,0396	8,6695	0,5943	0,3378	0,0292	15,5114	13,2787	0,5596	2,7529	4,2769	2,8002	0,9127	0,9722
	Std	0,0049	0,0106	0,1060	0,0614	0,0051	0,2205	2,7146	0,0960	1,2342	1,4680	1,0747	0,0219	0,3754
Logical	Best	**0,0160**	8,5558	0,1332	0,1182	0,0089	**14,8609**	**3,0946**	0,1003	0,0124	**0,0105**	0,0196	0,7978	0,0097
	Mean	0,0241	8,6567	0,2316	0,2185	0,0135	15,0573	6,8826	0,1826	1,7028	3,0435	1,3600	0,9082	0,2024
	Worst	0,0356	8,6698	0,6840	0,4373	0,0251	15,4999	15,1967	0,3269	3,0148	4,2225	2,5894	0,9127	0,9188
	Std	0,0042	0,0223	0,1051	0,0853	0,0033	0,1863	3,3612	0,0595	1,2194	1,5496	0,9605	0,0210	0,3109

The statistical results of the CMOALO algorithm according to the GD criteria are given in Table 4. Considering these results, the Tent map was the most successful chaotic map at the rate of 5/13, while the Logical map was 4/13, Singer and Sinusoidal map 2/13, and Tent map 1/13.

Table 4. CMOALO: Statistical results according to GD criterion

		BEAM	BNH2	CNEX	CTP8	FON1	KUR1	SRN1	VIE1	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
Singer	Best	0,0186	8,6630	0,0418	0,0455	**0,0078**	15,5486	1,7511	0,0666	**0,0786**	0,3093	0,0167	3,7666	0,0082
	Mean	0,0256	8,6664	0,0588	0,0582	0,0100	15,6041	2,1612	0,1000	0,2097	0,6869	0,0408	12,6633	0,0118
	Worst	0,0500	8,6694	0,0863	0,0899	0,0132	15,6291	3,4782	0,1764	0,3758	0,7310	0,0852	23,8325	0,0180
	Std	0,0074	0,0016	0,0160	0,0115	0,0014	0,0176	0,4171	0,0295	0,0804	0,1084	0,0170	5,8379	0,0026
Sinusoidal	Best	0,0162	8,6608	0,0420	0,0441	0,0085	15,5569	**1,6699**	**0,0595**	0,1145	0,5204	0,0241	1,5464	0,0082
	Mean	0,0270	8,6661	0,0597	0,0549	0,0099	15,6068	2,0984	0,0986	0,2270	0,7083	0,0456	12,0797	0,0106
	Worst	0,0548	8,6687	0,1558	0,0898	0,0125	15,6274	2,6700	0,1597	0,4048	0,7309	0,0940	34,3103	0,0162
	Std	0,0092	0,0017	0,0245	0,0112	0,0011	0,0144	0,3281	0,0311	0,0792	0,0578	0,0182	8,4790	0,0020
Tent	Best	**0,0154**	**8,6578**	0,0383	0,0405	**0,0078**	**15,1948**	1,7320	0,0620	0,1498	0,6685	0,0617	14,3162	**0,0081**
	Mean	0,0279	8,6664	0,0709	0,0902	0,0092	15,5432	2,2705	0,1103	0,3525	0,7267	0,1945	28,6343	0,0659
	Worst	0,1306	8,6694	0,2982	0,1859	0,0109	15,6171	3,6464	0,1689	0,4774	0,7337	0,3455	44,9801	0,2662
	Std	0,0211	0,0022	0,0460	0,0418	0,0008	0,0773	0,4291	0,0303	0,0823	0,0156	0,0645	8,3718	0,0748
Piecewise	Best	0,0162	8,6608	0,0444	0,0404	0,0085	15,5569	1,7920	0,0675	0,0976	**0,1794**	0,0206	2,5639	0,0089
	Mean	0,0270	8,6661	0,0672	0,0627	0,0099	15,6068	2,1759	0,0940	0,1957	0,6797	0,0474	13,3858	0,0117
	Worst	0,0548	8,6687	0,1146	0,1159	0,0125	15,6274	3,3305	0,1703	0,3145	0,7310	0,1109	30,4019	0,0168
	Std	0,0092	0,0017	0,0201	0,0185	0,0011	0,0144	0,4162	0,0272	0,0625	0,1333	0,0236	6,7338	0,0019
Logical	Best	0,0154	8,6597	**0,0374**	**0,0394**	0,0088	15,5582	1,6751	0,0713	0,0996	0,4693	**0,0162**	**1,4243**	0,0087
	Mean	0,0217	8,6666	0,0564	0,0604	0,0101	15,6036	2,0199	0,1015	0,2240	0,7033	0,0335	10,8912	0,0114
	Worst	0,0316	8,6691	0,0891	0,1100	0,0127	15,6296	2,6162	0,1530	0,3779	0,7309	0,0517	33,4715	0,0184
	Std	0,0049	0,0020	0,0138	0,0172	0,0010	0,0156	0,2617	0,0254	0,0738	0,0751	0,0097	7,6318	0,0022

The statistical results of the CMOGoldSA algorithm according to the RGD criteria are shown in Table 5. According to the results, the Sinusoidal map produced successful results in 7 of 13 functions and became the most successful chaotic map, while the Logical map succeeded 6/13, Singer, Tent and Piecewise maps achieved a success rate of 3/13. Also, the Sinusoidal map reached optimum results in ZDT2 and ZDT4 functions, Logical ZDT3 and ZDT4 functions, Piecewise ZDT2, Tent and Singer in ZDT4 functions.

Table 5. CMOGoldSA: Statistical results according to RGD criterion

		BEAM	BNH2	CNEX	CTP8	FON1	KUR1	SRN1	VIE1	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
Singer	Best	0,0010	**5,7684**	0,0022	0,0022	**0,0009**	15,3118	0,7678	0,0036	0,0009	0,0003	0,0009	**0,0000**	0,0002
	Mean	0,0012	8,2012	0,0091	0,0078	0,0011	15,7382	1,9500	0,0126	2,1017	3,1650	1,5488	0,0010	0,9466
	Worst	0,0015	8,5275	0,0601	0,0224	0,0014	16,0898	3,6985	0,0458	3,2244	4,3519	3,1861	0,0078	3,9783
	Std	0,0001	0,4677	0,0108	0,0056	0,0001	0,2057	0,8097	0,0090	1,2969	1,6264	1,3829	0,0018	1,1435
Sinusoidal	Best	0,0011	8,0936	0,0021	**0,0017**	**0,0009**	**15,2473**	0,5931	**0,0034**	**0,0007**	**0,0000**	0,0008	**0,0000**	0,0002
	Mean	0,0013	8,3032	0,0061	0,0078	0,0011	15,7588	1,8752	0,0102	1,4344	3,6564	1,4461	0,0001	0,9473
	Worst	0,0015	8,5889	0,0153	0,0253	0,0013	16,0273	5,7892	0,0300	3,3335	4,3234	3,0472	0,0025	3,0880
	Std	0,0001	0,1155	0,0037	0,0057	0,0001	0,2047	1,1901	0,0056	1,4656	1,0193	1,3846	0,0005	1,1456
Tent	Best	**0,0009**	7,9854	0,0023	0,0023	**0,0009**	15,2952	0,6781	0,0038	0,0008	0,0003	0,0006	**0,0000**	0,0002
	Mean	0,0012	8,3069	0,0075	0,0086	0,0011	15,8170	1,4388	0,0133	1,7489	3,2868	1,3595	0,0004	1,1408
	Worst	0,0014	8,5607	0,0554	0,0483	0,0014	16,0341	3,1036	0,0486	3,2696	4,3369	3,0960	0,0045	2,8227
	Std	0,0001	0,1173	0,0094	0,0107	0,0001	0,1761	0,6103	0,0098	1,4429	1,5090	1,3883	0,0011	1,0680
Piecewise	Best	0,0010	8,0970	**0,0019**	0,0027	0,0010	15,2508	**0,4966**	0,0052	0,0009	**0,0000**	0,0006	0,0000	0,0002
	Mean	0,0013	8,2607	0,0090	0,0095	0,0011	15,7468	1,6787	0,0115	1,3031	3,1907	1,4583	0,0003	0,8479
	Worst	0,0015	8,4658	0,0348	0,0538	0,0013	16,0427	5,0288	0,0252	3,3105	4,4828	3,2030	0,0033	3,0393
	Std	0,0001	0,0882	0,0084	0,0099	0,0001	0,2242	0,9679	0,0057	1,5127	1,6375	1,3919	0,0009	1,1469
Logical	Best	0,0010	8,1310	0,0025	**0,0017**	**0,0009**	15,2696	0,5750	0,0048	**0,0007**	0,0003	**0,0000**	**0,0000**	**0,0001**
	Mean	0,0012	8,3086	0,0139	0,0064	0,0011	15,7545	1,6344	0,0172	1,9242	3,0808	1,4622	0,0003	0,6460
	Worst	0,0014	8,5161	0,0904	0,0289	0,0016	16,0259	3,3369	0,0415	3,3962	4,1574	3,3535	0,0040	3,4738
	Std	0,0001	0,0920	0,0202	0,0049	0,0001	0,2072	0,7028	0,0099	1,3986	1,5776	1,4046	0,0009	1,0787

The statistical results of the CMOALO algorithm are given in Table 6. According to the results, the Piecewise map produced very successful results at the rate of 8/13. The Logical map is followed by 4/13 and the Tent map with 2/13. In addition, Piecewise and Logical maps have achieved optimum results in ZDT2 function.

Table 6. CMOALO: Statistical results according to RGD criterion

		BEAM	BNH2	CNEX	CTP8	FON1	KUR1	SRN1	VIE1	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
Singer	Best	0,0186	8,6630	0,0418	0,0455	0,0078	15,5486	1,7511	0,0666	0,0786	0,3093	0,0167	3,7666	0,0082
	Mean	0,0256	8,6664	0,0588	0,0582	0,0100	15,6041	2,1612	0,1000	0,2097	0,6869	0,0408	12,6633	0,0118
	Worst	0,0500	8,6694	0,0863	0,0899	0,0132	15,6291	3,4782	0,1764	0,3758	0,7310	0,0852	23,8325	0,0180
	Std	0,0074	0,0016	0,0160	0,0115	0,0014	0,0176	0,4171	0,0295	0,0804	0,1084	0,0170	5,8379	0,0026
Sinusoidal	Best	0,0162	8,6608	0,0420	0,0441	0,0085	15,5569	1,6699	0,0595	0,1145	0,5204	0,0241	1,5464	0,0082
	Mean	0,0270	8,6661	0,0597	0,0549	0,0099	15,6068	2,0984	0,0986	0,2270	0,7083	0,0456	12,0797	0,0106
	Worst	0,0548	8,6687	0,1558	0,0898	0,0125	15,6274	2,6700	0,1597	0,4048	0,7309	0,0940	34,3103	0,0162
	Std	0,0092	0,0017	0,0245	0,0112	0,0011	0,0144	0,3281	0,0311	0,0792	0,0578	0,0182	8,4790	0,0020
Tent	Best	0,0154	8,6578	0,0383	0,0405	0,0078	**15,1948**	1,7320	0,0620	0,1498	0,6685	0,0617	14,3162	**0,0081**
	Mean	0,0279	8,6664	0,0709	0,0902	0,0092	15,5432	2,2705	0,1103	0,3525	0,7267	0,1945	28,6343	0,0659
	Worst	0,1306	8,6694	0,2982	0,1859	0,0109	15,6171	3,6464	0,1689	0,4774	0,7337	0,3455	44,9801	0,2662
	Std	0,0211	0,0022	0,0460	0,0418	0,0008	0,0773	0,4291	0,0303	0,0823	0,0156	0,0645	8,3718	0,0748
Piecewise	Best	**0,0011**	**8,1834**	0,0025	**0,0020**	**0,0021**	15,8882	**0,2966**	**0,0025**	0,0006	**0,0000**	**0,0007**	3,3302	0,0276
	Mean	0,0013	8,3683	0,0044	0,0048	0,0029	16,0165	0,4103	0,0043	0,0010	0,0001	0,0014	14,6304	0,1274
	Worst	0,0016	8,5468	0,0067	0,0063	0,0037	16,1998	0,6741	0,0101	0,0022	0,0003	0,0024	33,4134	0,3635
	Std	0,0002	0,0943	0,0011	0,0012	0,0004	0,0739	0,0858	0,0018	0,0004	0,0001	0,0004	7,2690	0,0924
Logical	Best	**0,0011**	8,2153	**0,0021**	0,0021	0,0023	15,8881	0,3007	0,0028	**0,0005**	**0,0000**	0,0009	1,3971	0,0169
	Mean	0,0013	8,3646	0,0048	0,0045	0,0029	16,0029	0,4283	0,0043	0,0012	0,0001	0,0014	11,9345	0,0994
	Worst	0,0016	8,5110	0,0063	0,0070	0,0036	16,1689	0,5515	0,0062	0,0033	0,0005	0,0036	35,7860	0,2220
	Std	0,0002	0,0886	0,0010	0,0012	0,0004	0,0704	0,0803	0,0010	0,0009	0,0002	0,0006	8,0832	0,0530

The success of CMOGoldSA and CMOALO algorithms, which are two successful multi-objective optimization algorithms in the literature, with chaotic maps were compared. According to the statistical results in Table 3 and Table 4, CMOALO 7/13 and CMOGoldSA were successful in Singer and Logical maps, while CMOGoldSA was 8/13 in Tent map, 7/13 CMOGoldSA in Sinusoidal map, and 9/13 in Piecewise map. It has been observed that the CMOALO optimization algorithm produces successful results. According to the GD criterion, it was observed that CMOGoldSA produced more successful results in Tent and Sinusoidal maps.

If two multi-objective optimization algorithms are compared according to the statistics given in Table 5 and Table 6; According to Sinusoidal and Piecewise maps, the CMOGoldSA algorithm produced a successful result at the rate of 9/13, 8/13 in the Singer and Logical map, and 7/13 in the Tent map. In general, the CMOGoldSA optimization algorithm produced more successful results in all chaotic maps according to RGD criteria.

6. Conclusion

Chaotic maps are used to improve the performance of the multi-objective gold sine algorithm (MOGoldSA). Chaotic maps help MOGoldSA reach the large search area and gradually move into the exploitation phase. Based on these performance data, it was seen that better results were obtained in generating suitable solutions to the problem as a result of the small interventions made on random number strings used in the beginning in multi-objective optimization. The chaotic maps used in the study have generally produced successful results in producing solutions suitable for the constrained and unconstrained engineering problems and produced more successful results than the CMOALO algorithm, which has an important place in the literature.

References

[1] Mirjalili S, Lewis A. The Whale Optimization Algorithm, ADV ENG SOFTW 2016; 95: 51-67
[2] Tanyıldızı E, Cigal T. Kaotik Haritalı Balina Optimizasyon Algoritmaları, FÜMD 2017; 29(1): 307 – 317.
[3] Tanyıldızı E, Demir G. Nümerik Optimizasyon için Kaotik Altın Sinüs Algoritması, FUMBD 2019; 31(1): 91 – 97.
[4] S. M. P. J. S. S., “Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems,” Appl Intell, DOI 10.1007/s10489-016-0825-8, 2016.
[5] Wei G. Shoubin W. Chaos Ant Colony Optimization and Application. Fourth International Conference on Internet Computing for Science and Engineering;2009; Harbin, China.
[6] Ayan K. and Kilic U. Solution of multi-objective optimal power flow with chaotic artificial bee colony algorithm, INT REV ELECTR ENG-I; 2011; England, 6(3):1365–1371.
[7] Ying S, Yuelin G,Xudong S. Chaotic Multi-Objective Particle Swarm Optimization Algorithm Incorporating Clone Immunity. Mathematics 2019, 7, 146;
[8] Dunia S. Ramzy A. A Chaotic Crow Search Algorithm for High-Dimensional Optimization Problems. Basrah Journal for Engineering Sciences January 2018; 17(1):15-25
[9] Danqing G, Junping W, Jun H. Renmin H and Maoqiang S. Chaotic-NSGA-II: An effective algorithm to solve multi-objective optimization problems. ICISS; 2010; Guilin, China.
[10] Zhang H. Zhou J. Zhang Y. Fang N. and Zhang R. Short term hydrothermal scheduling using multi-objective differential evolution with three chaotic sequences, INT J ELEC POWER; 2013;England; 47: 85–99.
[11] Pei Y. and Hao J, "Non-dominated sorting and crowding distance based multi-objective chaotic evolution", ICSI 2017; Japan, pp. 15-22,
[12] Tanyıldızı E, Demir G. Golden Sine Algorithm: A Novel Math-Inspired Algorithm, ADV ELECTR COMPUT EN 2017.
[13] Eröz E. Yeni Çok Amaçlı Optimizasyon Algoritması: MOGoldSA, Fırat Üniversitesi Teknoloji Fakültesi Yazılım Mühendisliği Yüksek Lisans Tezi, 2020
[14] Mirjalili S, Gandomi AH. Chaotic gravitational constants for the gravitational search algorithm. APPL SOFT COMPUT 2017. 53: 407-419.
[15] Zawbaa HM, Emary E, Grosan C (2016) Feature Selection via Chaotic Antlion Optimization. PLoS ONE 11(3): e0150652. doi:10.1371/journal. pone.0150652
[16] P. J. S. S. Mirjalili, “Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems,” Springer Science+Business Media, 10.1007/s10489-016-0825-8, 2016.
[17] Vohra R, Patel B. An Efficient Chaos-Based Optimization Algorithm Approach for Cryptography. Communication Network Security. 2012;1(4):75–79.
[18] Ren B, Zhong W. Multi-objective optimization using chaos based PSO. Information Technology. 2011;10(10):1908–1916.
[19] Eröz, E., Tanyıldızı, E., (2020). Kaotik Haritalı Çok Amaçlı Altın Sinüs Algoritmasının Performans Analizi. Fırat Üniversitesi Mühendislik Bilimleri Dergisi, 32(2),391-402.
[20] Deb K. Multi-objective optimization using evolutionary algorithms. New York: John Wiley&Sons, 2001.
[21] Van Veldhuizen, D. A. and Lamont, G. B. Multiobjective evolutionary algorithm research: A history and analysis. Technical Report TR-98-03, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, WrightPatterson AFB, Ohio, 1998