Emperor Journal of Civil Engineering

Received : January 2022
Accepted : January 2022
Published : February 2022

www.ejceng.mayas.info

Volume - I Issue – 2 May - 2022

Annealed Grey Wolf Optimization

Saniya Bahuguna[1] , Ashok Pal[2]

Department of Mathematics

Chandigarh University, Gharuan-Mohali

ABSTRACT

A neoteric hybrid algorithm focused on Grey Wolf Optimizer (GWO) and Improved modified Simulated Annealing Algorithm called Annealed-Grey Wolf Optimizer is suggested here to gain advantages and overcome their disadvantages. Grey wolf optimizer is a metaheuristic swarm intelligence method and has been used widely due to its superb characteristics which take it one step ahead than other swarm intelligence methods as it efficiently solves various optimization problems. With various advantages it also has a disadvantage of bad local searching ability and here in this paper it is tried to overcome by simulated annealing as its major edge is to escape becoming captured in local minima.

Keyword: Grey wolf optimizer, Simulated annealing, Improved modified Simulated annealing, optimization, mtaheuristics

I. INTRODUCTION

Actual-world optimizing challenges are always really difficult to fix, and countless systems have to struggle with NP-hard problems. Optimization techniques must be used to resolve these issues, but there is no assurance that the optimum solution will be attained. In reality there have been no efficient algorithms for NP problems. As a consequence, several problems need to be fixed by trials and error utilizing different techniques of optimizations. Therefore, novel algorithms have been introduced to see if these complex optimization problems can be tackled. Under many ways, nature has influenced many scholars, and is thus a rich source of inspiration. Many recent algorithms nowadays are inspired with nature. Those are metaheuristic techniques which do not depend upon gradient information about the objective function and the constraints, and can use probabilistic concepts of transformation. These are centered on stochastic inquiry strategies, making them successful and adaptable. Yet it is still pretty hard to detect a perfectly effective clarification for almost every question due to no free lunch rule (Wolpert & Macready, 1997)[1]. In this manner, researchers and engineers around the globe are still during the time spent finding more optimization algorithms and increasingly applicable techniques. A few metaheuristics have been created to handle worldwide optimization issues, for instance, Particle Swarm Optimization (PSO) (Kennedy & Eberhart, 1995)[2], Genetics Algorithms (GA) (Goldberg & Holland, 1988)[3], Simulated Annealing (SA) (Kirkpatrick, Gelatt, & Vecchi, 1983)[4], Harmony Search (HS) (Geem, Kim, & Loganathan, 2001)[5], Ant Colony Optimization (ACO) (Dorigo, Maniezzo, & Colorni, 1996)[6], Cuckoo Search (CS) (Yang & Deb, 2010)[7], Colliding Bodies Optimization (CBO) (Kaveh & Mahdavi, 2014)[8], Water Wave Optimization (Zheng, 2015)[9], Big Bang–Big Crunch (Erol & Eksin, 2006)[10], Spider Monkey Optimization (Bansal, Sharma, Hazrati, & Clerc, 2014)[11], Dolphin Echolocation (DE) (Kaveh & Farhoudi, 2013)[12], Ray Optimization (RO) (Kaveh & Khayatazad, 2012)[13], Charged System Search (Kaveh & Talatahari, 2010)[14] among others.

Grey Wolf Optimizer (Mirjalili, Mirjalili, & Lewis, 2013) [15]is yet another novel metaheuristic strategy which solves problems of global optimization. The GWO algorithm examines the following, chasing, and social chain of authorization of the grey wolves. It empowers its search agents to refresh their position dependent on the position of the alpha (best solution), beta (second best), and delta (third best); and attack the prey. Numerous studies have shown that GWO is excellent at exploitation (Long, Jiao, & Tang, 2018)[16]; (Saxena, Kumar, & Das, 2019)[17]; (Gupta & Deep, 2019)[18]. However, with these operators, it is likely to stagnate in local solutions. To some degree, the encircling mechanism in it indicates exploration, but GWO needs more operators to underline exploration.

So it is hybridized with Improved and Modified Simulated annealing (Suarez, Millan, & Millan, 2018)[19] to tackle the abovementioned shortcomings of the traditional GWO. The classical SA is another great metaheuristic methodology that draws the inspiration from heating and then slows cooling of metals in metallurgy. The entire space of solution is investigated and the current solution is recouped with the worse ones with less probability and thus improves the likelihood of avoiding the local optima (Lim, Rodrigues, & Zhang, 2006)[20]; (Szu & Hartley, 1984)[21]. An Improved Modified Simulated Annealing Algorithm (I-MSAA) is used in this study with the goal of improving the solution performance. The functioning of simple Simulated Annealing (SA) and Grey Wolf Optimizer is briefly defined before summing up the characteristics of the Annealed Grey Wolf Optimizer (A-GWO).

Simulated Annealing

Annealing is one kind of a metallurgical heat treatment process where metal is heated to a predefined temperature and then allowed to cool. While cooling, the cooling rate can be different i.e. it will be fast or it will be slow, and depending upon the rate of cooling we get the final structure in the material heated with different properties and in this annealing the rate of cooling is the slowest. As we heat up the metal we give it the heat energy and all the molecules or particles inside the metal gets energized, i.e. the energy level goes up to a higher level, but as we cool it down the energy level of the particles gets reduced and thereby it finally comes to a state of equilibrium where the free energy state is at the minimum possible level and thus the process ends. In simpler words, we can say that the atoms in the melted phase will move freely concerning each other at specific elevated temperature, but as the temperature is decreased the movement is confined. In the event that the heating temperature is enough high to ensure irregular state and cooling course is amply moderate to assure thermal balance, then atoms have just enough time to organize themselves in a configuration that leads to the minimum energy state, i.e. a singular perfect crystal that is our ultimate goal to achieve in the annealing process as in the optimization process, our main aim is to reduce the value of the objective function if it is a question of minimization. If the rate of cooling is fast it ends up with polycrystalline state that has higher energy than the single crystalline state which is not our ultimate goal. Crystal formation depends on cooling rate and for an end state with minimum energy cooling must be slow.

Simulated annealing (SA) is a non-traditional optimization search technique where we use probabilistic laws so it is also a stochastic optimization technique where it mimics the annealing process and thus it got its name as simulated annealing. It was first developed by (Metropolis, Rosenbluth, Rosenbluth, & Teller, 1953)[22], and was later used by Kirkpatrick et.al. in 1983 to solve optimization problems. It is higher compared to the hill-climbing technique; the only different thing is that simulated annealing also allows a downward step or transition to weaker solutions as well through a series of random moves. It thus removed all the problems of the hill-climbing method as it would lead us to a local point rather than a global point. Here during initial iterations, SA randomly searches the entire feasible zone during the initial iterations and narrows it down to the regions containing the global optima.

Simulated annealing starts from a point which is selected arbitrarily, and here the core principle is that is move from point to point. So when it moves to the next point it can either be a good move or a bad move, good as in if the value of the objective function is getting minimized for a minimization function compared to the value of the objective function at the previous point, and thus we accept that new move and in that new point the free energy state is lesser than the previous point. But suppose the energy level in case of decreasing it increases i.e. if we move to the next point the value of the objective function is getting maximized we will still sometimes accept that bad move. In this not all bad moves are rejected and some of the bad moves are still accepted thus later on giving us a global optimum solution. Bad moves that gets accepted is decided a Boltzman probability distribution (P).

$$P = exp\left(-\frac{\Delta E}{kT}\right)$$

ΔE Is the change of objective function by two consecutive iterations, k is the boltzman constant and T is the control parameter which is analogously known as temperature for annealing. However this probability is not equivalent in all circumstances. P actually depends on ΔE and T. If T is large, the probability is more and any point will be accepted for large value of T, and when T decreases the chances of an arbitrary point to get accepted becomes less. The Pseudo code of classical SA is as follows:

Pseudo Code of Simulated Annealing:

1) Select the initial solution val
2) Initialize the parameters:

    Initial Temperature= $Tinit$

    Final Temperature= $Tfin$

    Boltzman's Constant k

    Reduction factor c
3) Set maximum number of perturbation at same temperature $iter_{max}$
4) $T = Tinit$
5) While $(T > Tfin)$
6) $for\ iter = 1\ to\ iter_{max}$
7) Generate new value as $newval$ in the neighborhood of val
8) $if\ (newval \leq val)$

    $val = newval$

    else

    Calculate difference between $newval$ and val , (ΔE)

    $Boltz\_Prob = exp(-\Delta E/kT))$

    $if(Boltz\_Prob > random(0,1))$

    $val = newval$

    end

    end

    end

    Decrease T by cooling schedule: $T = c * T$

    end
9) Return the best solution found

Grey Wolf Optimizer

GWO is a newly discovered swarm intelligence algorithm presented by Mirjalili et. al, in 2014, influenced by the grey wolves' social framework and attacking behavior. There are four levels: alpha (α), beta (β), delta (δ) and omega (ω) indexed by descending into force, in which the alpha wolf is the controller of the group, and ω wolves are the undermost among the unit ranking.

The key foraging procedure for grey wolf is as follows:

a) Constantly monitoring, outrunning and inching closer to the prey.
b) The target is tracked, blockaded and intimidated until it stops moving.
c) Attempting to kill the prey.

The social chain of command of the wolves is demonstrated numerically while structuring GWO is as follows, best solution out of the considerable number of solutions is cited as alpha (α), the following best is allotted as beta (β), third-best alludes as delta (δ) and all the rest of the solutions are accepted to be omegas.

The equations proposed for the mathematical model of encircling of grey wolves is as follows:

$$\vec{D} = \left|\vec{C}.\vec{X}_p(it) - \vec{X}(it)\right| \quad (1)$$

$$\vec{X}(it+1) = \vec{X}_p(it) - \vec{A}.\vec{D} \quad (2)$$

Where it is the current iteration, $\vec{A}$ and $\vec{C}$ are coefficient vectors, $\vec{X}_p$ position vector of the victim $\vec{X}$ is the position vector of the grey wolf.

Coefficient vectors are calculated as:

$$\vec{A} = 2\vec{a}.\vec{s}_1 - \vec{a} \quad (3)$$

$$\vec{C} = 2.\vec{s}_2 \quad (4)$$

Here s_1 and s_2 are random vectors in [0,1] and $\vec{a}$ is linearly decreased from 2 to 0 during iterations.

To display mathematically the chasing demonstration we assume that maybe the alpha, beta, and delta have detailed information on the conceivable objective area. We select the initial three best solutions that we have up until now and update the position of other search agents as indicated by the situation of the elite search agent found.

Mathematically it is proposed as follows:

$$\vec{D}_\alpha = \left|\vec{C}_1.\vec{X}_\alpha - \vec{X}\right| \; ; \; \vec{D}_\beta = \left|\vec{C}_2.\vec{X}_\beta - \vec{X}\right| \; ; \; \vec{D}_\delta = \left|\vec{C}_3.\vec{X}_\delta - \vec{X}\right| \quad (5)$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1.(\vec{D}_\alpha) \; ; \; \vec{X}_2 = \vec{X}_\beta - \vec{A}_2.(\vec{D}_\beta) \; ; \; \vec{X}_3 = \vec{X}_\delta - \vec{A}_3.(\vec{D}_\delta) \quad (6)$$

$$\vec{X}(it+1) = \frac{X_1 + X_2 + X_3}{3} \quad (7)$$

Alpha, beta, and delta anticipate the prey's area, and different wolves arbitrarily adjust their situations around the prey. Grey wolves finish up chasing by executing the prey with regards to a stop. By decreasing the value of $\vec{a}$ it mathematically models attacking of prey and simultaneously varying the range of $\vec{A}$. The estimation of $|A| < 1$ compels the wolves to assault the prey.

Proposed Method

Within this research a fresh hybrid is presented to improve the GWO calculations. Exploration is intended to assess the elite region of the search space, and exploitation is needed throughout this zone to come up with best solutions (Chen, Xudiera, & Montgomery, 2012)[23]. There is a well acknowledge fact that exploration and exploitation are both necessary to demonstrate incredible performance in any algorithm based on the population.

In contemporary GWO, the key issue is that all agents (wolves) that conduct the search are upgraded in the complete optimization mechanism as per the α (Ist best search agent), β (IInd best search agent) and δ (IIIrd best search agent) as seen in Eq.(7). Essentially, this role updating procedure prompts premature convergence, since search agents couldn't proficiently investigate the search space (Arora, Singh, & Sharma, 2019)[24]. In addition, the method of optimization that was stated in Eq. (7) has restricted potential of exploitation at the later phases of optimization contributing to slower convergence. Therefore, to rectify the fore mentioned shortcomings of the traditional GWO, an improved version of modified SA hybridizes a progressively adequate harmony among exploitation and exploration.

Classical SA is a methodology adapted from statistical thermodynamics to simulate the activity of atomic arrangements during the annealing phase in liquid or solid structures. Its principal advantages over other methodologies of local search are its versatility and ability to achieve global optimality.

In the proposed hybrid algorithm of GWO-SA an effort is made to improve the convergence rate as in the initial iterations exploration is done (Das, Konar, & Chakraborty, 2007)[25]. Here the concept of Simulated Annealing is used to improve the values of first three best solutions i.e. α, β, δ because on that basis other search agents improve their positions. The condition which decides whether or not the worse solution will be approved is determined by a probability which is given by the:

$$P = \frac{1}{1+2\exp(\Delta E/kT)} \tag{8}$$

Here in eq. (8), P the probability of accepting the new state. ΔE is the change in objective value of the current solution and the new solution. T is the system temperature, k is Boltzman Constant and e is the Euler number. The probability is defined in between the range of (0, 1/3).

It lowers the acceptance range of the algorithm of the worst solutions. The pseudo code of the proposed method is as follows:

Pseudo Code of Annealed Grey Wolf Optimizer

1) Create starting populace of search agent.
2) Calculate objective function value for every search agent
3) Select α candidate solution as Initial solution and initialize it as (val)
4) Initialize $maxiteration$
5) Set Initial Temperature $Tinit$
6) Set Final Temperature $Tfin$
7) Initialize Boltzman's Constant k
8) Initialize Reduction factor c
9) Set maximum number of perturbation at same temperature $iter_{max}$
10) $n = 0$
11) While $(n < maxiteration)$
12) $T = Tinit$
13) While $(T > Tfin)$
14) $for\ iter = 1\ to\ iter_{max}$
15) Generate $newval$ in the neighborhood of val
16) $if\ (newval \leq val)$

$val = newval$

else

Calculate difference between $newval$ and val (ΔE)

$Boltz\_Prob = 1/(1 + 2exp(\Delta E/kT))$

$if(Boltz\_Prob > random(0,1/3))$

$val = newval$

end

end

end

Decrease T by cooling schedule: $T = c * T$

$n = n + 1$

end

Return the best solutions found and initialize it to the value of α candidate solution

end

Do the same process as above simultaneously for beta and delta candidate solution as an initial solution, and then continue as it is done for the solution from step (7) from the pseudo code of A-GWO.

II. RESULT

Benchmark problems are utilized in this segment to assess Annealed GWO‘s efficiency. Such problems can be split into three distinct classes: unimodal , multimodal and fixed-dimensional multimodal functions. Annealed GWO was evaluated on 19 benchmark problems (Yao, Liu, & Lin, 1999)[26], seven unimodal, six multimidal and six fixed dimension multimodal problems.

The specific depictions of these testing problems are referenced in table 1 to table 3 in which we got better results.

Each function has been assessed 30 times with 30 search agents and the most extreme number of iterations as 500. The best values and the worst values have been reported in the table 2.

Our key goal here is to convey the best reasonable optimal solution when contrasted with different metaheuristics.

Table 1: Unimodal Benchmark Functions

Function	*Dim*	*Range*	f_{min}
$F_1(x) = \sum_{i=1}^{n} x_i^2$	30	[−100,100]	0
$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	[−10,10]	0
$F_3(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$	30	[−100,100]	0
$F_4(x) = max_i\{\lvert x_i \rvert, 1 \leq i \leq n\}$	30	[−100,100]	0
$F_5(x) = \sum_{i=1}^{n-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2 \right]$	30	[−30,30]	0

$F_6(x) = \sum_{i=1}^{n-1} (\lfloor x_i + 0.5 \rfloor)^2$	30	$[-30,30]$	0
$F_7(x) = \sum_{i=1}^{n} i x_i^4 + rand[0,1)$	30	$[-1.28,1.28]$	0

Table 2: Multimodal Benchmark Functions

Function	*Dim*	*Range*	f_{min}
$F_8(x) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	30	$[-500,500]$	-12569.5
$F_9(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	$[-5.12,5.12]$	0
$F_{10}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=n}^{n} cos2\pi x_i\right)$	30	$[-32,32]$	0
$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$[-600,600]$	0
$F_{12}(x) = \frac{\pi}{n}\left\{10sin(\pi y_i) + \sum_{i=1}^{n-1} (y_i - 1)^2[1 + 10sin^2(\pi y_{i+1}) + (y_n - 1)^2]\right\} + \sum_{i=1}^{n} u(x_i, 10,100,4)$ $y_i = 1 + \frac{x_i + 1}{4}$ $u(x_i,a,k,m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30	$[-50,50]$	0

$F_{13}(x) = 0.1\left\{sin^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i - 1)^2[1 + sin^2(3\pi x_{i+1})] + (x_n - 1)^2[1 + sin^2(2\pi x_n)]\right\} + \sum_{i=1}^{n} u(x_i, 5,100,4)$	30	[−50,50]	0

Table 3: fixed dimension multimodal benchmark problems

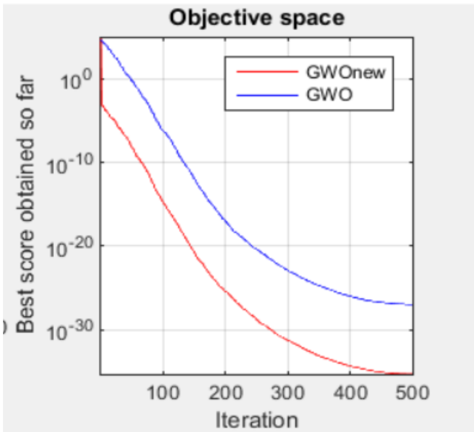
Function	*Dim*	*Range*	f_{min}
$F_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6}\right)^{-1}$	2	[−65.536,65.536,]	1
$F_{15}(x) = -\sum_{i=1}^{4} c_i exp\left[-\sum_{j=1}^{4} a_{ij}\left(x_j - p_{ij}\right)^2\right]$	3	[0,1]	−3.86
$F_{16}(x) = -\sum_{i=1}^{4} c_i exp\left[-\sum_{j=1}^{6} a_{ij}\left(x_j - p_{ij}\right)^2\right]$	6	[0,1]	−3.32
$F_{17}(x) = -\sum_{i=1}^{5}[(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	[0,10]	−10.
$F_{18}(x) = -\sum_{i=1}^{7}[(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	[0,10]	−10
$F_{19}(x) = -\sum_{i=1}^{10}[(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	[0,10]	−10"

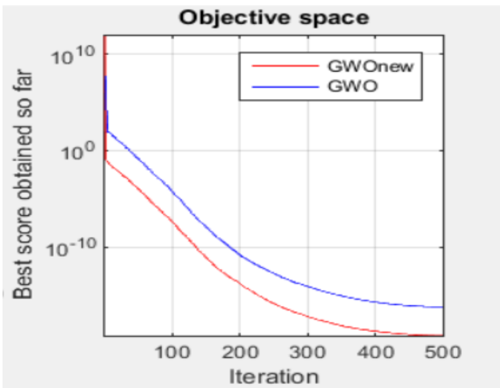
Table 4 shows the performance of Annealed Grey Wolf Optimizer, and is compared to the original GWO.

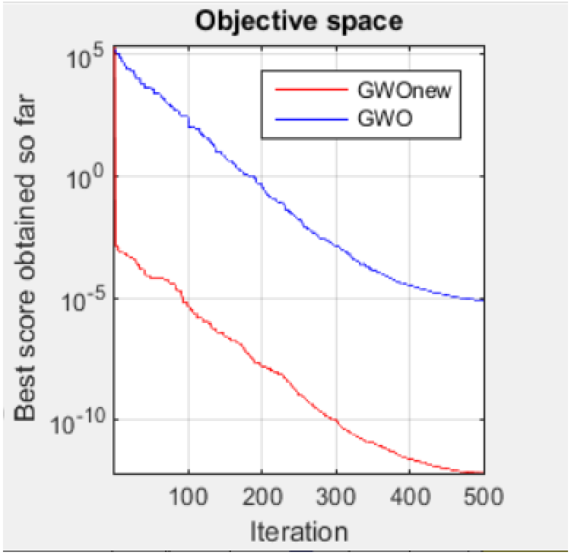
Table 4

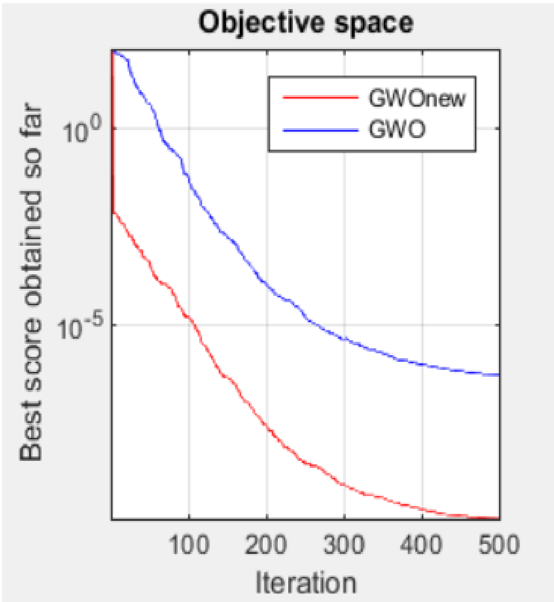
Function	GWO		A − GWO	
	Best	Worse	Best	Worse
F_1 (a)	$1.182e^{-28}$	$7.694e^{-27}$	$9.092e^{-37}$	$5.975e^{-35}$
F_2 (b)	$1.298e^{-17}$	$4.346e^{-16}$	$5.452e^{-20}$	$2.372e^{-19}$
F_3 (c)	$9.782e^{-8}$	$1.502e^{-5}$	$1.581e^{-16}$	$6.523e^{-14}$
F_4 (d)	$3.802e^{-7}$	$4.627e^{-6}$	$1.322e^{-11}$	$2.789e^{-10}$
F_5 (e)	26.136	28.019	25.5728	26.165
F_6 (f)	0.251	1.251	0.201	1.766
F_7 (g)	0.000495	0.00350	$1.599e^{-6}$	$4.705e^{-4}$
F_8 (h)	−7029.366	−3395.181	−7625.465	−3589.212
F_9 (i)	$1.1368e^{-13}$	9.066	0	$5.684e^{-14}$
F_{10}	$6.838e^{-14}$	$1.572e^{-13}$	$3.386e^{-14}$	$4.352e^{-14}$

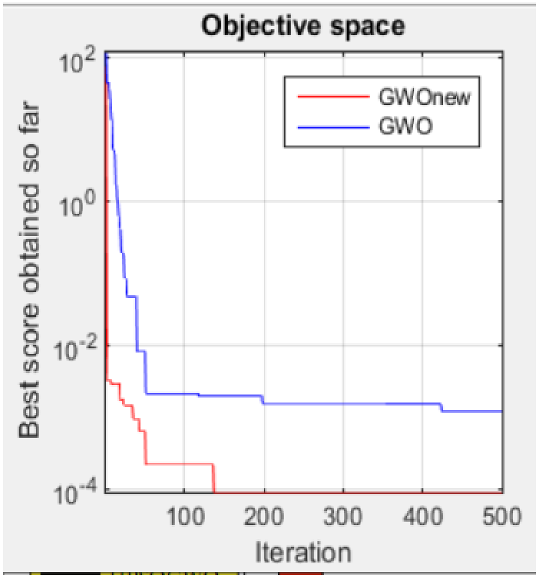
(j)				
F_{11} (k)	0	0.04027	0	0.0257
F_{12} (l)	0.0162	0.0847	0.0127	0.0777
F_{13} (m)	0.3896	1.321	0.3186	0.8265

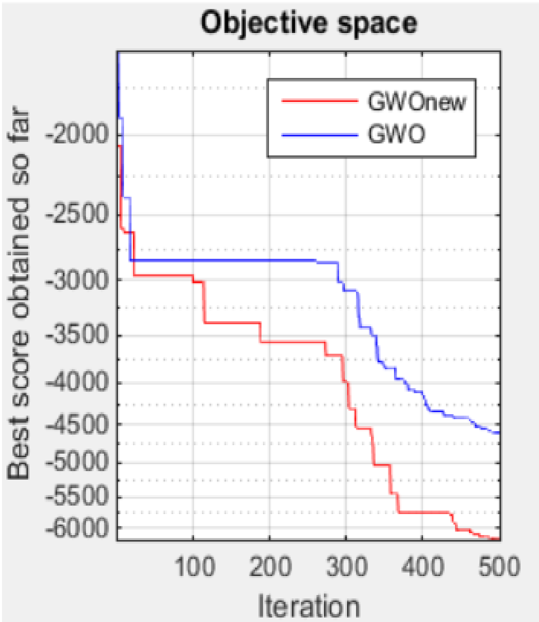


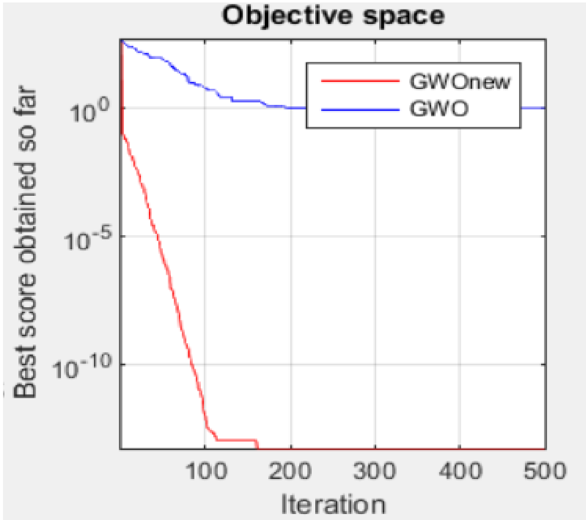


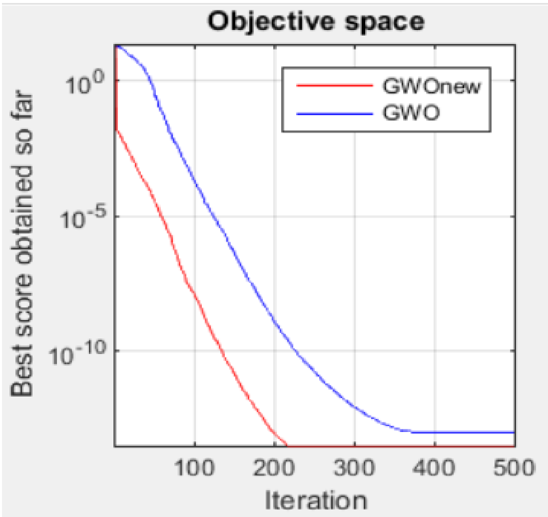

Objective space
GWOnew
GWO
Best score obtained so far
10^5
10^0
10^-5
10^-10
100
200
300
400
500
Iteration


Objective space
GWOnew
GWO
Best score obtained so far
10^0
10^-5
100
200
300
400
500
Iteration

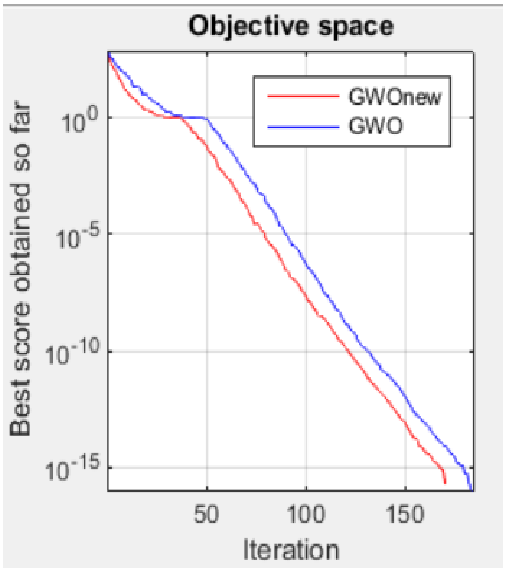

Objective space
GWOnew
GWO
Best score obtained so far
Iteration
10^2
10^0
10^-2
10^-4
100
200
300
400
500

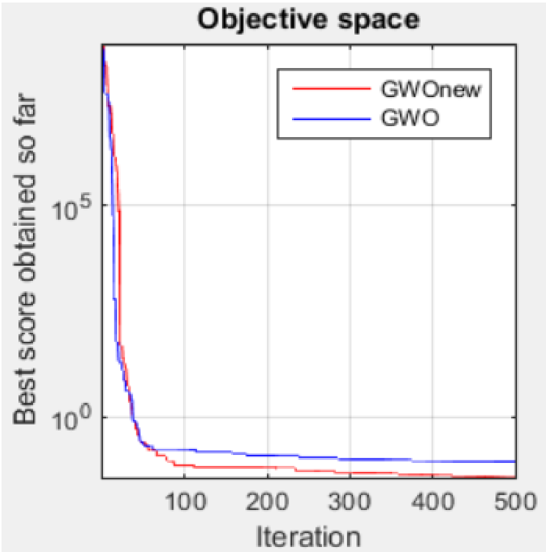

Objective space
GWOnew
GWO
Best score obtained so far
Iteration
-2000
-2500
-3000
-3500
-4000
-4500
-5000
-5500
-6000
100
200
300
400
500

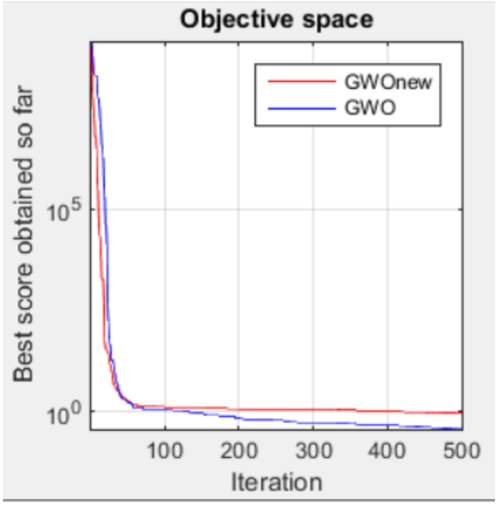

Objective space
GWOnew
GWO
Best score obtained so far
Iteration
10^0
10^-5
10^-10
100
200
300
400
500


Objective space
GWOnew
GWO
Best score obtained so far
Iteration
10^0
10^-5
10^-10
100
200
300
400
500

)


Objective space
GWOnew
GWO
Best score obtained so far
10^{0}
10^{-5}
10^{-10}
10^{-15}
50
100
150
Iteration


Objective space
GWOnew
GWO
Best score obtained so far
10^{5}
10^{0}
100
200
300
400
500
Iteration



III. CONCLUSION

A new hybrid version of Grey Wolf Optimizer with Improved Modified Simulated Annealing named Annealed Grey Wolf Optimizer (A-GWO) for optimizing global optimization problems has been introduced in this work. The main idea behind this paper was to lay more emphasis on exploration and local search ability in Grey Wolf Optimizer. Here in A-GWO, the values of alpha, beta, and delta operators are improved because based on these three operators other search agents improve their positions. These three operators are improved by introducing the concept of Simulated Annealing, as here the alpha, beta, and delta are considered as the starting point separately and the probability of accepting the worst solution is also reduced.

Annealed-GWO, as appeared, is a truly steady algorithm, fit for accomplishing dependable, sensible execution. The algorithm seeks the global optimum in various functions as per robustness. Annealed-GWO can thus be regarded as a robust optimization algorithm. Annealed-GWO is a good optimization algorithm that can be applied to practical engineering related issues.

IV. REFERENCE

1. Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for Optimization. *IEEE Transactions on Evolutionary Computation* .
2. Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *International Conference on Neural Networks*. IEEE
3. Goldberg, D. E., & Holland, J. H. (1988). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co.

4. Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science* .

5. Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A New Heuristic Optimization Algorithm: Harmony Search. *SAGE Journals* .

6. Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE* .

7. Yang, X. S., & Deb, S. (2010). Engineering Optimisation by Cuckoo Search. *International Journal of Mathematical Modelling and Numerical Optimisation* .

8. Kaveh, A., & Mahdavi, V. R. (2014). Colliding bodies optimization: A novel meta-heuristic method. *Computers & Structures* .

9. Zheng, Y. (2015). Water wave optimization: A new nature-inspired metaheuristic. *Computers & Operations Research* .

10. Erol, O. K., & Eksin, I. (2006). A new optimization method: Big Bang–Big Crunch. *Advances in Engineering Software,* .

11. Bansal, J. C., Sharma, H., Hazrati, G., & Clerc, M. (2014). Spider Monkey Optimization algorithm for numerical optimization. *Memetic Computing*

12. Kaveh, A., & Farhoudi, N. (2013). A new optimization method: Dolphin echolocation. *Advances in Engineering Software* .

13. Kaveh, A., & Khayatazad, M. (2012). A new meta-heuristic method: Ray Optimization. *Computers and Structures* .

14. Kaveh, A., & Talatahari, S. (2010). A novel heuristic optimization method: charged system search. *Acta Mechanica* .

15. Mirjalili, S., Mirjalili, M. S., & Lewis, A. (2013). Grey Wolf Optimizer. *Advances in Engineering Software* .

16. Long, W., Jiao, J., & Tang, M. (2018). An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization. *Engineering Applications of Artificial Intelligence* .

17. Saxena, A., Kumar, R., & Das, S. (2019). β-Chaotic map enabled grey wolf optimizer. In *Applied Nature-Inspired Computing: Algorithms and Case Studies.*

18. Gupta, S., & Deep, K. (2019). An Efficient Grey Wolf Optimizer with Opposition-Based Learning and Chaotic Local Search for Integer and Mixed-Integer Optimization Problems. *Arabian Journal for Science and Engineering* .

19. Suarez, J., Millan, C., & Millan, E. (2018). Improved Modified Simulated Annealing. *Contemporary Engineering Sciences* .

20. Lim, A., Rodrigues, B., & Zhang, X. (2006). A simulated annealing and hill-climbing algorithm for the traveling tournament problem. *European Journal of Operational Research* .

21. Szu, H., & Hartley, R. (1984). Fast simulated annealing. *Physics Letters A* .

22. Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., & Teller, A. H. (1953). Equation of State Calculations by Fast Computing Machines.

23. Chen, S., Xudiera, C., & Montgomery, J. (2012). Simulated Annealing with Thresheld Convergence. IEEE.

24. Arora, S., Singh, H., & Sharma, M. (2019). A New Hybrid Algorithm Based on Grey Wolf Optimization and Crow Search Algorithm for Unconstrained Function Optimization and Feature Selection. *IEEE Access* .

25. Das, S., Konar, A., & Chakraborty, U. K. (2007). Annealed Differential Evolution.

26. Yao, X., Liu, Y., & Lin, G. (1999). Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation* .