Grenze International Journal of Engineering and Technology, Jan Issue

Exploration and Comparative Implementation of Hybrid GWOPSO Algorithm against GWO

Rajbhupinder Kaur[1], Dr. Vijay Laxmi[2] and Dr. Bal Krishan[3]
[1]Research Scholar (Ph.D.), Guru Kashi University, Talwandi Sabo, Punjab
er.rajbhupinder@gmail.com
[2]Professor, Guru Kashi University, Talwandi Sabo, Punjab
[3]Assistant Professor, Yadavindra College of Engineering, Talwandi Sabo, Punjab

Abstract*—Whenever humans have tried to explore and overcome nature, a lot of glitches with gigantic complexities having no investigative solutions have been encountered. It has always been hard to find a universal solution to all the problems. With the snowballing prerequisite of exhausting computing resources as a non-scale bounded substructure, the swing towards cloud computing has witnessed an exponential increase to decrease the total cost. The developers make use of shared resources in a cloud environment. Exceptional algorithms are required to map resources with the demands of the users. To offer users effective solutions, more complex efficient algorithms are needed. GWO (Grey Wolf Optimizer) is one of the listed metaheuristic methods together with PSO (Particle Swarm Optimization). The research paper is one such attempt to enhance the performance of the GWO algorithm via proposing a hybrid GWOPSO algorithm comprising the effective features of both GWO and PSO algorithms. The research paper discusses the leadership hierarchy of grey wolves adopted to hunt the prey finding the best optimal solution. The research paper elaborates on the working principle of both GWO and hybrid GWOPSO algorithm via a mathematical application, flowcharts, and algorithms. The primary objective of the proposed research is to minimize the execution time and optimize the use of available resources to obtain an optimal solution. The implementation has been conducted to record the values of performance evaluation parameters like waiting time, turnaround time, throughput, the best solution, and best optimal values and it has been found that the proposed hybrid PSOGWO algorithms scores over the GWO algorithm.

***Index Terms*— Grey wolves, GWO, Hybrid GWOPSO, turnaround time, waiting time.**

I. Introduction

In recent years an adequate amount of literature has been investigated and several authors have proposed innovative meta-heuristic algorithms. These algorithms play a very important role in solving multifaceted optimization problems and so search space turns out to be a significant issue in such complications. Search space upsurges tremendously when the facet of the problem is amplified. Certain problem functions have perpetual magnitudes. Meta-heuristic algorithms are preferred because of their capability of providing reasonable solutions at an acceptable time [1]. It is for sure that meta-heuristic algorithms may not always offer the best explanations and even in rare circumstances, delivered solutions may not be satisfactory. The performance of the constructed algorithm may differ as per the problem. An algorithm proposing an effective solution to a problem may not accomplish the desired goal for an alternative problem at the same time [2, 3]. Meta-heuristic algorithms perform

Grenze ID: 01.GIJET.7.1.506_2
© Grenze Scientific Society, 2021

established on random inputs and expected outputs. Meta-heuristic algorithms are categorized as single based solutions or multi-based solutions. In the single-based algorithm there is a single solution along with search iteration whereas, in a multi-based algorithm, the whole of the population has an impact on the output [4, 5]. Some characteristics of meta-heuristic algorithms are a practical explanation, satisfactory response time, unpredictability, and numerous solutions for a problem in every run. The studies pertinent to the population-based procedure have been piloted more. These find their applications in areas like machine learning, engineering, industry, system modeling, etc. The primary aim of meta-heuristic algorithms is to solve problems at a faster rate, handle large and complex problems, and finding an optimized solution [6]. Meta-heuristic algorithms are imitated from the family of optimization algorithms and are categorized into two groups of exact and appropriate algorithms. Exact algorithms offer an optimum solution with precision but are not good enough for managing hard optimization snags and often their execution time upsurges with the magnitudes of the problem. On the other hand, approximate algorithms are good in finding optimal solutions in a short time for hard optimization problems [7, 8]. Meta-heuristic algorithms evade deceiving in local optima via exploitation and exploration concepts. Exploration refers to finding the best solutions in the concerned domain and exploitation refers to focusing on the best solution areas to obtain the best solution [9, 10].

Meta-heuristic algorithms are categorized into three categories: evolution-based (EA) algorithms, physics-based (PB) algorithms, and swarm intelligence (SI) algorithms. The SI algorithms are whose working is based on group performances. These algorithms are embraced by standardized members intermingling with each other and their environment. These agents collaborate in the local search space and jointly make effort to touch convergence close to the best solution. Particle Swarm Optimization (PSO) is the utmost prevalent algorithm in this group. PSO is motivated by the social behavior of birds. In PSO, there are communication channels amid particles that keep moving on search space. The best solution is determined by the Fitness function. Grey Wolf Optimization (GWO) is another algorithm that falls in the category of SI algorithms. GWO focuses on the hunting mechanism of the grey wolves. The functioning of GWO depends on the leadership hierarchy [11, 12]. GWO finds its applications in task scheduling and loads balancing in cloud computing, image segmentation, medical diagnosis, clustering in vehicular networks, etc.

II. State Of Art

In 2019, Gobalakrishnan Natesan et al. [13] focused on the primary objective of task scheduling to be planning the allotment of tasks on resources and curtailing the objective of the schedule. The authors proposed a Mean Grey-Wolf optimization algorithm to improve the performance of the system and curtailing the scheduling issues. The focus is laid on curtailing energy consumption and makespan. The authors made use of the CloudSim tool. The authors used two datasets; left-skewed and right-skewed. The obtained results proved the effectiveness of the Mean GWO algorithm over existing algorithms like PSO and standard GWO. The authors suggested considering parameters like load balancing, execution time, and security for future work. In 2019, N. M. Hatta et al. [14] stated that looking for a solution to real-world difficulties concentrating on combinatorial problems is a vital job. Optimization techniques are the one which can offer a feasible paramount solution for a precise problem. The authors focused on the working of Grey Wolf optimization as providing a solution to many engineering problems. The authors elaborate on the list of solutions as alpha, beta, and delta and also on the hunting technique of the grey wolves comprising tracking, encircling, and attacking the prey to find the best optimal solution. The authors considered 83 publications having relevance with GWO in several applications such as economy, parameter tuning, cost estimating, dispatch problem, and many more. A conversation is elaborated on the properties of the GWO algorithm and how it diminishes the diverse difficulties in the dissimilar applications and an investigation on the research trend of the GWO optimization method in numerous applications from the year 2014 to 2017 is conducted. Grounded on the literature, it was supposed that GWO can resolve single and multi-objective glitches capably due to its effective local search standards that execute remarkably fine for diverse snags and explanations. In 2019, Zheng-Ming Gao and Juan Zhao [15] proposed an improved Grey Wolf Optimization (GWO) algorithm assuming that the social hierarchy of the grey wolves would likewise be shadowed in their searching positions. The authors conducted simulation experiments to justify the conducted research. The inconstant weights are assigned to incisive positions. To curtail the probabilities of being stuck in local optima, the proposed algorithm Variable Weights - Grey Wolf Optimization (VM-GWO) offered better results when compared to standard GWO, PSO (Particle Swarm Optimization), ALO (Ant Loin Optimization), and BA (Bat Algorithm). To reduce the probability of being trapped in local optima, a prevailing equation of the monitoring parameter is presented, and thus, it deteriorates exponentially from the maximum. VW-GWO is predicted to be effective in solving broad complications. Although the anticipated developments are predominantly focusing on the capability to converge, it leads to quicker convergence and extensive applications. But it is not effective and proficient for all the benchmark functions. The authors suggested that the details can be expressed mathematically in future work. In 2020, Zhihang Yue et al. [16] elaborated on GWO (Grey Wolf Optimization) and FWA (Fireworks) algorithm. The authors applauded the strong and optimal searchability of both algorithms. But it is often observed that GWO converges to the local optimum and FWA is a bit slow. The research conducted proposes an innovative algorithm FWGWO which is a mix of benefits of these two algorithms. The proposed algorithm allies the exploration capability of the fireworks algorithm with the exploitation capability of the grey wolf optimizer (GWO) by fixing a balance coefficient. The authors made use of 16 benchmark functions with a wide diversity of complexities and dimensions. The obtained results are compared with standard FWA, the standard GWO, EGWO (Enhanced Grey Wolf Optimization), and AGWO (Augmented Grey Wolf Optimization) algorithms. The experimental results demonstrate that the FWGWO commendably increases the global optimal searchability and convergence speed of the GWO and FWA.

III. Leadership Hierarchy of Grey – Wolf Optimization (GWO)

Several meta-heuristic algorithms dedicated to deal with optimization complications came into existence in recent years and Grey wolf optimization is one such algorithm. Grey wolf algorithm enjoys resilient optimal research competence [17, 18]. The working principle behind the GWO algorithm is analogous to the hunting mechanism and leadership pyramid of wolves to explore the targets [19, 20]. Grey wolves prefer remaining in groups with each group comprising 5 to 12 wolves on average. The four levels of wolves' hierarchy are mentioned below.

- Alpha (α) - This is the first level hierarchy. Its job is to make decisions related to hunting, find a place to sleep, and plan walk time, etc.
- Beta (β) – The job of beta wolves is to assist alpha candidates in making decisions or engaging in supplementary actions.
- Delta (δ) – Delta wolves works under the expertise of alpha and beta wolves. These are accountable for inspection, patrol, lookout, and other responsibilities.
- Omega (ω) - Omega wolves have to defer to the alpha, beta, and delta wolves. The task of omega wolves is to preserve the integrity of the hierarchical structure.

The grey wolves are smart enough to surround the prey with the capability to pinpoint its location. Alpha heads the complete hunting process. Often in complex situations, locating the prey at the beginning becomes cumbersome. The first three solutions of alpha, beta, and delta carry more information concerned with the position of the prey. The other wolves update their positions based on positions of alpha, beta, and delta.

IV. Implementation of GWO

This section elaborates the mathematical enactment, few terminologies, algorithm, and implementation of GWO to calculate different performance evaluation parameters like waiting time, turnaround time, throughput, and best optimal solution.

A. Mathematical Implementation of GWO

The grey wolf encompasses the prey when hunting. The mathematical equations for this behavior are mentioned below.

$$E = |N. W_p (c) - W (c)| \text{ and } W (c+1) = W_p (c) - M.E$$

where
E – Denotes to the distance amid a wolf and the prey
W – Denotes to grey wolf's position vector
Wp – Denotes to prey's position vector
c – Denotes to the in-progress iteration
M and N – Denotes to the coefficient vectors which are calculated as

$$M = 2b.v_1 - b \text{ and } N = 2v_2$$

where b declines linearly with the number of iterations from 2 to 0. $v1$ and $v2$ are the random vectors in [0,1].

B. Flowchart

Fig. 1 illustrates the working of the GWO algorithm.

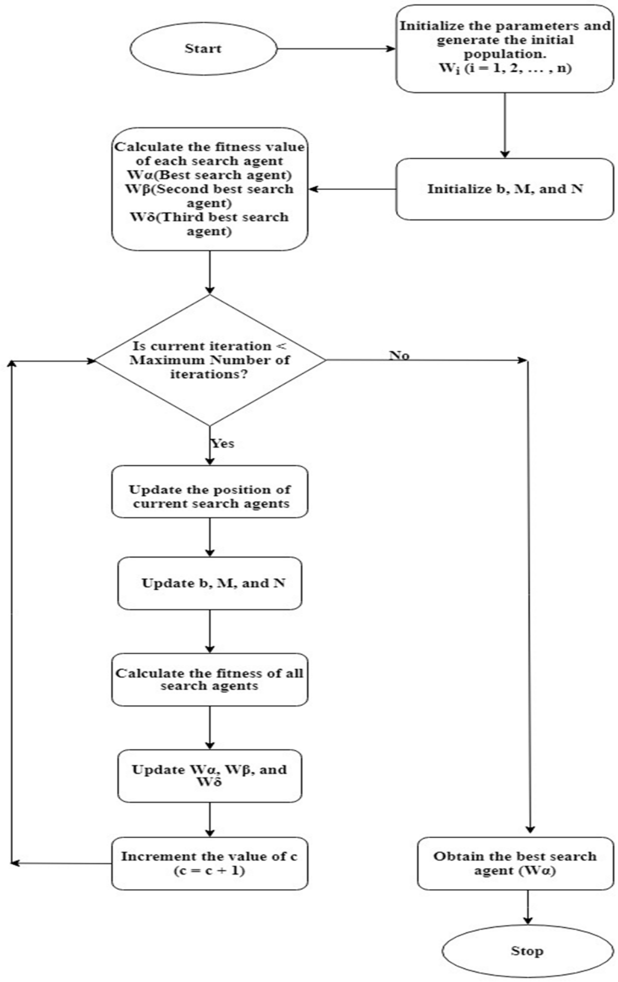

Fig. 1. Working of GWO algorithm

C. Algorithm of GWO

The working algorithm for GWO is mentioned below.

- Initialize the wolf population W_i (i = 1, 2, … , n)
- Initialize b, M, and N
- Calculate the fitness of each search agent
  Wα = Denotes the best search agent, Wβ = Denotes the second-best search agent, Wδ= Denotes the third-best search agent
- while (c < Max number of iterations)
      for each search agent
          Apprise the position of the current search agent
      end for
  Update b, M, and N
  Calculate the fitness of all search agents
  Update Wα, Wβ, and Wδ
  c = c + 1
  end while
  return Wα

D. Implementation - GWO (Grey Wolf Optimization)

The different parameters considered with assigned values to execute the working of GWO are given below.

Case 1:

Number of search agents (SearchAgents_no) = 30
Test functions that ranges from F1 to F23 (Function_name) = 'F22'
Maximum Number of iterations (Max_iteration) = 200
The readings obtained after the execution of GWO as per Case 1 are mentioned below.
Iteration Number = 1.000000 Burst time = 0.605476 Waiting time = 0.000000 Turnaround time = 0.605476
Iteration Number = 2.000000 Burst time = 0.614578 Waiting time = 0.605476 Turnaround time = 1.220054
Iteration Number=3.000000 Burst time = 0.616634 Waiting time = 1.220054 Turnaround time = 1.836688
Iteration Number=4.000000 Burst time = 0.618442 Waiting time = 1.836688 Turnaround time=2.455130
Iteration Number=5.000000 Burst time = 0.620617 Waiting time = 2.455130 Turnaround time=3.075747
.
.
Iteration Number = 195.000000 Burst time = 0.945400 Waiting time = 151.794260 Turnaround time = 152.739660
Iteration Number = 196.000000 Burst time= 0.946524 Waiting time = 152.739660 Turnaround time = 153.686183
Iteration Number = 197.000000 Burst time = 0.947760 Waiting time = 153.686183 Turnaround time = 154.633943
Iteration Number=198.000000 Burst time = 0.949271 Waiting time = 154.633943 Turnaround time = 155.583214
Iteration Number = 199.000000 Burst time = 0.950622 Waiting time = 155.583214 Turnaround time = 156.533836
Iteration Number=200.000000 Burst time = 0.951883 Waiting time = 156.533836 Turnaround time = 157.485719
Throughput = 200 / 157.485719
= 1.269956
Results:
The total waiting time is calculated to be 156.533836 Deci seconds.
The total Turnaround time is 157.485719 Deci seconds.
The throughput is evaluated to 1.269956.
The best solution acquired by GWO is: 3.9997 4.0042 3.9899 3.9951
The best optimum value of the objective function found by GWO is: -10.3904
Fig. 2 shows the graphical representation of the obtained results from the implementation of GWO as per Case 1. The left figure shows the results in relevance with the parameter space and the right figure shows the graph depicting Best Score (Y-axis) against the number of iterations (X-Axis).

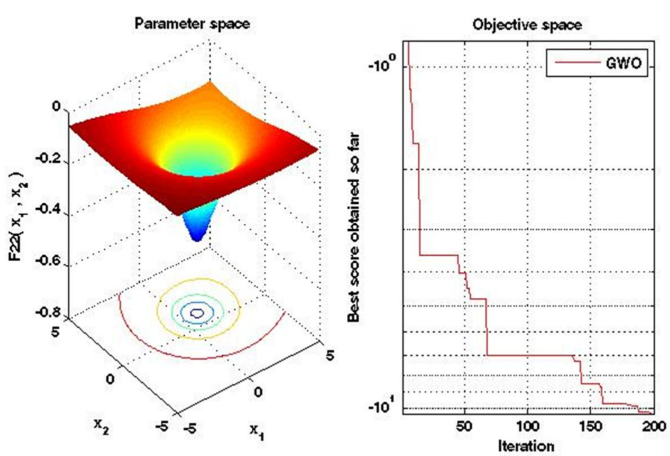

Fig. 2. Figure shows the obtained results obtained after execution of GWO in graphical form

V. MODIFIED HYBRID GWO-PSO

This section elaborates on the working of the proposed hybrid GWOPSO algorithm via a detailed flowchart, algorithm, and implementation. The conducted work performs a comparison of GWO and hybrid GWOPSO.

A. Flowchart

The proposed flowchart for hybrid GWOPSO is shown in Fig. 3.

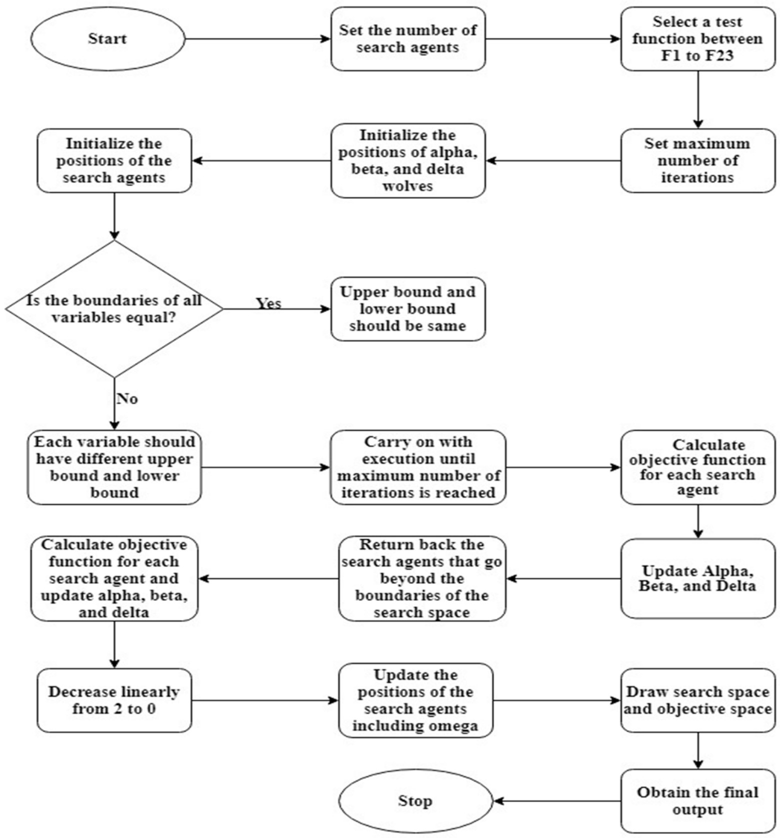

Fig. 3. Flowchart of proposed hybrid GWOPSO algorithm

B. Algorithm

- Set the number of search agents (*SearchAgents_no*)
- Opt for appropriate test function from F1 to F23(*Function_name*)
- Set maximum number of iterations (*Max_iter*)
- Load details of the selected benchmark function
- Initialize positions of alpha, beta, and delta wolves
  Alpha_pos=zeros(1,dim); Alpha_score=inf;Beta_pos=zeros(1,dim);
  Beta_score=inf;Delta_pos=zeros(1,dim);Delta_score=inf;
- Initialize the positions of search agents
  If the boundaries of all variables are equal and user enter a single number for both ub (upper bound) and lb (lower bound)
    if Boundary_no==1
      Positions=rand(SearchAgents_no,dim).(ub-lb)+lb;*
    End
    Else If each variable has a different lb and ub
     if Boundary_no>1
      for i=1:dim

ub_i=ub(i); lb_i=lb(i); Positions(:,i)=rand(SearchAgents_no,1).(ub_i-lb_i)+lb_i;*
End
End
End

- Execute until maximum number of iterations is reached.
  Calculate the objective function for each search agent.
  Update Alpha, Beta, and Delta.
  For Alpha
  if fitness<Alpha_score (value of fitness is less than Alpha_score)
  Alpha_score=fitness; (assign fitness to Alpha_score)
  Alpha_pos=Positions(i,:); (update positions)
  end
  For Beta
  if fitness>Alpha_score && fitness<Beta_score (value of fitness is more than Alpha_score and less than
  Beta_score)
  Beta_score=fitness; (assign fitness to Beta_score)
  Beta_pos=Positions(i,:); (update positions)
  end
  For Delta
  if fitness>Alpha_score && fitness>Beta_score && fitness<Delta_score (value of fitness is more than
  Alpha_score and Beta_score and less than Delta_score)
  Delta_score=fitness; (assign fitness to Delta_score)
  Delta_pos=Positions(i,:); (update positions)
  End
- Return the search agents that go beyond the boundaries of the search.
- Calculate objective function for each search agent and update Alpha, Beta, and Delta
- Decrease linearly from 2 to 0.
  a=2-l((2)/Max_iter)*
- Update the Position of search agents including omegas
- Draw search space, objective space, and obtain results.
  The best solution obtained by GWO, the best optimum value of the objective function found by GWO,
  the best solution acquired by PSOGWO, the best optimal value of the objective function found by PSOGWO

C. Implementation – Proposed Hybrid GWOPSO algorithm

The different parameters considered with assigned values to execute the working of Modified Hybrid GWO-PSO are given below.
Case 1:
Number of search agents (SearchAgents_no) = 30
Test function ranges from F1 to F23 (Function_name) = 'F22'
Maximum Number of Iterations (Max_iteration) = 200
The readings obtained after the execution of Modified Hybrid GWO-PSO as per Case 1 are mentioned below.
Iteration Number=1.000000 Burst time consumed in up to this iteration in seconds=0.004152 Turnaround time=0.004152
Final Total Burst Time=0.004152 Final Total Turnaround time=0.004152 Final Total Waiting time in seconds=0.000000
Iteration Number=2.000000 Burst time consumed in up to this iteration in seconds=0.008265 Turnaround time=0.012417
Final Total Burst Time=0.008265 Final Total Turnaround time=0.012417 Final Total Waiting time in seconds=0.004152
Iteration Number=3.000000 Burst time consumed in up to this iteration in seconds=0.009803 Turnaround time=0.022220

Final Total Burst Time=0.009803 Final Total Turnaround time=0.022220 Final Total Waiting time in seconds=0.012417

…

Iteration Number=198.000000 Burst time consumed in up to this iteration in seconds=0.313362 Turnaround time=31.731103

Final Total Burst Time=0.313362 Final Total Turnaround time=31.731103 Final Total Waiting time in seconds=31.417741

Iteration Number=199.000000 Burst time consumed in up to this iteration in seconds=0.315511 Turnaround time=32.046613

Final Total Burst Time=0.315511 Final Total Turnaround time=32.046613 Final Total Waiting time in seconds=31.731103

Iteration Number=200.000000 Burst time consumed in up to this iteration in seconds=0.317687 Turnaround time=32.364300

Final Total Burst Time=0.317687 Final Total Turnaround time=32.364300 Final Total Waiting time in seconds=32.046613

Throughput = 200 / 32.364300

= 6.17964856

Results:

The total waiting time is calculated to be 32.046613 Deci seconds.

The total Turnaround time is 32.364300 Deci seconds.

The throughput is evaluated to 6.17964856.

The best solution achieved by PSOGWO is: 8.0675 8.0556 8.0136 8.0208

The best optimal value of the objective function found by PSOGWO is: -4.9267

Fig. 4 shows the graphical representation of the obtained results from the implementation of Modified Hybrid GWO-PSO as per Case 11. The left figure shows the results in relevance with the parameter space and the right figure shows the graph depicting Best Score (Y-axis) against the number of iterations (X-Axis) in the case of GWO (blue line) and Hybrid PSOGWO (red line). As the graph depicts the Modified Hybrid GWO-PSO algorithm scores over the GWO algorithm in terms of Best Cost achieved.

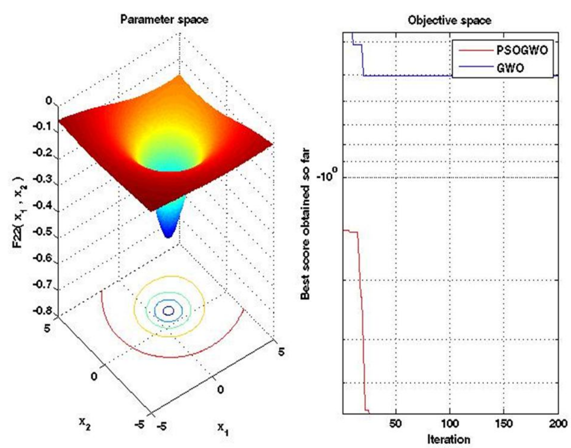

Fig. 4. Figure shows the graphical representation of the obtained results of Modified Hybrid GWO-PSO as per Case 1

The comparative evaluation of considered evaluation parameters is shown in Table I.

TABLE I. TABLE ILLUSTRATES THE VALUES OF PERFORMANCE EVALUATION PARAMETERS OF GWO AND HYBRID GWOPSO ALGORITHMS

Parameters / Algorithms	GWO	Hybrid GWOPSO
Waiting Time (deci seconds)	156.533836	32.046613
Turnaround Time (deci seconds)	157.485719	32.3643
Throughput	1.269956	6.17964856
Optimal Value	-10.3904	-4.9267

Fig. 5 shows the comparative graphical representation of performance evaluation parameters as per readings of GWO and Hybrid GWOPSO algorithms according to Table I.

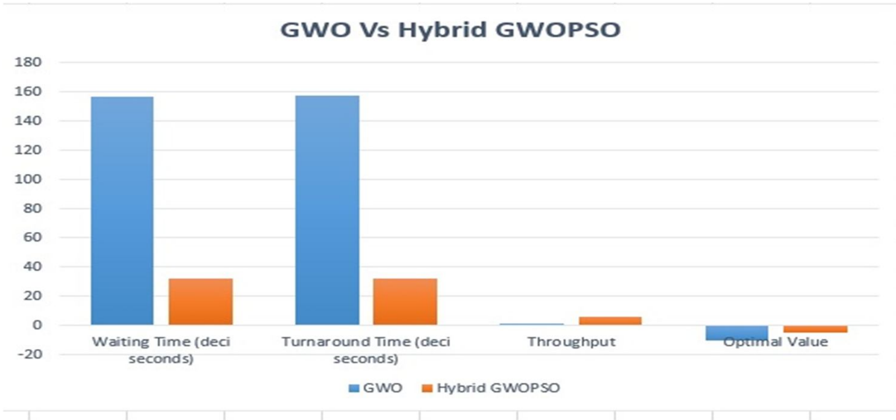

Fig. 5. Comparative graphical depiction of performance evaluation parameters as per readings of GWO and Hybrid GWOPSO algorithms rendering Table I

VI. Conclusion

The research work conducted in the research paper illustrated the hybrid meta-heuristic algorithm GWOPSO for optimization of resources available in cloud computing. The proposed algorithm is the hybrid version of PSO and GWO algorithms. The objectives of the proposed algorithm are to optimize the total execution cost and total execution time. The obtained results indicate a significant reduction in the execution time (waiting time and turnaround time) and increased throughput and optimal best solution. In the future, some other parameters like total energy consumption and load balancing can be considered for the valuation purpose. More algorithms can be considered for constructing the new hybrid algorithm and evaluation can be performed with the same parameters.

References

[1]. N. Gobalakrishnan, C. Arun, "Opposition learning-based grey wolf optimizer algorithm for parallel machine scheduling in cloud environment," Inter. J. Intell. Engine. Syst., Vol. 10 Issue 1, 2017, pp. 186 – 195.

[2]. Mirjalili, S.M. Mirjalili, A. Lewis, "Grey wolf optimizer," Adv. Eng. Soft., Vol. 69, 2014, pp. 46 – 61.

[3]. Cheng S, Qin Q, Chen J, Shi Y, "Brain storm optimization algorithm: a review," Artif Intell Rev, Vol. 46 Issue 4, 2016, pp. 445 – 458.

[4]. Kamboj VK, "A novel hybrid PSO–GWO approach for unit commitment problem," Neural Comput Appl, Vol. 27 Issue 6, 2016, pp. 1643 – 1655.

[5]. Chatterjee U., "A study on efficient load balancing algorithms in cloud computing environment," *International Journal of Current Engineering and Technology*, Vol. 3, 2013, pp. 1767 - 1770.

[6]. Gaidhane PJ, Nigam MJ., "A hybrid grey wolf optimizer and artificial bee colony algorithm for enhancing the performance of complex Systems," *J Comput Sci*., Vol 27, 2018, pp. 284 - 302.

[7]. Nema L, Sharma A, Jain S., "Load balancing algorithms in cloud computing: an extensive survey," *Int J Eng Sci Comput*., Vol. 6 Issue 6, 2016, pp. 7463 - 7468.

[8]. H. Zang, S. Zhang, and K. Hapeshi, "A review of nature inspired algorithms," *Journal of Bionic Engineering*, vol. 7, Issue. 4, 2010, pp. S232–S237.

[9]. S. Mirjalili, S. Saremi, S. M. Mirjalili, and L. D. S. Coelho, "Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization," *Expert Systems with Applications*, vol. 47, 2016, pp. 106–119.

[10]. S. Saremi, S. Z. Mirjalili, and S. M. Mirjalili, "Evolutionary population dynamics and grey wolf optimizer," *Neural Computing and Applications*, vol. 26, Issue 5, 2015, pp. 1257–1263.

[11]. M. Jamil and X. S. Yang, "A literature survey of benchmark functions for global optimisation problems," *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 4, Issue 2, 2013, pp. 150–194.

[12]. Trivedi IN, Jangir P, Kumar A, Jangir N, Totlani R, "A novel hybrid PSO–WOA algorithm for global numerical functions optimization," Adv Intell Syst Comput, Vol. 554, 2018, pp. 53 – 60.

[13]. Gobalakrishnan Natesan, Arun Chokkalingam, "Task Scheduling in heterogeneous cloud environment using mean grey wolf optimization algorithm", ScienceDirect, ICT Express 5, 2019, pp. 110 – 114.

[14].N. M. Hatta, Azlan Mohd Zain, Roselina Sallehuddin, Z. Shayfull, and Yusliza Yusoff, “Recent studies on optimisation method of Grey Wolf Optimiser (GWO): a review (2014–2017)”, Springer Nature, Artif Intell Rev, 52, 2019, pp. 2651 – 2683.
[15].Zheng-Ming Gao and Juan Zhao, “An Improved Grey Wolf Optimization Algorithm with Variable Weights”, Hindawi Computational Intelligence and Neuroscience, Volume 2019, Article ID 2981282, pp. 1 – 14.
[16].Zhihang Yue, Sen Zhang, and Wendong Xiao, “A Novel Hybrid Algorithm Based on Grey Wolf Optimizer and Fireworks Algorithm”, sensors, MDPI, 20, 2147, 2020, pp. 1 -17.
[17].Li Xingjun, Shao Zhiwei, Cheng Hongping, Bayan Omar Mohammed, “A new fuzzy-based method for load balancing in the cloud-based Internet of things using a grey wolf optimization algorithm”, Int J Commun Syst. 2020, e4370, Wiley, pp. 1 – 19.
[18].Panda M, Das B, “Grey wolf optimizer and its applications: a survey,” Lecture notes in electrical engineering. Springer, Singapore, 2019, pp. 179 – 194.
[19].Kohli M, Arora S, “Chaotic grey wolf optimization algorithm for constrained optimization problems,” J Comput Des Eng5(4), 2018, pp. 458 – 472.
[20].Panwar LK, Reddy S, Verma A, Panigrahi BK, Kumar R (2018)Binary Grey Wolf Optimizer for large scale unit commitment problem. Swarm Evol Comput, Issue 38, 2018, pp. 251 – 266.
[21].Barraza J, Rodrı´guez L, Castillo O, Melin P, Valdez F, A” new hybridization approach between the fireworks algorithm and grey wolf optimizer algorithm,” J Optim, Issue 18, 2018, pp. 1 – 18.