International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Opposition Learning-Based Grey Wolf Optimizer Algorithm for Parallel Machine Scheduling in Cloud Environment

Gobalakrishnan Natesan[1*] **Arun Chokkalingam**[2]

[1]*Sathyabama University, Chennai, Tamil nadu, India*
[2]*R.M.K College of Engineering and Technology, Chennai, Tamil nadu, India*
* Corresponding author's Email: gobalakrishnan0508@gmail.com

Abstract: Cloud computing is a novel developing computing paradigm where implementations, information, and IT services are given over the internet. The parallel-machine scheduling (Task-Resource) is the important role in cloud computing environment. But parallel-machine scheduling issues are premier that associated with the efficacy of the whole cloud computing facilities. A good scheduling algorithm has to decrease the implementation time and cost along with QoS necessities of the consumers. To overcome the issues present in the parallel-machine scheduling, we have proposed an oppositional learning based grey wolf optimizer (OGWO) on the basis of the proposed cost and time model on cloud computing environment. Additionally, the concept of opposition based learning is used with the standard GWO to enhance its computational speed and convergence profile of the proposed method. The experimental results show that the proposed method outperforms among all methods and provides quality schedules with less memory utilization and computation time.

Keywords: Parallel machine scheduling, Task, Resource, Multi-objective, Oppositional learning based grey wolf optimizer, Time, Cost.

1. Introduction

Cloud computing is the Internet-linked mode of supercomputing. As the skills are mounting day by day, the prerequisite of computing and storage resources are quickly increasing. So capitalizing more and more equipment is not a cost-effective technique for an organization to please the even growing computational and storage need. Thus Cloud Computing has developed an extensively recognized paradigm for great performance computing [1, 2]. It simplifies mainly to decrease capital cost, decouple facilities from the fundamental technology and gives flexibility in the name of resource provisioning [3]. The chief benefit of cloud computing is the skill to provision IT resources on request [4, 5]. But these resources are used by the consumer without having enough information about the methodological details [6, 7]. Cloud computing gives some services that are presented under numerous deployment models: platform as a service (PaaS), infrastructure as a service (IaaS) [8], software as a service (SaaS), and network as a service (NaaS) [9, 10]. Scheduling is utilized here to control the order of work to be achieved using a computer scheme [11] to exploit the resource operation and diminish processing time of the tasks [12]. The area of scheduling algorithm investigation is to attain an optimal value that can be the uppermost performance or the shortest implementation time, over a sequence of intentions [13].

In recent years, scheduling approach plays a significant role in modern applications and especially, task scheduling has been received a great transaction of attention among the studies due to its wide applicability and abundant growth of cloud computing based system [14]. A good scheduler familiarizes its scheduling approach according to the altering environment and the type of task. Rendering to this, F. A. Omara and M. M. Arafa [15] have elucidated the task scheduling issue by genetic algorithm. At this time, two genetic algorithms were utilized to resolve these scheduling issues. To overcome this issue, the author S. Abraham and M. Naghibzadeh [16] have elucidated the Deadline-constrained workflow preparation in software as a service Cloud. In moreover, to decrease the cost of the dispensation the author L. Goo *et al.* [17] have elucidated the Task Scheduling by the optimization algorithm (PSO) that is on the basis of minor position value rule. To date, the workflow issue further familiarized the workflow scheduling for cloud atmosphere on the basis of Artificial Bee Colony algorithm by P. Kumar and S. Anand [18].

Similarly, to overcome the deadline-driven resource allocation issue S. Di and C. L. Wang [19] have clarified the Error-Tolerant Resource Allocation and Payment Minimization for Cloud Scheme. J. T. Tsai *et al.* [20] have elucidated the optimize task scheduling and resource allocation by an enhanced differential evolution algorithm (IDEA) on the basis of the cost and time models on cloud computing atmosphere. Additionally, A. Agarwal and S. Jain [21] have enlightened an Efficient Optimal Algorithm for Task Scheduling in Cloud Computing Environment on the basis of priority. To overcome the issue, the author X. Zuo *et al.* [22] industrialized a Self-Adaptive Learning PSO-Based Deadline Constrained Task Scheduling for Hybrid infrastructure as a service (IaaS) Cloud. The important problem of scheduling was how to assign users' tasks to exploit the profit of IaaS provider though guaranteeing QoS. This issue was expressed as an integer programming (IP) model, and resolved with the help of a self-adaptive learning particle swarm optimization (SLPSO)-depended scheduling method in [22]. But, their method cannot appropriate for high issue instance types because of the lacking presentation of computational time.

The main aim of this paper is to optimize parallel- machine scheduling (task and resource) using an oppositional grey wolf optimization algorithm (OGWO) based on the proposed multi-objective models in cloud. The proposed parallel machine scheduling that hybridizes the grey wolf optimization (GWO) with oppositional-based learning (OBL), where OBL is improving the performance of the GWO algorithm while optimizing the task and resources. The organization of the paper is as follows: Section II presents the background of the research and Section III presents proposed parallel machine scheduling using OGWO algorithm. Section IV present the Result and discussion part. The conclusion part is given in section V.

2. Problem Formulation

Table 1. Parameters used in the parallel machine scheduling

symbol	definition
T_i	*Task i* , $1 \le i \le k$
S_i	*Subtask i,* $1 \le i \le m$
R_i	*Resource i,* $1 \le i \le N$
T^{pro}	Processing time of subtask
T^{Rec}	Receiving time of subtask
T_{wait}	Waiting time
$C^{P}_{\mathrm{Re}nt}$	Rent cost of processing subtask
$C^{R}_{\mathrm{Re}nt}$	Rent cost of receiving subtask
C_{Total}	Total cost

In parallel machine scheduling, we have obtained two types of problems such as routing problem and sequencing problem. To assign each task to the corresponding resources, we can obtain routing problem and to series the subtask on the resources (sequencing problem) to decrease the entire cost and makespan. Let as considering the user task T_i and each task has numerous subtask S_i and each subtask is permissible to be administered on any specified accessible resources R_i. Primarily, it is presumed that there are k tasks $T_i = (T_1, T_2, ..., T_k)$, m subtask $S_i = (S_1, S_2, ..., S_m)$ and n resources $R_i = (R_1, R_2, ..., R_N)$ in the current scheme of cloud computing. A cloud resource has an assumed level of capacity (e.g., CPU, memory, network, storage). A subtask is administered on one resource at a time and the given resources are available continuously. Task scheduling of cloud computing can be quantified as follows.

3. Proposed Methodology of Parallel Machine Scheduling

The main intention of this paper is to optimize task and resource (called parallel-machine scheduling) using oppositional learning based grey wolf optimizer (OGWO) based on the proposed cost and time models on cloud computing environment. To optimize the parallel machine, we utilize multi-objective function based on cost and time model of proposed approach. Two types of cost are included in the proposed model such as processing and receiving a cost. Similarly, the time model includes receiving, processing and waiting time. The good parallel machine scheduling decreases the total running time and cost function. The overall diagram of the proposed method is illustrated in figure 1.

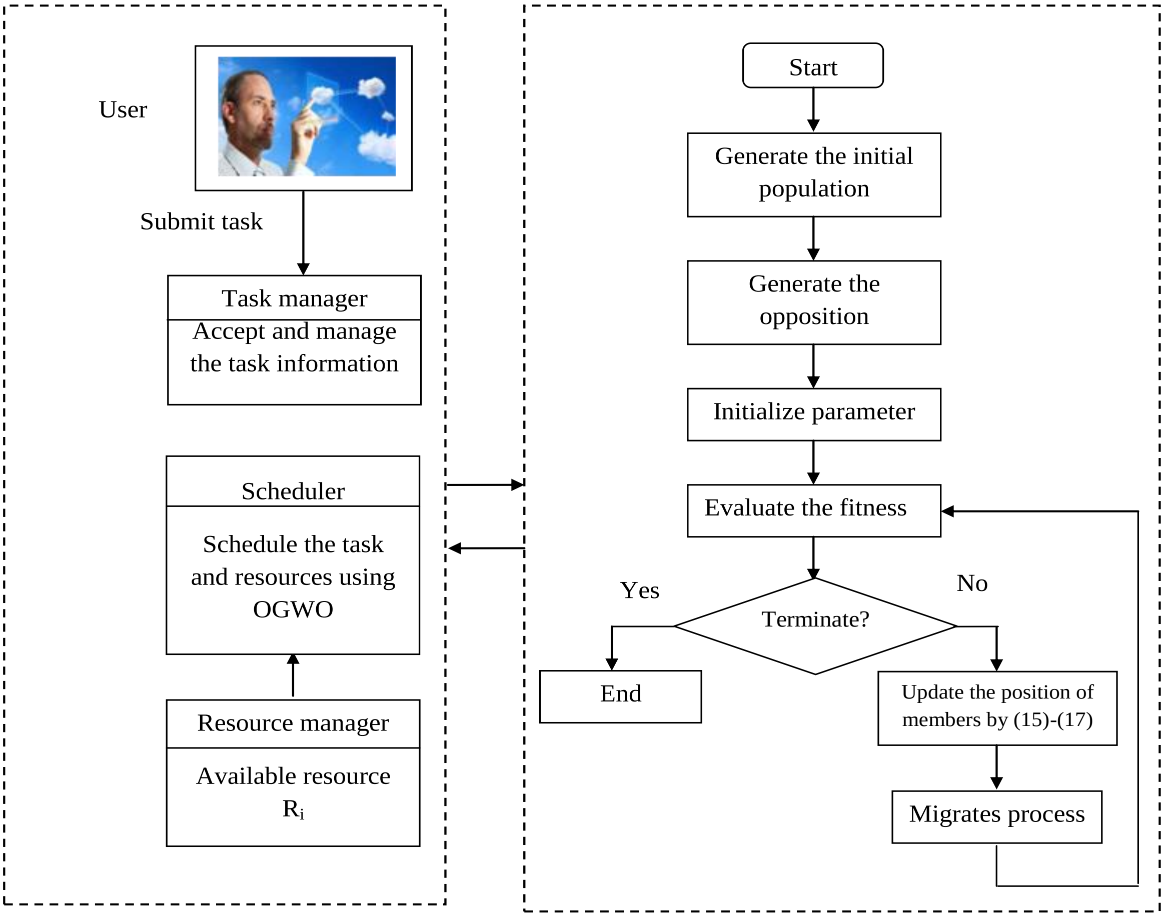

Figure.1 Overall diagram of the proposed parallel –machine scheduling

3.1. Scheduling Optimization Model based on Multi-Objective Function

In this paper, we proposed a parallel machine scheduling based on multi-objective function using Opposition learning-based Grey wolf optimizer. The Grey Wolf Optimizer (GWO) encouraged by grey wolves (Canis lupus). The GWO algorithm mimics the leadership hierarchy and hunting mechanism of grey wolves in nature. To progress the performance of the scheme, in our paper we utilize Opposition learning-based Grey wolf optimizer (OGWO). By uniting opposition-based learning with GWO, it overawed the separate drawbacks of GWO algorithms and it effortlessly understands and rapidly converges, so this scheduling method is able to obtain an optimal or suboptimal result in a minimum computational cost and time. Through our assumptions, the solution segment illustrates that the projected optimization of OGWO attained better performance than the separate performance. The step by step process of proposed parallel machine scheduling is explained below;

Step 1: Solution encoding

In optimization algorithm, the solution encoding is the important process. In this work, the solution consists of two components such as task and resources. The task consists of *k* a number of the subtask. At first, we randomly assign each subtask to any one resource. For example, we consider four tasks each of that has four subtasks. With the help of this subtask, we generate a sixteen-dimensional vector that is V_F=(1,2,3,4,2,4,3,1,2,3,4,1,3,4,1,2). The primary element “1” of V_F is the first subtask of task 1. The secondary element “2” of V_F is the first of task 2. The Tertiary element “3” of V_F is the first subtask of task 3. The fourth element of “4” of V_F is the first subtask of task 4, and so on. In this, each subtask is assigned in any one of the resources. For the encoding procedure, each solution includes a series of subtasks and resources. For instance, we yield five resources for scheduling. The main objective of this paper is to schedule these 16 subtasks to corresponding five tasks. At first, we randomly assign each resource which are displayed in equation (1).

$$Y_{ij} = \begin{cases} (1,R1),(2,R4),(3,R2),(4,R5)(3,R3),(2,R2),(4,R1), \\ (3,R4),(1,R5),(2,R1),(3,R2),(4,R3),(3,R4),(2,R3) \\ (1,R3),(2,R2),(4,R1),(3,R4),(3,R1),(1,R3), \\ (4,R2),(1,R5),(2,R1),(1,R3),(4,R4),(3,R2), \\ (4,R2)(3,R3),(1,R5),(2,R4) \\ \vdots \\ (3,R2),(2,R3),(1,R4),(4,R1),(1,R2),(3,R4),(4,R3), \\ (2,R5),(1,R1),(2,R3),(4,R2),(3,R5),(4,R4),(3,R2) \end{cases} \quad (1)$$

Where,
R1,…R5 → resources
1,…,4 → Subtasks

Step 2: Generate opposite solution

As per opposition based learning (OBL) presented by Tizhoosh in 2005 [23], the present wolves and its inverse wolves are considered all the while to show signs of improvement guess for current wolves solution. It is given that an inverse wolf's solution has a superior opportunity to be nearer to the global optimal solution than arbitrary wolf's solution. Every solution Y_i has a unique opposite Y_{opi} solution. The opposite solution $OP(Y_1', Y_2', ... Y_n')$ is calculated based on the equation;

$$Y'_{ij} = a_i + b_i - Y_i \quad , i\epsilon 1,2,...,n \quad (2)$$

Step 3: Fitness calculation

Once the initial solution is generated, the fitness value of each individual is evaluated and stored for future reference. The fitness function is defined as the following expression;

$$FF_i = \min\,(C_{Total}\ , Makespan) \quad (3)$$

Here, we used a multi-objective function which is including cost and time model. The proposed cost model consists of two types of cost such as processing C^{Pro} and receiving C^{Rec} subtask. Subsequently, the time model T^{Pro} and T^{Rec} be processing and receiving time, respectively, of a subtask. The total cost C_{Total} is calculated based on the equation (4).

$$C_{Total} = \sum^{all\,subtask} \left(C^{\mathrm{Pro}} + C^{\mathrm{Rec}}\right) \quad (4)$$

$$C^{\mathrm{Pro}} = T^{\mathrm{Pro}} \times C^{P} \quad (5)$$

$$C^{\mathrm{Rec}} = T^{\mathrm{Rec}} \times C^{R} \quad (6)$$

Where, C^{Pro} is Processing cost, C^{Rec} is Receiving cost, C^{P} is Processing cost of per unit time, C^{R} is Receiving cost of per unit time.

Moreover, the total time taken to complete the task T^{i}_{Total} is given in equation (7). The total time includes processing time T_{Pro}, receiving time T_{Rec} and waiting time T_{Wait}.

$$T^{i}_{Total} = \sum T_{\mathrm{Rec}} + \sum T_{\mathrm{Pro}} + \sum T_{Wait} \quad (7)$$

$$Makespan = Min\left(T_{Total\_1}, T_{Total\_2}, ..., T_{Total\_m}\right) \quad (8)$$

Where; m is the number of given available resources.

Step 4: Calculating *α, β, δ* and *ω*

After the fitness calculation, we find out α, β, δ and ω. Here, the alpha (α) is esteemed as the most suitable arrangement with a perspective to replicating logically the social pecking order of wolves while conceiving the OGWO. Thus, the second and the third best arrangements are named as beta (β) and delta (δ) separately. The remaining applicant arrangements are regarded to be the omega (ω). Let the first best fitness solutions be F_α, the second best fitness solutions F_β and the third best fitness solutions F_δ.

Step 5: Encircling prey

The hunting is guided by α, β, δ and ω follow these three candidates. In order for the pack to hunt a prey is first encircling it.

$$F(t+1) = F(t) + \vec{A}.\vec{K} \quad (9)$$

$$\vec{K} = |\vec{C}.F(t+1) - F(t)| \quad (10)$$

$$\vec{A} = 2\vec{a}r_1 - \vec{a} \text{ And } \vec{C} = 2r_2 \quad (11)$$

Step 6: Hunting

We undertake that the alpha (best candidate solution), beta and delta have the enhanced information about the potential location of the prey to replicate mathematically the hunting behavior of the grey wolves. For recurrence, the novel solution $d(t+1)$ is assessed by using the formulae cited underneath.

$$\vec{K}^{\alpha} = |\vec{C}_1.F_\alpha - F|,\ \vec{K}^{\beta} = |\vec{C}_2.F_\beta - F|, \\ \vec{K}^{\delta} = |\vec{C}_3.F_\delta - F| \quad (12)$$

$$F_1 = F_\alpha - \vec{A}_1.(\vec{K}^{\alpha}),\ F_2 = \\ F_\beta - \vec{A}_2.(\vec{K}^{\beta}),\ F_3 = F_\delta - \vec{A}_3.(\vec{K}^{\delta}) \quad (13)$$

$$F(t+1) = \frac{F_1 + F_2 + F_3}{3} \quad (14)$$

It can be recognized that the concluding location would be in a random place within a circle that is distinct using the positions of alpha, beta, and delta in the search space. In another aspects alpha, beta, and delta assess the location of the prey, and additional wolves updates their positions arbitrarily around the prey.

Step 7: Attacking prey (exploitation) and Search for prey (exploration)

Exploration and exploitation are definite using the adaptive values of a and A. The adaptive values of parameters a and A permit OGWO to smoothly transition amongst exploration and exploitation. With declining A, half of the iterations are dedicated to exploration ($|A|\geq 1$) and the other half are devoted to exploitation ($|A|<1$). The GWO has only two chief parameters to be attuned (a and C). Though, we have retained the OGWO algorithm as simple as conceivable with the smallest operators to be attuned. The procedure will be sustained until the maximum number of iteration is attained. Lastly, the optimal results are selected on the basis of the fitness value.

Step 8: Termination Criteria

The algorithm discontinues its execution only if a maximum number of iterations is achieved and the solution which is holding the best fitness value is selected and it is specified as the best solution to parallel machine scheduling. Once the best fitness is attained by means of OGWO algorithm, selected task is allocated for cloud computing process. The pseudo code of proposed parallel machine scheduling is illustrated in table 2.

Table 2. Pseudo code of proposed parallel machine scheduling

Input:
The parameter of OGWO algorithm
The parameter of Parallel machine scheduling
output:
A scheduled task
Assumption:
Input solution Y_i, Fitness FF_i, opposite solution $Y_i{}'$, the data size of each task D^S, Rent cost, processing capacity.

Initialization:
Initialize the number of tasks T_i, number of the subtask S_i, number of resources R_i, Coefficient vector A, C.

Start:
Generate the initial population Y_{ij}, $i= 1, 2,....n$ and $j=1, 2,....D$
Calculate the opposite population of wolves $Y_{ij}{}'$ using equation (2)
for all Y_i, $OY_i{}'$ do
Evaluate the fitness (FF_i) of the population using (3)
end for
Set cycle to 1
Repeat
Select the best search agent F_α
Select the second best agent F_β
Select the third best agent F_γ
While (*t<max number of iterations*)
for each search agent
Update the position of the current search agent u sin g equation (14)
end for
Update α, A, C
Calculate fitness of all search agents
Update F_α, F_β and F_γ
$t=t+1$
end while
return F_α
end
scheduled task
stop

 DOI: 10.22266/ijies2017.0228.20

4. Result and Discussion

In this section, we discuss the result obtained from the proposed OGWO algorithm based task scheduling technique. We have implemented our proposed task scheduling using Java (jdk 1.6) with Cloudsim tools and a series of experiments were performed on a PC with Windows 7 Operating system at 2 GHz dual core PC machine with 4 GB main memory running a 64-bit version of Windows 2007. The utilization rate of CPU or Memory at the S_{th} time slot for the cloud is calculated by

$$M(s)=\sum_{i=1}^{D} \mathrm{Re}\, s_i\, E_{is} / Total-\mathrm{Re}\, s \quad ; s\epsilon\{1,2..,S\} \qquad (15)$$

Where,

Res_i → Number of CPU or the size of memory requested by task T_i

Total-Res→ Total CPU or memory in the private cloud

4.1. Experimental Results

The basic idea of our proposed methodology is parallel-machine scheduling using Oppositional grey wolf optimization algorithm. Here, at first, we assign the N number of task and M number of resources. To schedule the task based on the cost and time function. In this work, we utilized two examples such as (i) A five task and five resources and (ii) a fifteen task and eight resources.

❖ **Example 1: A five task and five resources**

In this we schedule five task and five resources in cloud. Similarly, each task consists of five subtasks. Totally, the scheduling approach has twenty-five task and five resources. The Task T= (T_1, T_2, T_3, T_4, S_{45}). Where T_1= (S_{11}, S_{12}, S_{13}, S_{14}, S_{15}), T_2= (S_{21}, S_{22}, S_{23}, S_{24}, S_{25}), T_3= (S_{31}, S_{32}, S_{33}, S_{34}, S_{35}), T_4= (S_{41}, S_{42}, S_{43}, S_{44}, S_{45}) and T_5= (S_{51}, S_{52}, S_{53}, S_{54}, S_{55}). The five resources are R= (R_1, R_2, R_3, R_4, R_5). The aim of proposed work is reducing the objective function using equation (10) and (14). In this, we design the Data size (DS) value of each subtask which is given in table 3. The cost value and processing capacity of each resource are given in table 4. Using DS value and processing capacity we calculate the processing time which is given in table 5.

Table 3. Data size (DS) of an each task

T_i	S_{i1}	S_{i2}	S_{i3}	S_{i4}	S_{i5}
T_1	1.2	0.5	1.0	0.6	0.9
T_2	0.7	1.2	1.6	0.8	-
T_3	1.4	0.9	1.0	-	-
T_4	1.6	1.0	-	-	-
T_5	1.4	1.8	1.0	0.5	0.9

Table 4. Rent cost and processing capacity on available resources

Resources	Rent cost (USD/per hour)	Processing capacity (GB/per hour)
R_1	0.16	0.2
R_2	0.22	0.3
R_3	0.5	0.45
R_4	0.6	0.8
R_5	0.93	1.2

Table 5. Subtask of processing time

Resources	T_{11}	T_{12}	T_{13}	T_{14}	T_{15}
R_1	6 (=1.2/0.2)	2.5 (=0.5/0.2)	5 (=1.0/0.2)	3 (=0.6/0.2)	4.5 (0.9/0.2)
R_2	4 (=1.2/0.3)	1.66 (=0.5/0.3)	3.33 (=1.0/0.3)	2 (=0.6/0.3)	3.0 (=0.9/0.3)
R_3	2.66 (=1.2/0.45)	1.11 (=0.5/0.45)	2.222 (=1.0/0.45)	1.33 (=0.6/0.45)	2 (=0.9/0.45)
R_4	1.5 (1.2/0.8)	0.625 (=0.5/0.8)	1.25 (=0.1/0.8)	0.75 (=0.6/0.8)	1.125 (=0.9/0.8)
R_5	1 (=1.2/1.2)	0.4166 (=0.5/1.2)	0.438 (=0.1/1.2)	0.0833 (=0.6/1.2)	0.5 (=0.6/1.2)

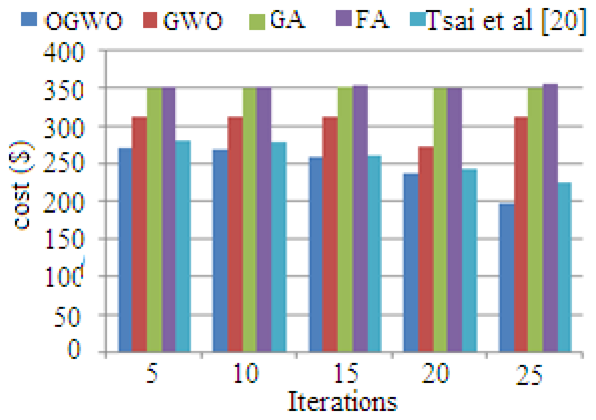

Figure.2 Performance analysis of proposed against existing using cost function

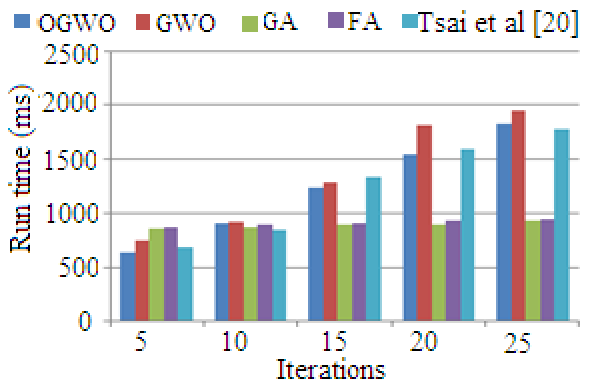

Figure.3 Performance analysis of proposed against existing using Runtime

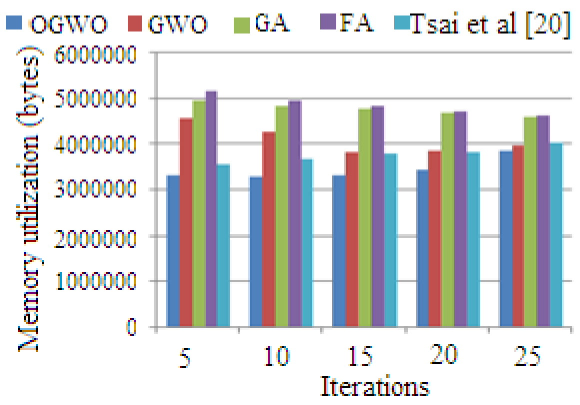

Figure.4 Performance plot for memory utilization

The above figure 2 to 4 shows the performance of proposed methodology based scheduling using five tasks and five resources. Here, we compare our proposed OGWO algorithm with GWO, FA GA algorithm and Tsai *et al*. [20]. The GA is a metaheuristic algorithm which is usually used to create high-quality solutions to optimization and search problems by relying on bio-inspired operators such as mutation, crossover and selection. Similarly, FA is mathematical optimization algorithms which are inspired by the flashing behaviour of fireflies. The GWO algorithm is a meta-heuristic methodology recreating wolves' behavior while they are hunting. The above mentioned three optimization algorithms are most accurate optimization algorithm but it has some limitations. To improve the performance of the fitness function, the oppositional method is developed. In the oppositional based approach, the dimension of one agent can become opposite with respect to the source. Moreover, here we compare one more approach to the proposed method which is Tsai *et al*. [20]. Here, the author develops the parallel machine scheduling based on Improved Differential Evolution Algorithm. The above figure 2 shows the performance of proposed approach based on a cost function. When analyzing figure 2, we obtain the minimum cost of 285.27$ which is 312.173$ for using GWO, 349$ for GA 351$ for FA and 300$ for Tsai *et al*. [20]. Moreover, figure 3 shows the performance of proposed approach based on runtime. From the figure 3, we understand our proposed approach takes a minimum time of 899ms. From the figure 4, we clearly understand our proposed approach takes minimum memory utilization rate of 3301696 bytes to schedule the five take and five resources. From the result part, we clearly understand our proposed approach achieves the better result compare to the other approaches.

- **Example 2: A fifteen task and eight resources**

Here, we used fifteen task and eight resources for parallel machine scheduling in the cloud.Here, each task consists of ten sub-task and the corresponding data size is shown in table 7. Totally, we have 150 tasks. Rent cost and processing capacity on available resources are shown in table 6 and the calculated processing time is shown in table 8.

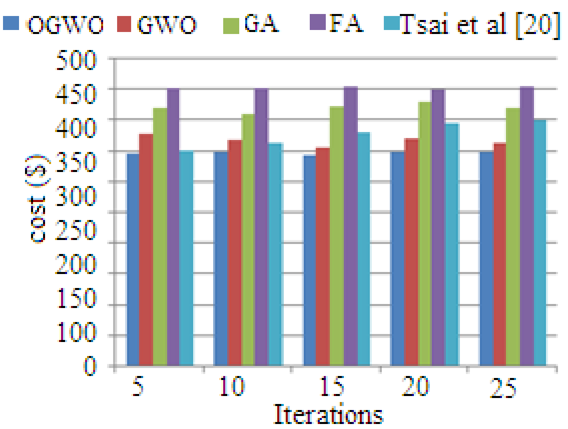

Figure.5 Performance analysis of proposed against existing using cost function

Table 6. Rent cost and processing capacity on available resources

resources	Rent cost (USD/per hour)	Processing capacity (GB/per hour)
R_1	0.69	1.5
R_2	0.70	0.7
R_3	0.08	1.0
R_4	0.63	0.19
R_5	0.85	1.18
R_6	0.55	0.85
R_7	0.41	0.34
R_8	0.31	0.63

Table 7. Data size of each task

T_i	S_{i1}	S_{i2}	S_{i3}	S_{i4}	S_{i5}	S_{i6}	S_{i7}	S_{i8}	S_{i9}	S_{i10}
T_1	1.32	2.3	2.86	0.34	0.83	2.65	0.33	1.28	2.82	0.78
T_2	2.18	2.77	1.57	2.29	0.32	1.57	1.39	1.28	2.82	0.78
T_3	0.3	2.8	1.61	2.14	1.79	0.42	0.81	2.74	1.88	2.19
T_4	2.76	1.68	0.43	1.72	1.62	0.83	2.64	1.32	2.3	2.86
T_5	2.01	0.81	0.64	2.81	1.09	0.74	0.29	2.18	2.77	1.57
T_6	2.89	0.27	2.87	1.67	0.31	1.07	2.12	0.3	2.8	1.61
T_7	1.09	1.78	2.76	1.45	2.9	0.62	0.52	2.76	1.68	0.43
T_8	2.287	0.28	1.347	2.47	1.39	0.23	1.96	2.01	0.81	0.64
T_9	2.47	2.32	1.93	0.12	1.42	2.14	1.41	2.89	0.27	2.87
T_{10}	0.89	1.78	1.89	0.47	1.41	2.84	1.74	1.09	1.78	2.76
T_{11}	2.77	2.29	1.28	2.82	0.78	0.6	0.36	2.287	0.28	1.347
T_{12}	2.77	2.29	1.28	2.82	0.78	0.6	0.36	2.47	2.32	1.93
T_{13}	1.52	2.45	2.74	1.88	2.19	0.39	2.27	0.89	1.78	1.89
T_{14}	1.6	2.19	1.32	2.3	2.86	0.34	0.83	2.78	1.48	0.43
T_{15}	1.45	2.12	2.18	2.77	1.57	2.29	0.32	1.57	1.39	2.64

Table 8. Subtask of processing time

Resources	T_{11}	T_{12}	T_{13}	T_{14}	T_{15}	T_{16}	T_{17}	T_{18}	T_{19}	T_{110}
R_1	0.88	1.45	1.81	0.21	0.525	1.6	0.20	0.81	1.78	0.49
R_2	1.88	3.02	3.75	0.44	1.09	3.4	0.433	1.68	3.70	1.02
R_3	1.32	2.1022	2.61	0.31	0.75	2.4	0.301	1.1	2.57	0.71
R_4	6.947	11.706	4.55	1.73	4.22	13.48	1.67	6.5	14.3	3.9
R_5	1.11	1.934	2.405	0.28	0.69	2.22	0.277	1.07	2.3	0.65
R_6	1.54	2.683	3.33	0.39	0.96	3.09	0.38	1.4	3.2	0.91
R_7	3.80	6.62	8.23	0.97	2.3	7.63	0.95	3.6	8.12	2.2
R_8	2.09	3.609	4.48	0.533	1.3	4.15	0.51	2.0	4.42	1.2

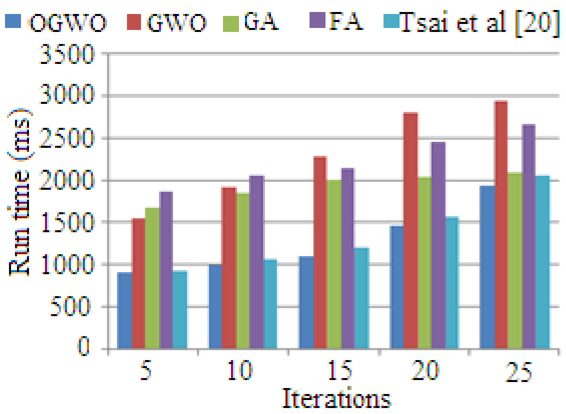

Figure.6 Performance analysis of proposed against existing using Runtime

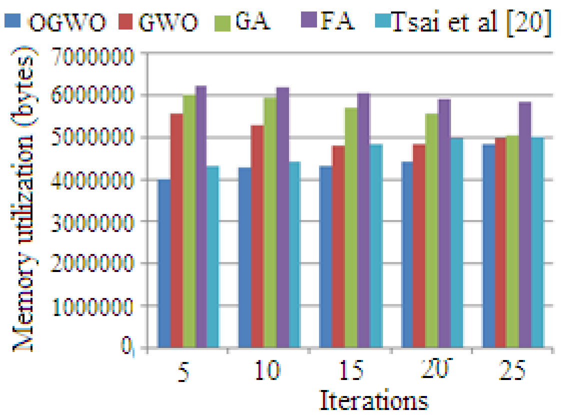

Figure.7 Performance plot for memory utilization

The above figure 5-7 shows the performance of proposed methodology based on parallel machine scheduling using fifteen task and eight resources. Figure 5 shows the performance analysis of proposed against existing approach using cost function. When analyzing figure 5, we obtain the minimum cost of 342.455$ which is 363.173$ for using GWO, 411$ for GA, 450$ for FA and 350$ for [20] based parallel machine scheduling. Figure 6 shows the performance analysis of proposed against existing using Runtime. The system achieves the minimum runtime of 1102 ms which is 1543ms for using GWO algorithm. Also, the system indicates the number of iteration is increases means the runtime of scheduling process also increases. Similarly, figure 7 shows the Performance plot for memory utilization. Here, also we obtain the better result compare to the other approaches. From the above figures, we clearly understand our proposed approach achieves the minimum time and cost model compare to other approaches.

5. Conclusion

In this paper, a multi-objective parallel machine scheduling method was proposed based on the oppositional grey wolf's optimization (OGWO). To achieve the multi-objective function, we proposed a novel method that hybridizes the GWO with Opposition-based learning (OBL), where OBL is improving the performance of the GWO algorithm while optimizing the task and resources. The multi-objective optimization approach is used to improve the scheduling performance compared to single objective function. The experimental results took based on two examples such as five task and five resources and fifteen task and eight resources. The result shows our proposed multi-objective-based parallel machine scheduling better than other approaches. In future, new algorithm will propose for parallel machine scheduling and comparative with existing algorithms. We are also trying to extend our work to support real time.

References

[1] P. Kumar and A. Verma, "Independent Task Scheduling in Cloud Computing by Improved Genetic Algorithm", *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol. 2, No. 5, 2012.

[2] N. Noman, H. Iba, "Accelerating differential evolution using an adaptive local search", *IEEE Trans. on Evolutionary Computation*, Vol. 12, No.1, pp. 107-125, 1999.

[3] M. Othman, S. A. Madani & S. U. Khan, "A Survey of Mobile Cloud Computing Application Models" *IEEE Communications Surveys and Tutorials*, Vol. 16, No. 1, pp. 393-413, 2014.

[4] A. S. Prasad and S. Rao, "A Mechanism Design Approach to Resource Procurement in Cloud Computing", *IEEE transactions on computers*, Vol. 63, No. 1, pp. 17-30, 2014.

[5] D. Breitgand, A. Maraschini and J. Tordsson, "Policy-Driven Service Placement Optimization in Federated Cloud", *IBM Research Report*, pp. 11-15, 2011.

[6] R. Dhivya and C. Senbagavalli, "Intelligent Strategy of Task Scheduling in Cloud computing using Swarm Optimization with Griewangk's function", *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 4, No. 12, 2015.

[7] R. Storn, K. Price, "Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces", *Journal*

of Global Optimization, Vol. 11, No.4, pp. 341-359, 1997.

[8] A. N. Toosi, R. N. Calheiros, P. K. Thulasiram and R. Buyya, "Resource provisioning policies to increase IaaS provider's profitinafederated cloud environment", *High Performance Computing and Communications*, pp. 279-287, 2011.

[9] S. Ortiz, "The problem with cloud-computing standardization", *Computer*, Vol. 44, No. 7, pp. 13-16, 2011.

[10] M. A. Sharkh, M. Jammal, A. Shami and A. Ouda, "Resource Allocation in a Network-Based Cloud Computing Environment: Design Challenges", *IEEE Communications Magazine*, Vol. 51, No. 11, pp. 46-52, 2013.

[11] A. Agarwal and S. Jain, "Efficient Optimal Algorithm of Task Scheduling in Cloud Computing Environment", *International Journal of Computer Trends and Technology*, Vol. 9, 2014.

[12] J. L. Hitesh and A. Bheda, "An Approach to Optimized Resource Scheduling using Task Grouping in Cloud", *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol. 3, No. 9, 2013.

[13] Z. Yong, C. Liang and L. Youfu, "Efficient Task Scheduling for Many Task Computing with Resource Attribute Selection", *China Communications*, Vol. 11, No. 12, pp. 125-140, 2015.

[14] J. Rameshkumar, S. Ganesan, M. Abirami and S. Subramanian, "Cost, emission and reserve pondered pre-dispatch of thermal power generating units coordinated with real coded grey wolf optimization", *IET Generation, Transmission & Distribution*, Vol. 10, No. 4, pp.972-985, 2016.

[15] F. A. Omara and M. M. Arafa, "Genetic algorithms for task scheduling problem", *Journal of Parallel and Distributed Computing*, Vol. 70, No. 1, pp. 13-22, 2010.

[16] S. Abrishami and M. Naghibzadeh, "Deadline-constrained workflow scheduling in software as a service Cloud", *Scientia Iranica*, Vol. 19, No. 3, pp. 680-689, 2012.

[17] L. Guo, S. Zhao, S. Shen and C. Jiang, "Task Scheduling Optimization in Cloud Computing Based on Heuristic Algorithm", *Journal Of Networks*, Vol. 7, No. 3, pp. 547-553, 2012.

[18] P. Kumar and S. Anand, "An Approach to Optimize Workflow Scheduling For Cloud Computing Environment", *Journal of Theoretical and Applied Information Technology*, Vol. 57, No. 3, pp. 617-623, 2013.

[19] S. Di and C. L. Wang, "Error-Tolerant Resource Allocation and Payment Minimization for Cloud System", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 24, No. 6, pp. 1097-1106, June 2013.

[20] J. T. Tsai, J. C. Fang and J. H. Chou, "Optimized Task Scheduling and Resource Allocation on Cloud Computing Environment Using Improved Differential Evolution Algorithm", *Computers & Operations Research*, Vol. 40, No. 12, pp. 3045-3055, 2013.

[21] A. Agarwal, S. Jain, "Efficient Optimal Algorithm of Task Scheduling in Cloud Computing Environment", *International Journal of Computer Trends and Technology (IJCTT)*, Vol. 9, No. 7, Mar 2014.

[22] X. Zuo, G. Zhang, and W. Tan, "Self-Adaptive Learning PSO-Based Deadline Constrained Task Scheduling for Hybrid IaaS Cloud", *IEEE transactions on automation science and engineering*, Vol. 11, No. 2, pp. 564-573, April 2014.