An Energy Efficient Hybrid PSO Algorithm in Cloud Environment

Ranjandeep Kaur Khera, V. K. Banga

***Abstract:** Algorithms exist to schedule various tasks in real time cloud environment. Nowadays many researchers are trying to schedule heavily loaded situations in real time cloud environment using swarming technique. For such studies many parameters need to be considered like cost of the system, processor latency, number of tasks and so on. With the increase in the number of tasks in the set, processing time also increases. In this situation, processor latency is at peak as the number of tasks increases and system costs increase. So the above mentioned problem is handled by proposing a task scheduler that uses a PSO algorithm to remove the limitations of past studies in a heavily loaded situation. The Particle Swarm optimization (PSO) and Invasive Weed Optimization (IWO) are combined to propose a new technique called the HWO algorithm. The proposed algorithm is recommended for preventive tasks in the single-processor in real-time environment systems.*

***Keywords:** Cloud Computing, Particle Swarm optimization, Invasive Weed optimization.*

I. INTRODUCTION

A cloud model is represented using a DAG (V, E) where V stands for vertex. Each vertex stands for a single task from n number of task $\{j_1, j_2, ..., j_n\}$ and E stands for edges. Edges represent the dependencies between the available tasks. If there exists a condition $(j_j, j_i) \in J_{pq}$ then j_j is represented as parent of j_i and j_i is represented as child of j_j. Any task without its parent is considered to be at initial point whereas the one without its child is known to be at its exit point. The task is big in size and has Million Instructions (MI) which has to be executed. The designed Cloud model has service provider which provides N number of resources $= \{Re_1, Re_2, \ldots, Re_N\}$, with varied processing powers and operating costs. Moreover, it is assumed that all the service Re_{ce} from the group, can execute all task that are being provided by service provider. Every resource has its processing power to execute a task rep $\in Re_{ce}$ and it is depicted in MIPS (Millions of Instruction per Second) and is represented by P_{ro}. Every cloud has its cost model pay as per use, so is the case in this cloud model also.

The Cloud's burst time $T_{bt(ip)}$, of a particular task j_j on a resource Re is calculated with the help of following equation:

$$T_{bt(ip)} = \frac{TX}{P_{ro}} \quad (1)$$

The Cloud's execution cost E_{cost} is with the help of following equation:

$$E_{cost} = \mu p T_{bt(ip)} \quad (2)$$

In the above equation μp is the per unit cost of utilizing resource Re.

The mentioned Energy model is for semiconductor based on metal oxide and is used from the capacitive energy (e_c) of logic based complementary circuits (Kessaci et al., 2011) and is as follows:

$$p_{EM} = S\, C_{cl} V_g{}^2 f_r \quad (3)$$

In above equation S stands for switch count in a single clock cycle, C_{cl} represents load of capacitance, V_g is the supplied voltage, and f_r represents the frequency. So, to calculate the energy consumed by running workflow tasks using the available resource from pool of resources can be calculated as follows:

$$E_{gy} \sum_{j=1}^{m} S\, C_{cl} V_g{}^2 f_r = T_{bt(ip)} + E_{cost} \quad (4)$$

In the above equation V_g is the HES's supplied voltage on which particular task j_j executed.

II. RELATED WORK

The efficient work schedule is the prime requirement of a real-time environment for an efficient work schedule system [1]. The system performance gets the adverse affect if these scheduled CPU times are not carefully handled for task scheduling decisions. Short (2010) proposed the first important and familiar scheduling strategies to be proven and known. These are called as Primitive non-synchronous task. These are such tasks which have the same start time and cannot be interrupted in between. On the contrary the Early Deadline First (EDF) algorithm are such in which priority scheduling algorithm create priority as per deadlines for specific jobs. As the total density of the system is less than one, N independent and primitive functions is responsible for creating the correct schedule under the EDF algorithm.

Despite EDF is the optimal scheduling algorithm, as it does not work well with increasing the workload queue. This implies that EDF cannot handle heavy loading conditions. To remove this, concept of Swarm Intelligence (SI) was introduced by Benny and Jing Wang in 1989 and is based on artificial intelligence. The Swarm intelligence involves agents or bird populations that interact with each other and their environment.

Jian-Bo (2010) proposed an algorithm, which is a suite of embedded real-time environment systems. System performance is measured by the unavailable ratio of the system. Jian-Bo also suggested an algorithm that works better than a simple EDF algorithm. The suggested system-missing ratio is lower than that of other systems in the overload state. The system analyzes input and defines priority in tasks, categorizing important or complex tasks by less important tasks.

Revised Manuscript Received on February 25, 2020.

Ranjandeep Kaur Khera, Assistant Professor, Department of Computer Science, Khalsa College for Women, Amritsar, Punjab, India.

Dr. Vijay Kumar Banga, Principal and Professor, Department of Electronics & Communication Engineering, Amritsar College of Engineering and Technology, Amritsar, Punjab, India.

According to the Xian-Bo output result, the algorithm is suggested by increasing the system workload and determining most tasks based on different priority[14].
Shah and Kotecha proposed an adaptive algorithm system based on EDF and Ant colony optimization. The overload condition is solved by hybrid algorithms using the EDF and ACO algorithms. "This algorithm provides a balance between exploration and exploitation as well as robustness and simplicity of the individual drug" [5][10]. There are two types of multiprocessor systems, called homogeneous and heterogeneous [3]. The anti-colony optimization algorithm has been shown to outperform the underlying equilibrium function [11]. The optimization algorithm for ant colony under homogeneous multiprocessor works well in overload conditions in real-time systems. Although it works well in crowded conditions, it takes longer to explore and exploit than EDF.
When the system is not overloaded has higher priority than when it is overloaded, in such situations the proposed algorithm takes advantage of EDF scheduling. He uses a centralized scheduler to convert the system to ACO, which sets the deadline and execution time for each assignment. In this model, the authors assume that the system has no problem with resource constraints. The execution of tasks in the ACO algorithm depends on the value of the pheromone determined for each scheduled task and the heuristic function [3].

III. PROPOSED TECHNIQUE

This paper proposes a load balancing technique that works according to working of particle swarm optimization and it simultaneously optimizes numerous set objectives and improves the quality of task execution while reducing energy consumption.
To achieve the desired results a Multi-objective fitness function is used:

$$\text{Max}(x) = \alpha * \frac{1}{Energy(y)} + (1 - \alpha) * Quality(y) \qquad (5)$$

Here, x defines the values achieved for each objective for each solution. Energy (y) represents energy consumed by schedule y. $Quality(y)$ indicate the quality of given schedule. Subsequent sections describe various steps which are used to achieve the best from the designed fitness function.
The following section discusses the different steps that are used to optimize the above designed fitness function.

A. Updating velocity

$$v_i^{\propto+1} = \omega v_i^{\propto} + c_1R_1 * (pBest_i - x_i^{\propto}) + c_1R_2 * (\text{gBest} - x_i^{\propto}) \qquad (6)$$

Here c1 and c2 are constants and they demonstrate cognitive coefficients. Here inertia weight is depicted using ω and it is used to control movement of a particle. The change in value of ω is directly proportional to improved value of swarm. $v_i^{\propto+1}$ depicts particle's velocity at α+1th iteration. R1 and R2 represent random number whose value can be between 0 and 1. $pBest_i$ shows the best position of ith particle. gBest best particle's position in whole population.

$$\omega = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{T} * \text{t} \qquad (7)$$

B. Updating Position Vector

$$x_i^{\mu+1} = x_i^{\mu} + x_i^{\mu-1} \qquad (8)$$

Here x_i^{μ} defines particle's position at i_{th} generation, $x_i^{\mu-1}$ represents velocity of particles at i_{th} generation.

IV. PROPOSED ALGORITHM (HWO)

This algorithm is a hybrid algorithm. It is combination of PSO (Particle Swarm Optimization) and IWO (Invasive Weed Optimization). The majority of its steps are same as PSO except the standard deviation of every generation is introduced to evenly spread new particles to search space as is the case in IWO (Invasive weed Optimization). The steps are as follows:

1. Set generation counter = 0.
    a. Set initial population of with M swarm particles by randomly allocating tasks on the available resources of cloud.
    b. Initialize all particles velocity to zero
    c. Set $pBest_i$ (personal best position) to current best solution.
2. The particles are evaluated using fitness function as per given in Eq. 5.
3. Increment A by 1
4. For every particle repeat the steps
    a. Set the particle (i)'s gbest from the using binary event selection.
    b. Calculate jth particle velocity using Eq. 6.
    c. Mutate particle jth particle position using mutation mentioned in

$$p = 1 - \frac{\text{J}}{\text{max \_J}}$$

Where J represents present generation and max_J is highest count of generations. So, for every particle a random number that lies between (0, 1) is picked.

5. Evaluate each particle present in the population according to standard deviation.

$$\propto_{int} = \frac{(int_{max} - int)^n}{(int_{max}} (\propto_{initial} - \propto_{final}) + \propto_{final}$$

Where int represents the N number of iterations and ∝ shows the standard deviation.

6. Calculate the gbest (global best) and cbest (current best) from last generation.
7. Select the best M solutions on the basis of standard deviation.
8. Update pbest and gbest for every particle.
9. Set A = A + 1.
10. If $(\text{Count} < max_{count})$ then move to step 3, otherwise show the solutions.

V. RESULTS

In the Table-I along with Fig.1 the comparative studies of ACO, PSO with proposed approach considering calculation time (calculated in seconds) is presented. From the Table-I and its corresponding Fig.1, it is clear that the proposed approach takes less time in contrast to other (above mentioned) techniques.

So, proposed technique seems to be more efficient when compared to other techniques considering calculation time. Moreover, calculation time increases as number of tasks increases. The proposed load balancing approach has quite less increase in calculation time than the previous techniques. The comparative studies of the proposed technique with other techniques depicts considerable decrease in calculation time (by 1.8812 %). It clearly shows from the analysis that proposed load balancing technique performs much better in real time cloud based environment.

Table-I: Calculation time (HWO)

No. of Tasks	PSO	ACO	Proposed
1000	1.31±0.57	0.85±0.61	0.76±0.54
1500	1.67±0.68	1.52±0.79	1.26±0.63
2000	2.36±0.93	2.20±0.74	1.57±0.60
2500	2.80±0.70	2.30±0.75	2.08±0.60
3000	2.96±0.72	2.40±0.76	2.15±0.61
3500	4.28±0.74	3.56±0.88	2.50±0.69
4000	4.56±0.78	3.75±0.85	3.41±0.51

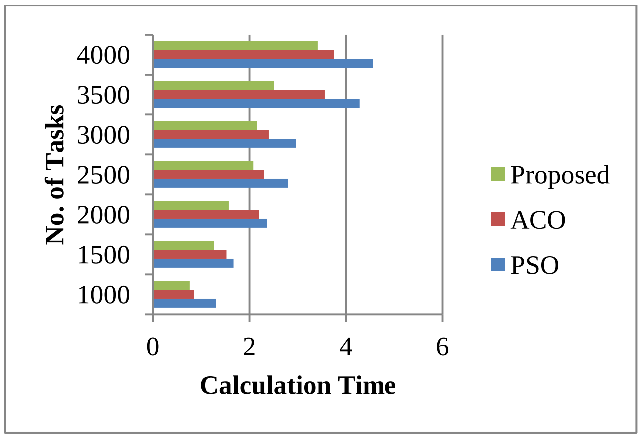

Fig.1. Calculation time analysis (HWO)

In the Table-II along with its corresponding Fig.2 the comparative studies of ACO, PSO with proposed technique considering makespan time (calculated in seconds) is presented. From the Table-II and Fig.2, it is clear that the proposed technique has less makespan time in contrast to other (above mentioned) techniques. So, proposed technique seems to be more efficient when compared to other techniques considering makespan time. Because the average decrease in makespan time (calculated in seconds) is around 5.543%. So, it is clear that the proposed technique has less makespan time when compared to the earlier techniques. Moreover, while carrying logical analysis, the proposed technique seems to be quite effective than previous techniques. As average variation in proposed technique in makespan is 122 seconds which were 162 and 157 in PSO, ACO respectively.

Table-II: Makespan (HWO)

No. of Tasks	PSO	ACO	Proposed
1000	13245±129	12675±137	11611±68
1500	16772±140	15702±158	14864±92
2000	28406±169	28621±144	26464±126
2500	34644±171	30521±160	28787±121
3000	37696±169	35486±142	34363±126
3500	38764±167	36490±170	35812±164
4000	47242±190	47666±188	46764±160

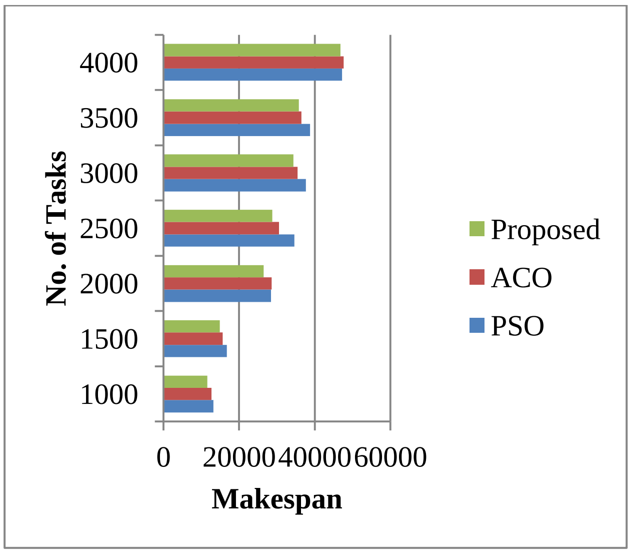

Fig.2. Makespan analysis (HWO)

The Table-III along with its corresponding Fig.3 represents the comparative studies of ACO, PSO with proposed approach on the basis of efficiency. A best schedule is the one which has efficiency close to 1. So, it is clear that the proposed approach has better efficiency. The comparative studies with other scheduling approaches depicts the average improvement in the efficiency by 0.078 %.

Table-III: Efficiency analysis (HWO)

No. of Tasks	PSO	ACO	Proposed
1000	0.77±0.029	0.80±0.044	0.88±0.012
1500	0.76±0.031	0.76±0.034	0.84±0.045
2000	0.77±0.058	0.78±0.062	0.86±0.056
2500	0.78±0.041	0.79±0.032	0.87±0.025
3000	0.71±0.067	0.73±0.064	0.84±0.063
3500	0.67±0.045	0.68±0.049	0.78±0.049
4000	0.69±0.050	0.696±0.053	0.72±0.046

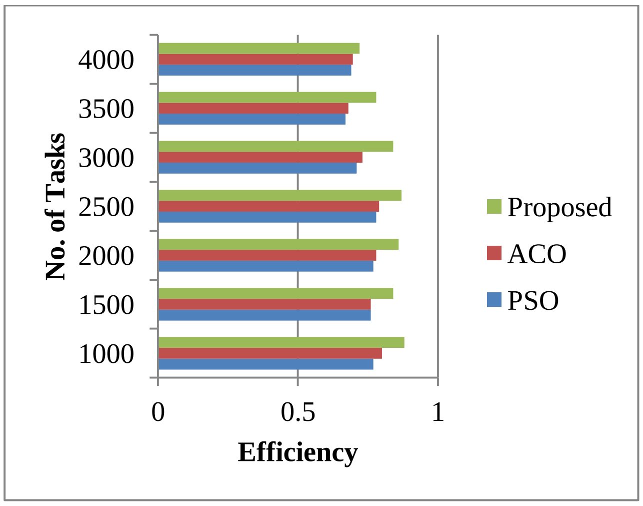

Fig.3 Comparison based on efficiency (HWO)

The Table-IV along with its corresponding Fig.4 represents the comparative studies of ACO, PSO with proposed approach on the basis of speedup. A best schedule is the one which has maximum speedup.

So, the Table-IV and Fig.4, clearly shows that the proposed approach outperforms other techniques in terms of speedup with 0.42% improvement in results.

Table-IV: Speedup analysis (HWO)

No. of Tasks	PSO	ACO	Proposed
1000	1.75±0.12	1.41±0.19	1.82±0.35
1500	1.36±0.21	1.83±0.37	1.96±0.62
2000	1.48±0.36	1.76±0.75	1.84±0.63
2500	1.39±0.57	1.88±0.36	1.94±0.49
3000	1.54±0.89	1.78±0.54	1.92±0.46
3500	1.61±0.76	1.32±0.80	1.83±0.61
4000	1.78±0.74	1.90±0.62	1.96±0.68

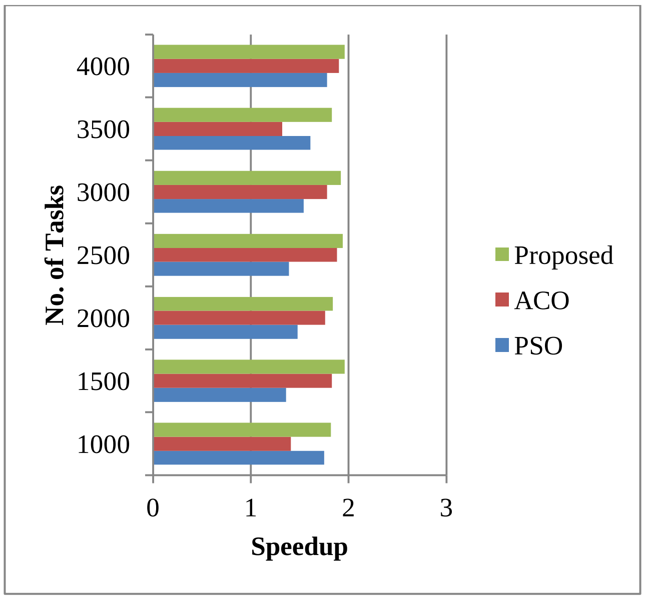

Fig.4 Comparison based on speedup (HWO)

The Table-V along with its corresponding Fig.5 represents the comparative studies of ACO, PSO and the proposed approach on the basis of energy consumption. A best schedule is the one which needs minimum energy. So, the Table-V shows that the proposed technique has 0.47% drop in energy utilization as compared to other techniques.

Table-V: Energy consumption (HWO)

No. of Tasks	PSO	ACO	Proposed
1000	0.97±0.028	0.96±0.038	0.95±0.036
1500	0.95±0.123	0.98±0.048	0.94±0.020
2000	0.88±0.063	0.94±0.081	0.86±0.062
2500	0.93±0.058	0.94±0.036	0.90±0.526
3000	1.38±0.079	0.97±0.031	0.88±0.035
3500	1.68±0.086	0.99±0.086	0.93±0.062
4000	1.87±0.081	0.98±0.096	0.78±0.072

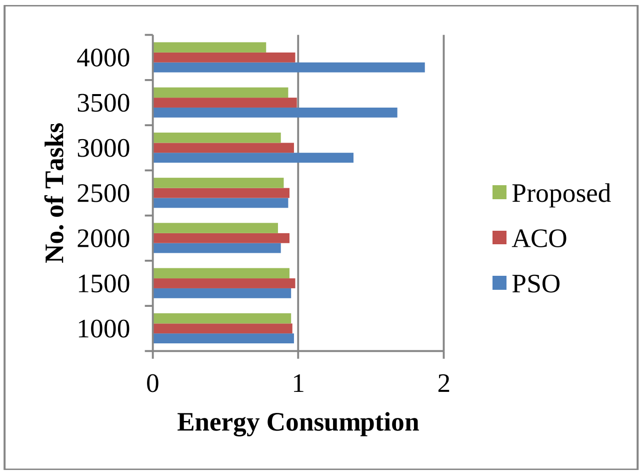

Fig.5 Comparison based on energy consumption (HWO)

VI. CONCLUSION

The process of scheduling task is quite easy but with increase in the number of jobs the complexity rises, so the completion of scheduling becomes quite tedious. The comparative analysis of the hybrid PSO with the PSO and ACO algorithm shows that, it is more suitable on the basis of various performance metrics namely speedup, energy consumption, completion time, makespan. So, the discussed load balancing technique can work efficiently in real time cloud environment.

In future, the authors are working on other performance metrics like load balancing factor using other optimization techniques.

REFERENCES

1. Andrei S. et al. (2010), *Optimal Scheduling of Urgent Preemptive Tasks*, IEEE 16th International Conference on Embedded and Real-Time Computing Systems and Applications, pp: 377-386.
2. Anon. (1995), *HRT-HOODTM: A Structured Design Method for Hard Real-Time Ada Systems*, pp: 3.
3. Apurva S. and Ketan K. (2009), *Adaptive Scheduling Algorithm for Real-Time Multiprocessor Systems*, pp: 6-7.
4. Davis R.I. and Burns A. (2011), *A survey of hard real-time scheduling for multiprocessor systems*, ACM Computing Surveys, pp: 1-44.
5. Dorigo M., Caro G. Di and Gambardella L.M. (1999), *Ant Algorithms for Discrete Optimization*. pp: 1-36.
6. Hajimirsadeghi H. and Lucas C. (2009). *A Hybrid Iwo/Pso Algorithm For Fast And Global Optimization*, pp: 1964-1971.
7. Karimi M., Motameni H. and Branch S. (2013), *Tasks Scheduling in Computational Grid using a Hybrid Discrete Particle Swarm Optimization*, pp: 29-38.
8. Kıran M.S., Gündüz M. and Baykan Ö.K. (2012*). A novel hybrid algorithm based on particle swarm and ant colony optimization for finding the global minimum*, Applied Mathematics and Computation, pp: 1515-1521.
9. Rahmani R. et al. (2013), *Hybrid technique of ant colony and particle swarm optimization for short term wind energy forecasting*, Journal of Wind Engineering and Industrial Aerodynamics, pp: 163-170.
10. Ramos V., Muge F. and Pina P. (2002), *Self-Organized Data and Image Retrieval as a Consequence of InterDynamic Synergistic Relationships in Artificial Ant Colonies*, pp: 87.
11. Shah A. and Kotecha K. (2011), *ACO Based Dynamic Scheduling Algorithm for Real-Time Multiprocessor Systems, International Journal of Grid and High Performance Computing*, pp: 20-30.
12. Short M. (2010), *Improved Task Management Techniques for Enforcing EDF Scheduling on Recurring Tasks*, 16th IEEE Real-Time and Embedded Technology and Applications Symposium, pp: 56-65.
13. Technology I.Haghnazar R. and Rahmani A.M. (2010), *Prune PSO: A new task scheduling algorithm in multiprocessors systems*, pp: 161-165.

14. Xian-bo H. (2010). *An improved EDF scheduling algorithm based on fuzzy inference being suitable for embedded soft real-time systems in the uncertain environments*, 2nd International Conference on Advanced Computer Control, pp: 588-592.

AUTHOTRS PROFILE

Ranjandeep Kaur Khera, is working as a Assistant Professor in Department of Computer Science, Khalsa College for Women, Amritsar, Punjab, India. She obtained her M.I.T. from Guru Nanak Dev University, Amritsar, Punjab, India and M. Tech. (Information Technology) from IASE Deemed University, Rajasthan, India

Dr. Vijay Kumar Banga, is working as a Principal and Professor in Department of Electronics & Communication Engineering of Amritsar College of Engineering and Technology, Amritsar, Punjab, India. He obtained his B. E (Electronics and Instrumentation Engineering) from Punjabi University, Patiala, Punjab, India. M. Tech (Electronics and Instrumentation) from Panjab University, Chandigarh, India and Ph.D. in Electronics (Artificial Intelligence) from Thapar University, Patiala., India. Presently, he has 20 years of research and UG & PG teaching experience. He has 115 research papers to his credit in various international journals and conferences. He is member of Board of Studies in Electronics and Communication Engineering of Punjab Technical University, Jalandhar, Punjab, India. He has visited various countries to present paper and to deliver invited talk in International conferences in USA, Malaysia, Indonesia, Thailand, UAE, and Singapore etc. His areas of research and interest include Autonomous Robotics, Artificial Intelligence and Image Processing.