Research Paper on Optimized Utilization of Resources Using PSO and Improved Particle Swarm Optimization (IPSO) Algorithms in Cloud Computing

[I]Gunvir Kaur, [II]Er. Sugandha Sharma
[I,II]Dept. of CSE, CGC, Gharun, Punjab, India

Abstract

In cloud computing, resources are provided as a service to its users. Dynamic services are provided by the cloud computing model which provides virtualized resources over the internet for enabling applications and these services comes from large scale data Centre's which are called as cloud. Cloud computing totally depends on internet to deliver its services to the users. Cloud computing has large number of benefits i.e. user has to pay as they use(weekly, monthly or yearly), any person having internet connection can access cloud, there is no need to buy resources(hardware, software) by themselves. This paper proposes a new algorithm i.e Hybrid improved particle swarm optimization with mutation crossover (HIPSOMC) to get maximum benefit from resources, optimized utilization of resources is important and for this scheduling plays an important role. The proposed algorithm is much more efficient then the previously designed algorithms i.e. particle swarm optimization, simulated annealing and improved particle swarm optimization and consumes less execution time.

Keywords

Cloud Computing, Particle Swarm Optimization, Simulated Annealing, Improved Particle Swarm Optimization Algorithm, Proposed Algorithm, Resource allocation.

I. Introduction

In this modern era cloud computing is one of the hottest topic which comes into market because many organizations or companies does not afford proper resources due to high cost of the software and hardware's. To overcome this problem cloud computing is originated which provides required software and hardware virtually to the users with acceptable renting cost, so organizations does not need to buy resources(hardware, software, operating system, etc.) by themselves. As electricity is provided to the users, cloud computing is also provided to the users in the same way. Users does not know where the resources are stored, where it comes from, they just use the resources and pay as per used. Cloud computing delivers an elastic execution environment of resources over the internet. Cloud computing is based on various other computing research areas i.e. grid computing, distributed computing, virtualization etc. Cloud computing has characteristics like loose coupling, service oriented, resource pooling, rapid elasticity and ease use which distinguish this from other research areas. Cloud computing is not expensive and is also very convenient. If their internet access, users can easily access any data or application on their personal computers and there is no need to install software on PC , users can directly access through internet. When people uses social networking sites or webmail's to store their photos, notes or any other important material online instead of on their personal computers, they were using cloud computing services. Cloud computing allows delivery of services over the internet, so that any user can access software and hardware which is managed by the third party at some remote locations.

Resource allocation is a challenging issue because scheduling of resources becomes harder when number of users is more than one and asks for the same resources simultaneously. Task scheduling is necessary to efficiently increase the working of cloud computing, to gain maximum profit, improving reliability and flexibility of the system. Under some logical constraints, how the tasks can be executed is the main purpose of scheduling. Generally, scheduling is concerned with allocation of resources to its users to optimize the task completion time, quality of service (QoS), etc. Task scheduling is a challenging issue to gain maximum profit and to efficiently increase working of cloud computing. Task scheduling plays a vital role to improve reliable and flexible systems. Cloud computing provides many services such as , shared resources, software , information to customers at relatively low cost. When number of customers increases, fulfilling their requirements becomes an important issue. Resource allocation is the major issue because more than one resources may be needed at the same time.

Many issues are related to task scheduling to get maximum benefit from resources proper utilization of resources is necessary task. The key technologies of cloud environment are resource management and task scheduling. Previously designed task scheduling algorithms considers only that how QoS is maintained but do not consider that how to optimize the utilization of resources to gain maximum profit. Many task scheduling algorithms have been originated and proposed. Some of the algorithms are:

- Particle Swarm Optimization Algorithm (PSO).
- Particle Swarm-Simulated Annealing Algorithm (P-S).
- Improved Particle Swarm Optimization Algorithm (IPSO).
- Proposed Algorithm

II. Related Work

Task scheduling plays similar role in both cloud and grid computing environment. To get maximum benefit from resources, optimized utilization of resources is important and for this scheduling plays an important role. Many scheduling algorithms were designed so that proper utilization of resouces and guareenting the QoS. [Arash Ghorbannia Delavar, Yalda Aryan,2011] adopted genetic algorithm to solve the NP-hard problem by considering many objectives like total makespan, resource load balancing as well as QoS and finds an ideal solution for task scheduling in cloud environment[3]. [Pinal Salot,2013] analyze various task scheduling algorithms and issues related to them and conclude that disk space management is a critical issue in virtual environment. This paper also conclude that Batch mode heuristic scheduling algorithms(BMHA) and Online mode heuristic scheduling algorithms gives high throughput and cost effective but do not consider reliability and availability[4]. [Suraj Pandey, Linlin Wu, Siddeswara Mayura Guru, Rajkumar,2 009] presents a scheduling heuristic method based on particle swarm optimization(PSO) to minimize total cost of execution . This paper compares PSO and 'Best Resource Selection'(BRS) algorithm and results show that PSO is better because it achieves three times cost saving as compared to BRS and better distribution of workload on resources[5].

[Mingyue Feng, Xiao Wang, Yongjin Zhang, Jianshi Li,2012] concerns with problem of resource allocation. So a particle swarm optimization algorithm is designed to overcome this problem by introducing Pareto-dominate theory to the algorithm. This theory search for optimal schedulers in the multi-objective optimizing matter on resources based on total task executing time, resource reservation, and QoS of each task[6]. [Xingquan Zuo, Guoxiang zhang, Wei Tan,2013] proposes a self-adaptive learning PSO(SLPSO)-based scheduling approach which is used to effectively schedule inter-cloud resources. When resources are not sufficient, Iaas provider outsources its workload to the external clouds. This paper also compares SLPSO with cloud federation and results that this approach finds optimal and sub-optimal allocation scheme of internal and external resources which improves the quality and maximize the profit of IaaS provider[7].

[Simsy Xavier, S. P. Jeno Lovesum,2013] surveys various scheduling algorithms and compare their parameters. Existing workflow algorithms i.e. PSO based heuristic, Heterogeneous Earliest Finish Time Algorithm (HEFT) etc. does not consider the execution time so there is need of new scheduling algorithm that minimize the execution time[8]. [Prof. Dr. Jayat. S. Umale, Priyanka A. Chaudhari,2013] presents revised discrete particle swarm optimization to optimize the schedules of workflow application in cloud computing environment. This method is used to improve the performance of algorithm and also reduces search space [9].

[Shaobin Zhan, Hongying Huo, 2012] proposes improved particle swarm optimization which shortens the average operation time, increases utilization ratio of resources and supplies proper resources to user task efficiently. PSO does not solve large scale optimization then simulated annealing algorithm is added into the PSO algorithm which avoids sinking into the local optima and also increases the convergence speed of PSO . The results shows that the proposed method redues the task average running time and the rate of availability of resources is raised [10]. [Yi Cai, Zhutian Chen, Huaqing Min,2013] proposes a swarm intelligence based algorithm which reduces searching time. Traditional PSO algorithm is modified with a random factor to tackle with the premature convergence problem. Results show that proposed system is more feasible, robust and scalable than previous methods. Proposed method gives better performance when compared with traditional PSO-based algorithm. Total time taken to complete the task is shorter and stable than traditional PSO-based method which results in low power consumption [11].

III. Particle swarm optimization (PSO) algorithm

Particle swarm optimization (PSO) is a group based searching algorithm. PSO is similar to Genetic algorithm (GA) which is based on population but in PSO re-combination of individuals of the population is not direct. PSO is a self-adaptive global search based optimization technique introduced by Kennedy and Eberhart [12]. In wide range of applications PSO is simple and effective with low computational cost and fast speed that is the reason PSO becomes popular but it has low convergence accuracy.

A group of particles are used to represent the solution . Every particle uses two vectors i.e. velocity and position. Firstly, initialize particle with random position and velocity vectors in the search space. Every particle will have fitness value then calculate fitness value for each particle(p) will be evaluated according to its present position. Every particle knows its best position (pbest),which stores the best solution. If fitness(p) is better than fitness(pbest) than pbest=p. After this ,for all particles, select the best particle as gbest. At last, calculate and update the velocity and position of each particle using equations (1) and (2).

$$v_i^{t+1} = \omega v_i^{\,t} + c_1 r_1 (pbest_i - x_i^t) + c_2 r_2 (gbest_d - x_i^t) \quad (1)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (2)$$

Where:

v_i^t = velocity of ith particle at iteration t.

v_i^{t+1} = velocity of ith particle at iteration t+1.

$pbest_i$ = best position of ith particle.

$gbest$ = position of best particle in a population.

x_i^t = current position of ith particle at iteration t.

x_i^{t+1} = position of ith particle at iteration t+1.

c_1 and c_2= acceleration constants.

r_1 and r_2= random values.

ω= inertia weight.

Algorithm 1 – Particle Swarm Optimization (PSO) algorithm

Step 1: Initialize particles with some random position and velocity vectors.

Step A: Calculate fitness value for every particle.

Step B: If fitness(p) better than fitness(pbest) than pbest=p.

Step C: Assign best particle pbest value to gbest.

Step D: Calculate velocity for each particle using equation (1).

Step E: Update particle velocity and position using equation (2).

Overall performance of PSO is increased by the inertia weight. If the value of inertia weight is large it encourage global exploration but if it is small than it tends to promote local exploration. To provide balance between global and local exploration and to reduce number of iterations to find the optimum solution suitable selection of inertia weight is necessary. When value of inertia weight is decreased than performance is increased. Many combinational optimization problems are successfully solved by PSO. Some applications that used PSO are: chemical engineering [13], pattern recognition [14], reactive voltage control problem [15] etc.

IV. Particle Swarm Simulated-Annealing (P-S) Algorithm

PSO algorithm has many advantages like easy to realize, high flexibility, strong robustness, scalability etc. because of these merits PSO solves many combinational problems. But there are some disadvantages in PSO, convergence rate of PSO is low when solving large scale optimization problems and PSO easily sink into defects of local optima due its strong randomicity. PSO is good during intial phase but while going through iterations convergence rate becomes low, particles lose variety. To overcome these problems there is need of some other algorithm then PSO is combined with Simulated annealing and mixed algorithm is designed i.e. P-S algorithm. This algorithm provides better results by combining the properties of both algorithms i.e. PSO has fast searching ability to search better swarm and SA has jumping ability to partly optimize a better individual so, by advantages of both PSO and SA, probability and speed of convergence to the optimum solution is improved. SA algorithm takes optimal individuals as initial solutions which advances scheduling efficiency.

V. Improved Particle Swarm Optimization (Ipso) Algorithm

Firstly, select N initial solutions randomly , initialize ω, c_1, c_2 and inertia weight decreases coefficient λ, set annealing coefficient β and intial temperature, the minimum sampled length U_{min} and maximum sampled length U_{max}. Then at the end of iteration, update x_i^t and calculate fitness. If particle fitness is better than $pbest_i$ set $pbest_i = x_i^t$. If particle fitness is better than $gbest_i$ set $gbest_i = x_i^t$. Generate new state from $gbest_i$ and set new state by accepting rules. This strategy avoids sinking into local optima and population diversity is increased. IPSO based cloud computing server cluster fastly discover resources, task execution resource matching. Information flows in one direction in global PSO system, $gbest^t$ transfer information to other particles further other particles search for the near $gbest^t$ i.e. whole group of particles evolve to the optima with $gbest^t$. PSO algorithm is premature because it has poor searching ability for $gbest^t$. PSO has fast searching ability to search better swarm and SA has jumping ability to partly optimize a better individual. Advantages of both PSO and SA are combined and this combined algorithm is known as improved particle swarm optimization algorithm.

In this method to get better optimization and fast convergence component is divided into two parts i.e. good experience (pbest) of each particle and bad experience (pworst) of each particle. Each particle remembers its pbest and pworst to reach the destination in lesser time and with optimum solution. By using both experiences particle may bypass its worst position and reach the final position easily and fastly. At last, velocity and position of particle is calculated and updated by using the equation(3).

$$v_i^{t+1} = \omega v_i^t + c_{1g} r_1 (pbest_i - x_i^t) pbest_i + c_{1b} r_2 (x_i^t - pworst_i)$$
$$pworst_i + c_2 r_3 (gbest - x_i^t) \qquad (3)$$

Where:

v_i^t = velocity of ith particle at some iteration t.

v_i^{t+1} = velocity of ith particle at iteration t+1.

$pbest_i$ = best position of ith particle.

$gbest$ = position of best particle in whole population.

x_i^t = current position of ith particle at some iteration t.

x_i^{t+1} = position of ith particle at iteration t+1.
$pworst_i$ = worst position of ith particle
c_1 and c_2 = acceleration constants.
r_1, r_2 and r_3 = random values.
ω= inertia weight.

Algorithm 2 – Improved Particle Swarm Optimization (PSO) algorithm

Step 1: Initialize each particle with random position and velocity vectors.
Step 2: Randomly select N initial solutions and initialize ω, c_1, c_2 , to search optimal solution.
Step 3: If fitness(p) better than fitness(pbest) than pbest=p.
Step 4: Assign best particle pbest value to gbest.
Step 5: Select the particle individual worst value i.e (pworst) when particle is away from solution
Step 6: Update pbest, gbest, pworst, position and velocity using equation (3).

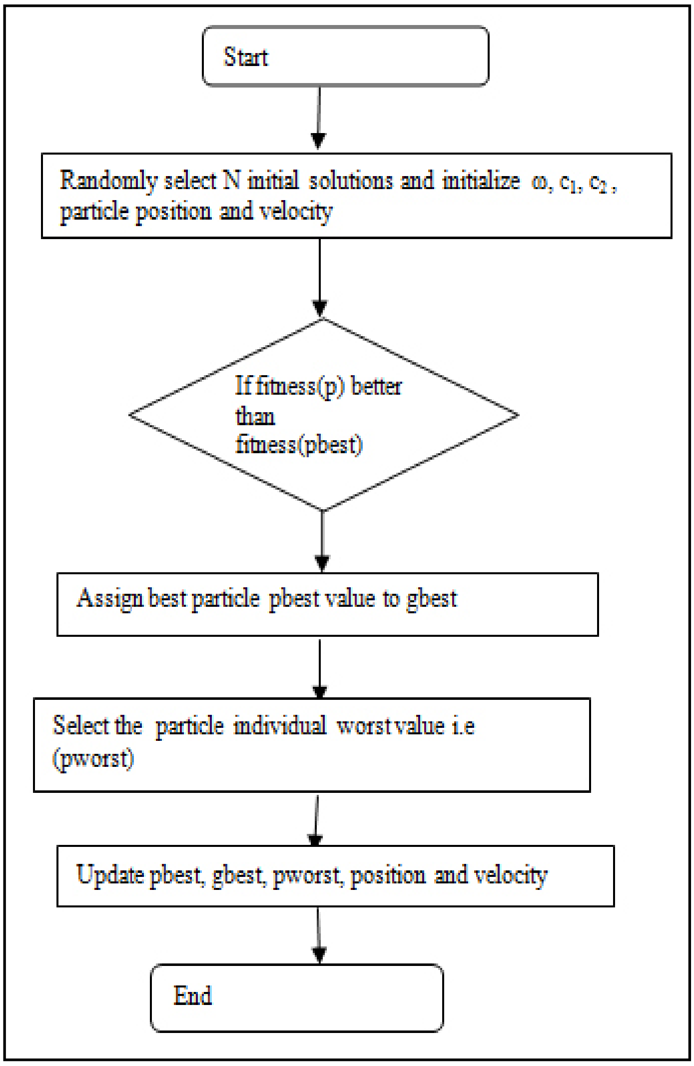


Fig. 1 : IPSO

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

VI. Proposed Algorithm (Hybrid Improved Particle Swarm Optimization With Mutation CrossoveR)

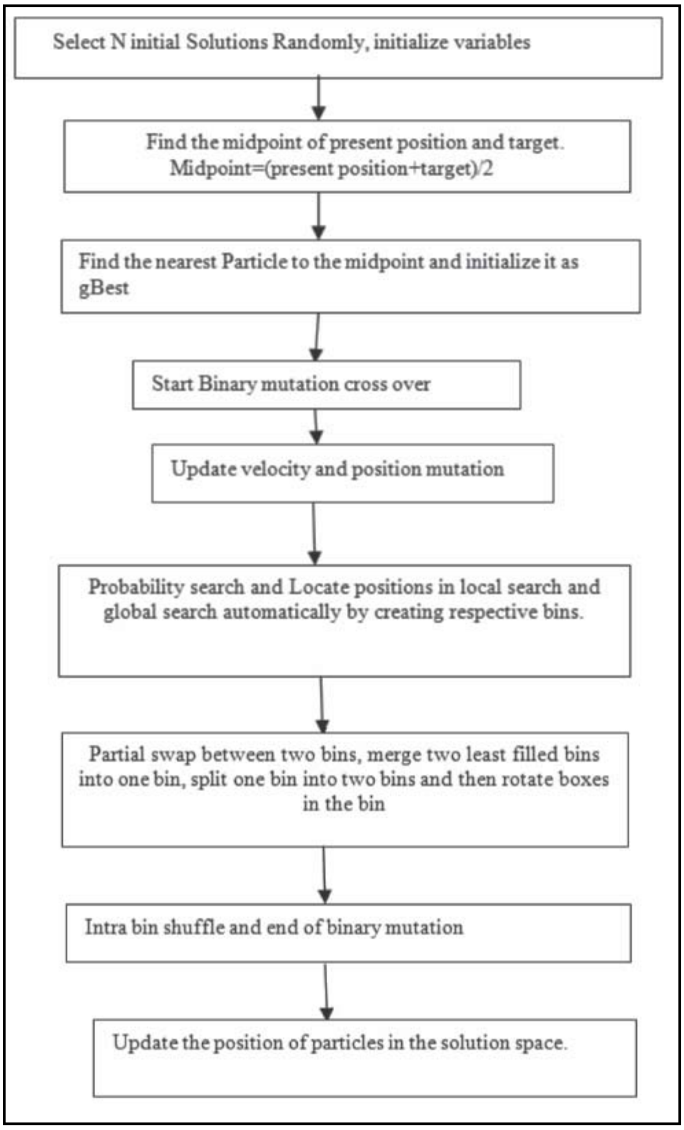


Fig. 2 : Proposed Algorithm (Hybrid IPSO with Mutation Crossover)

Algorithm 3 – Improved Particle Swarm Optimization (PSO) with mutation crossover algorithm

Step 1. Select N initial solutions randomly, initialize variables.
Step 2. Find midpoint of present position and target particles
Step 3. Find the nearest particle to the midpoint and initialise it as gbest (stimulated annealing algorithm).
Step 4. Start of binary mutation cross over.
Begin :- create an initialize, while (stop condition is false).
Begin
Evaluation
Update velocity and position mutation
End
Step 5. Probability search and Locate positions in local search and global search automatically by creating respective bins.
Step 6. Partial swap between two bins, merge two least filled bins into one bin, split one bin into two bins.
Step 7. Rotate boxes in the bin.
Step 8. Intra bin shuffle.
Step 9. End of binary mutation and cross over.
Step 10. Preservation of non-dominated particles in archive.
Step 11. Update the position of particles in the solution space.

$$v_i(t+1) = w\ _i(t) + c_1 r_1 [p\ .i(t)] + c_2 r_2 [g\ (t) - x_i(t)] \quad (4)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (5)$$

Where v_i and x_i are the particle velocity and position the tth iteration respectively. C_1 and C_2 are the scaling constants, usually taken as C1=C2=2.0. w is the iteration weight and used to control trade off between the global and local exploration ability of swarm. Random numbers r_1 and r_2 are uniformly distributed in [0,0].

$$v_i(t+1) = w\ _i(t) + [1 + r_3) p\ .i(t) - x_i(t)] /3 + [1 + r_4) g\ (t) - x_i(t)] /3 \quad (6)$$

Update the position of particles using equation (5). Randomly initialize all particles with some random positions and velocities in the search space.
While (termination conditions are not met)
{
For each particle i do
{
Update the position of ith particle according to the equation (4)
Update the position of ith particles according to the equation (5)
Map the position of ith particle in the solution space and evaluate its fitness value according to the fitness function
Update pbesti(t) and gbesti(t) if necessary according to equation (5) and(6)
}
}

VII. Results and Discussion

The implementation of algorithm is done in this section. The standards are also described and implementation is done in three sections:

1. CPU performance, response time between the user and the resources.
2. Execution time consideration for local search and global search.
3. Inputs of each particle in the group(number of particles).

1. CPU performance, response time between the user and the resources:

Results of the implementation are discussed in this section. we will compute the results for the values of CPU performance Simulated Annealing with mutation and crossover which computes utilization time which are proposed to handle problems between consumer (user) and resources caused by network properties. Total time taken to establish a connection between the user and the resources at different ports is calculated at different instances.

Table 1: Shows execution time at different instances to establish a connection between user and resources

Execution Time in seconds		
Epoch	PSO Algo	Proposed Algo
1	1.044	1.467
2	3.058	5.951
3	0.861	1.369
4	4.749	2.293
5	4.640	1.972
6	2.400	2.588
7	3.327	0.771
8	2.013	1.815
9	0.934	1.853
10	5.900	0.805
Average	28.926	23.159

As evaluated from the above table, it is observed that there is a big difference between the execution time taken to reach the specified target by the particles in PSO algorithm and Proposed algorithm. Proposed algorithm is more efficient as the average time taken to reach the target is less in proposed algorithm as compared to PSO. And new algorithm is more beneficial as it saves both time and cost of execution of particles.

2. Execution time consideration for local search

Evaluation of results shows that in local search total number of particles is first taken Upto 500 and Epoch number(Inputs) is also set to 1500. Comparison of both the algorithm is done. Simailar results are shown when total number of particles taken is increased from 500 to 1000 and so onAs the experimental result shows that the average execution time in local search is very less as compared to PSO when particles are taken less than 500 in number in hybrid PSO. Results are obtained as we increase the number of particles from 500 to 1500 and so on. Better performance is seen in proposed algorithm.

Table 2: Shows the comparison of average execution time of PSO and Proposed algorithm in Local search in Seconds

Epoch	Number of Particles	PSO Algo	Proposed algo
33	245	0.23	0.001
37	319	0.56	o.49
63	220	0.90	0.63
86	309	110.0	0.86
110	322	112.2	0.89
158	394	124.0	0.91
274	445	133.4	0.99
447	403	136.3	1.02
1110	115	144.7	1.19
2980	218	160.09	1.56
Averag-e		92.238	0.8541

3. Execution time consideration for global search

Table 3: Shows the comparison of average execution time of PSO and Proposed algorithm in Global search in Seconds

Epoch	Number of Particles	PSO Algo	Propo-sed algo
1	9999	81.01	78.01
2	9999	81.19	78.79
3	9999	81.90	79.67
4	9999	82.00	79.90
5	9999	83.50	79.93
6	9999	93.05	88.52
7	9999	93.98	89.59
8	9999	94.10	89.90
9	9999	96.14	90.96
10	9999	116.09	110.32
Average		90.296	86.559

Proposed algorithm is more efficient for global search also as the average time taken to reach the target is less in proposed algorithm as compared to PSO. And new algorithm is more beneficial as it saves both time and cost of execution of particles.

4. Inputs of each particle in the group(number of particles).

Epoch number of each particle corresponding to the total number of particles commuted and time of execution will be calculated in this section. We have taken the same specification as in PSO and comparison of results in terms of execution time will be calculated. The average execution time to achieve the target is calculated by manually updating the number of particle and number of inputs of each particle by using both previously existing algorithm i.e. PSO and the new algorithm i.e Hybrid algorithm (PSO with mutation & Crossover). The comparison of both the algorithm is shown in the tables below in number of cases.
The number of particles is set to Maximum
The number of inputs each particle is maximum
Comparison of average execution time of both the algorithms:

Table 4 : Shows the comparison of average execution time of PSO and Hybrid PSO for both Local search and Global Search

Execution Time in seconds		
Epoch	PSO Algo	Propo-sed Algo
1	2.368	0.872
2	4.841	0.467
3	4.188	0.687
4	2.880	0.282
5	0.206	0.687
6	3.428	1.405
7	3.447	0.303
8	2.753	1.213
9	0.314	0.308
10	2.574	1.497
Average	2.6993	1.497

The best output values are considered for obtaining the average execution time of new proposed algorithm and PSO .The Proposed algorithm takes very less execution time as compared to other algorithm. The average of execution time of Particle Swarm Optimization is 2.699 and proposed algorithm's value is 1.497 when the number of particle are chosen maximum and number of inputs of each particles are also maximum. The highest value of execution time in PSO is 3.428 where as in case of proposed algorithm the highest value is 1.497. This shows that there is a big difference between the execution time taken to reach the specified target by the particles. And new algorithm is more beneficial as it saves both time and cost of execution of particles.

As the average execution time decrease the cost of execution also decreases The results are also compared graphically in the fig. In this graph it is worth noting that as the number of particle increases the difference between the average execution time (in second) of both the algorithms increases. Proposed Hybrid algorithm consumes less time as compared to the Particle Swarm Optimization algorithm.

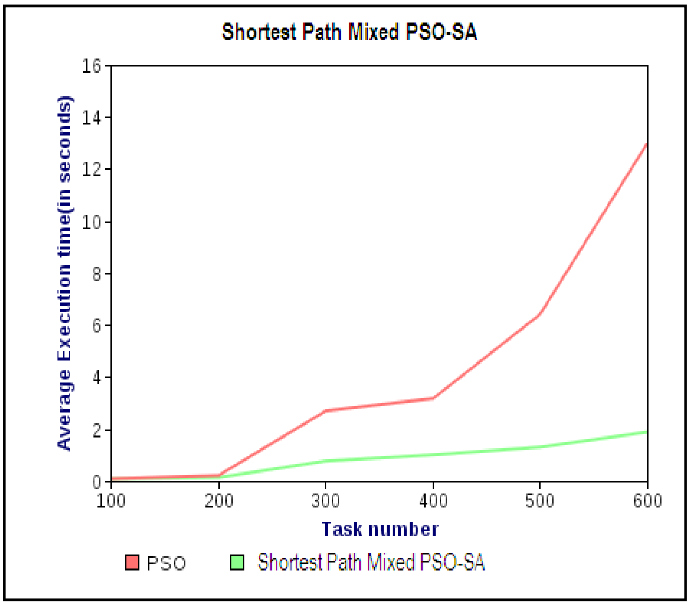


Fig. Graph shows the Overall performance analysis of PSO and Hybrid IPSO with Mutation Crossover algorithm

Where red line indicates PSO and green line indicates Proposed algorithim i.e Hybrid IPSO with mutation crossover.

VIII. Conclusion

PSO algorithm has many defects such as convergence rate of PSO is low while solving large scale optimization problems and PSO easily sink into defects of local optima due its strong randomicity. PSO algorithm is good during intial phase but while going through iteration particles lose variety. Then simulated annealing algorithm is combined with PSO algorithm to solve the problems of PSO which increases convergence rate and not sink into local optima i.e improved particle swarm optimization. Performance of cloud environment is directly affected by the efficiency of scheduling algorithms. Improved particle swarm optimization (IPSO) algorithm improves average operation time required to complete a task, provides proper resources to the user efficiently. Experimental shows that the difference between the average execution time of both the algorithms increases. The new algorithm i.e. proposed algorithm i.e Hybrid improved particle swarm optimization with mutation cross is much more efficient as it consumes very less time as compared to Particle Swarm Optimization algorithm. Proposed algorithm consumes less time as compared to the Particle Swarm Optimization algorithm. And new algorithm is more beneficial as it saves both time and cost of execution of particles.

References

[1] *M.-E. BEGIN, An egee comparative study: Grids and clouds – evolution or revolution. EGEE III project Report, vol. 30 (2008).*

[2] *XIAO Zhi-Jiao, CHANG Hui-You, YI Yang "An Optimization M ethod of W orkflow Dynamic Scheduling Based on Heuristic GA", Computer Science, Vo1.34 No.2 2007.*

[3] *Arash Ghorbannia Delavar, Yalda Aryan. A Synthetic Heuristic Algorithm for Independent Task Scheduling in Cloud Systems. IJCSI International Journal of Computer Science Issues, Vol. 8, No 2, 2011, pp.289-295.*

[4] *Pinal Salot, " A Survey of Various Scheduling Algorithms in Cloud Computing Enviornment" IJRET, Vol.2, Issue.2,2013.*

[5] *Suraj Pandey, Linlin Wu, Siddeswara Mayura Guru, Rajkumar Buyya, " A Particle Swarm Optimization-based Heuristic for Scheduling Workflow Applications in Cloud Computing Enviornments" technical reports, University of Melbourne,2009.*

[6] *Mingyue Feng, Xiao Wang, Yongjin Zhang, Jianshi Li, "Multi-Objective Particle Swarm Optimization For Resource Allocation In Cloud Computing" IEEE,2012.*

[7] *Xingquan Zuo, Guoxiang zhang, Wei Tan, "Self-Adaptive Learning PSO-Based Deadline Constrained Task Scheduling for Hybrid IaaS Cloud" IEEE Transaction on Automation Science and Engineering,2013.*

[8] *Simsy Xavier, S. P. Jeno Lovesum, " A Survey of Various Workflow Scheduling Algorithms in Cloud Environment " International Journal of Scientific and Research Publication (IJSRP), vol. 3, Issue 2, 2013.*

[9] *Prof. Dr. Jayat. S. Umale, Priyanka A. Chaudhari, " Survey on Job Scheduling Algorithms of Cloud Computing" International Journal of Computer Science and Management Research(IJCSMR), ISSN 2278-733X, 2013.*

[10] *Shaobin Zhan, Hongying Huo, "Improved PSO-based Task Scheduling Algorithm in Cloud Computing" Journal of Information and Computational Science(JOICS) 9:13(2012) 3821-3829 .*

[11] *Yi Cai, Zhutian Chen, Huaqing Min, " Improving Particle Swarm Optimization Algorithm for Distributed Sensing and Search" International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, IEEE,2013.*

[12] *J. Kennedy and R. Eberhart, Particle Swarm Optimization. In IEEE International Conference on Neural Networks, Vol. 4, pages 1942-1948,1995.*

[13] *C . u. O. Ourique, E. C. J. Biscaia, and J. C. Pinto. The use of particle swarm optimization for dynamical analysis in chemical processes. Computers and Chemical Engineering,26(12):1782-1793, 2002.*

[14] *J. Louchet, M. Guyon, M. J. Lesot, and A. Boumaza. Dynamic files: a new pattern recognition tool applied to stereo sequence processing. Pattern Recognition Letters, 23(1-3):335-345,2002.*

[15] *H. yoshida, K. Kawata, Y. Fukuyama, and Y. Nakanishi. A particle swarm optimization for reactive power and voltage*

control considering voltage stability. In the International Conference on Intelligent System application to Power System, pages 117-121,1999.

[16] C . u. O. Ourique, E. C. J. Biscaia, and J. C. Pinto. The use of particle swarm optimization for dynamical analysis in chemical processes Computers and Chemical Engineering,26(12):1782-1793, 2002.

[17] J. Louchet, M. Guyon, M. J. Lesot, and A. Boumaza. Dynamic files: a new pattern recognition tool applied to stereo sequence processing. Pattern Recognization Letters, 23(1-3):335-345,2002.

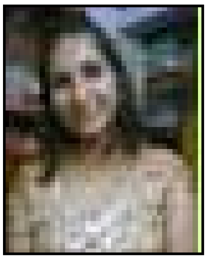
[18] H. yoshida, K. Kawata, Y. Fukuyama, and Y. Nakanishi. A particle swarm optimization for reactive power and voltage control considering voltage stability. In the International Conference on Intelligent System application to Power System, pages 117-121,1999

[19] A.P. Engelbrecht, " Fundamentals of Computational Swarm Intelligence", John Wiley & Sons, 2005.

[20] R.C. Eberhart and Y. Shi, Comparison between Genetic Algorithm and Particle Swarm Optimization", Evolutionary Programming VII 919980, Lecture Notes in Computer Science 1447,pp611-616,Springer.

Authors

Er. GUNVIR KAUR received B,tech degree in computer science and technology from Baba farid group of institutions (PTU)and now pursuing M.Tech (CSE) from CGC Gharuan .Research interest includes cloud computing and task scheduling



ER. SUGANDHA SHARMA is working as Assistant professor in CGC Gharuan in CSE department and she has almost 4 year teaching experience and she has done her M.E. From Punjab university. Her research interest includes grid computing and cloud computing.