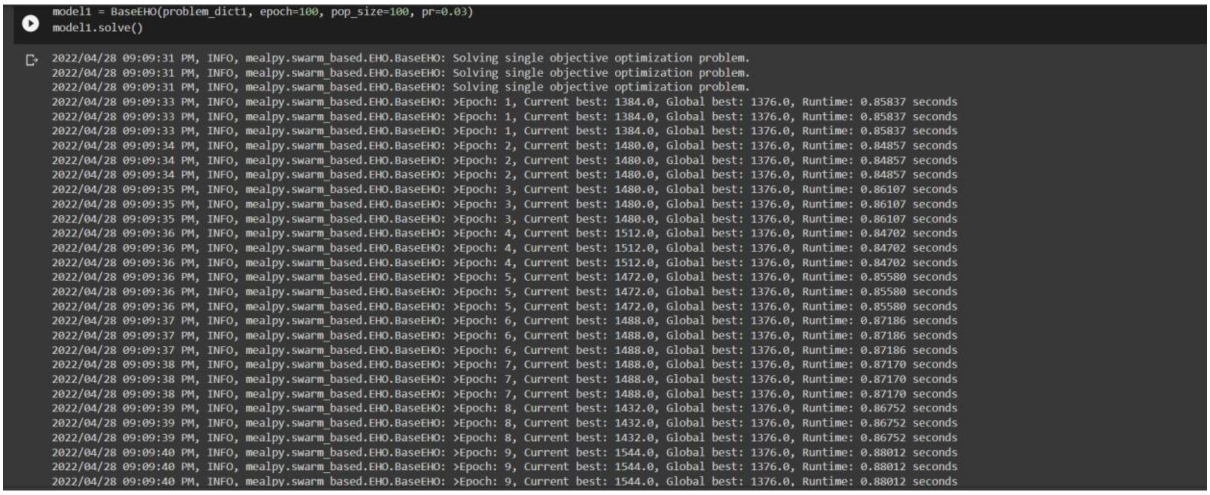
RESULTS

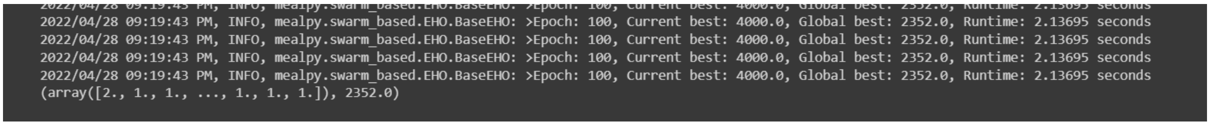
Code Output-



```
model1 = BaseEHO(problem_dict1, epoch=100, pop_size=100, pr=0.03)
model1.solve()

2022/04/28 09:09:31 PM, INFO, mealpy.swarm_based.EHO.BaseEHO: Solving single objective optimization problem.
2022/04/28 09:09:31 PM, INFO, mealpy.swarm_based.EHO.BaseEHO: Solving single objective optimization problem.
2022/04/28 09:09:31 PM, INFO, mealpy.swarm_based.EHO.BaseEHO: Solving single objective optimization problem.
2022/04/28 09:09:33 PM, INFO, mealpy.swarm_based.EHO.BaseEHO: >Epoch: 1, Current best: 1384.0, Global best: 1376.0, Runtime: 0.85837 seconds
2022/04/28 09:09:33 PM, INFO, mealpy.swarm_based.EHO.BaseEHO: >Epoch: 1, Current best: 1384.0, Global best: 1376.0, Runtime: 0.85837 seconds
2022/04/28 09:09:33 PM, INFO, mealpy.swarm_based.EHO.BaseEHO: >Epoch: 1, Current best: 1384.0, Global best: 1376.0, Runtime: 0.85837 seconds
2022/04/28 09:09:34 PM, INFO, mealpy.swarm_based.EHO.BaseEHO: >Epoch: 2, Current best: 1480.0, Global best: 1376.0, Runtime: 0.84857 seconds
2022/04/28 09:09:34 PM, INFO, mealpy.swarm_based.EHO.BaseEHO: >Epoch: 2, Current best: 1480.0, Global best: 1376.0, Runtime: 0.84857 seconds
2022/04/28 09:09:34 PM, INFO, mealpy.swarm_based.EHO.BaseEHO: >Epoch: 2, Current best: 1480.0, Global best: 1376.0, Runtime: 0.84857 seconds
2022/04/28 09:09:35 PM, INFO, mealpy.swarm_based.EHO.BaseEHO: >Epoch: 3, Current best: 1480.0, Global best: 1376.0, Runtime: 0.86107 seconds
2022/04/28 09:09:35 PM, INFO, mealpy.swarm_based.EHO.BaseEHO: >Epoch: 3, Current best: 1480.0, Global best: 1376.0, Runtime: 0.86107 seconds
2022/04/28 09:09:35 PM, INFO, mealpy.swarm_based.EHO.BaseEHO: >Epoch: 3, Current best: 1480.0, Global best: 1376.0, Runtime: 0.86107 seconds
2022/04/28 09:09:36 PM, INFO, mealpy.swarm_based.EHO.BaseEHO: >Epoch: 4, Current best: 1512.0, Global best: 1376.0, Runtime: 0.84702 seconds
2022/04/28 09:09:36 PM, INFO, mealpy.swarm_based.EHO.BaseEHO: >Epoch: 4, Current best: 1512.0, Global best: 1376.0, Runtime: 0.84702 seconds
2022/04/28 09:09:36 PM, INFO, mealpy.swarm_based.EHO.BaseEHO: >Epoch: 4, Current best: 1512.0, Global best: 1376.0, Runtime: 0.84702 seconds
2022/04/28 09:09:36 PM, INFO, mealpy.swarm_based.EHO.BaseEHO: >Epoch: 5, Current best: 1472.0, Global best: 1376.0, Runtime: 0.85580 seconds
2022/04/28 09:09:36 PM, INFO, mealpy.swarm_based.EHO.BaseEHO: >Epoch: 5, Current best: 1472.0, Global best: 1376.0, Runtime: 0.85580 seconds
2022/04/28 09:09:36 PM, INFO, mealpy.swarm_based.EHO.BaseEHO: >Epoch: 5, Current best: 1472.0, Global best: 1376.0, Runtime: 0.85580 seconds
2022/04/28 09:09:37 PM, INFO, mealpy.swarm_based.EHO.BaseEHO: >Epoch: 6, Current best: 1488.0, Global best: 1376.0, Runtime: 0.87186 seconds
2022/04/28 09:09:37 PM, INFO, mealpy.swarm_based.EHO.BaseEHO: >Epoch: 6, Current best: 1488.0, Global best: 1376.0, Runtime: 0.87186 seconds
2022/04/28 09:09:37 PM, INFO, mealpy.swarm_based.EHO.BaseEHO: >Epoch: 6, Current best: 1488.0, Global best: 1376.0, Runtime: 0.87186 seconds
2022/04/28 09:09:38 PM, INFO, mealpy.swarm_based.EHO.BaseEHO: >Epoch: 7, Current best: 1488.0, Global best: 1376.0, Runtime: 0.87170 seconds
2022/04/28 09:09:38 PM, INFO, mealpy.swarm_based.EHO.BaseEHO: >Epoch: 7, Current best: 1488.0, Global best: 1376.0, Runtime: 0.87170 seconds
2022/04/28 09:09:38 PM, INFO, mealpy.swarm_based.EHO.BaseEHO: >Epoch: 7, Current best: 1488.0, Global best: 1376.0, Runtime: 0.87170 seconds
2022/04/28 09:09:39 PM, INFO, mealpy.swarm_based.EHO.BaseEHO: >Epoch: 8, Current best: 1432.0, Global best: 1376.0, Runtime: 0.86752 seconds
2022/04/28 09:09:39 PM, INFO, mealpy.swarm_based.EHO.BaseEHO: >Epoch: 8, Current best: 1432.0, Global best: 1376.0, Runtime: 0.86752 seconds
2022/04/28 09:09:39 PM, INFO, mealpy.swarm_based.EHO.BaseEHO: >Epoch: 8, Current best: 1432.0, Global best: 1376.0, Runtime: 0.86752 seconds
2022/04/28 09:09:40 PM, INFO, mealpy.swarm_based.EHO.BaseEHO: >Epoch: 9, Current best: 1544.0, Global best: 1376.0, Runtime: 0.88012 seconds
2022/04/28 09:09:40 PM, INFO, mealpy.swarm_based.EHO.BaseEHO: >Epoch: 9, Current best: 1544.0, Global best: 1376.0, Runtime: 0.88012 seconds
2022/04/28 09:09:40 PM, INFO, mealpy.swarm_based.EHO.BaseEHO: >Epoch: 9, Current best: 1544.0, Global best: 1376.0, Runtime: 0.88012 seconds
```

Final array displaying Global Best.



```
2022/04/28 09:19:43 PM, INFO, mealpy.swarm_based.EHO.BaseEHO: >Epoch: 100, Current best: 4000.0, Global best: 2352.0, Runtime: 2.13695 seconds
2022/04/28 09:19:43 PM, INFO, mealpy.swarm_based.EHO.BaseEHO: >Epoch: 100, Current best: 4000.0, Global best: 2352.0, Runtime: 2.13695 seconds
2022/04/28 09:19:43 PM, INFO, mealpy.swarm_based.EHO.BaseEHO: >Epoch: 100, Current best: 4000.0, Global best: 2352.0, Runtime: 2.13695 seconds
2022/04/28 09:19:43 PM, INFO, mealpy.swarm_based.EHO.BaseEHO: >Epoch: 100, Current best: 4000.0, Global best: 2352.0, Runtime: 2.13695 seconds
2022/04/28 09:19:43 PM, INFO, mealpy.swarm_based.EHO.BaseEHO: >Epoch: 100, Current best: 4000.0, Global best: 2352.0, Runtime: 2.13695 seconds
(array([2., 1., 1., ..., 1., 1., 1.]), 2352.0)
```

Comparing Execution Time (Global Best) of Elephant herd optimization VS Particle swarm optimization-

Parameters Used-

- No. of VMs (5,10)
- Epoch
- Population Size
- No. of task
- Global best

When VM=5

Epoch	Population Size	No. of Task	Global Best (EHO)	Global Best (PSO)
100	100	100	112	95.9
100	100	500	192	571
100	100	1000	824	1187
100	100	1500	992	1733
100	100	2000	2272	2318
100	100	2500	2352	2816
100	100	3000	368	3550

When VM=10

Epoch	Population Size	No. of Task	Global Best (EHO)	Global Best (PSO)
100	100	100	62.8	64
100	100	500	336	300
100	100	1000	639.9	611
100	100	1500	952	902
100	100	2000	1280	1228
100	100	2500	1664	1540
100	100	3000	2040	1714

Findings-
After comparing the dataset by testing it with increasing no. of tasks and by keeping other parameters same we observed that when the resources were less like in VM=5 table EHO gave better results compared to PSO but when tasks were very less(appx.100) PSO was better.
But when the resources were more like in VM=10 table we didn't saw much difference in our results so when resources are more both the algorithms gave good results.
Comparing both the tables we can say using EHO is good.

Comparing Execution Time (Global Best) of Elephant herd optimization VS Particle swarm optimization-
Parameters Used-

- No. of VMs (5,10)
- Epoch
- Population Size
- No. of task
- Global best

When VM=5

Epoch	Population Size	No. of Task	Global Best (EHO)	Global Best (PSO)
100	100	500	192	562
100	200	500	628	524
100	300	500	616	524.57
100	100	1500	992	1838.85
100	200	1500	2040	1747
100	300	1500	2088	1588

When VM=10

Epoch	Population Size	No. of Task	Global Best (EHO)	Global Best (PSO)
100	100	500	336	305.14
100	200	500	308	301
100	300	500	314	297
100	100	1500	952	908.57
100	200	1500	976	920
100	300	1500	937	913

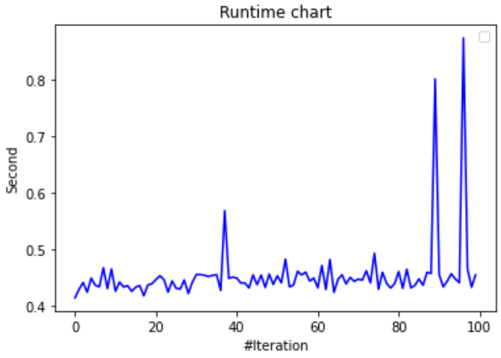
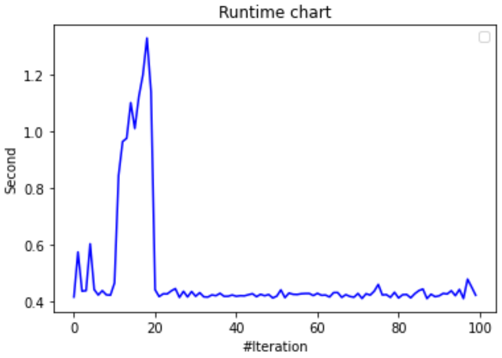
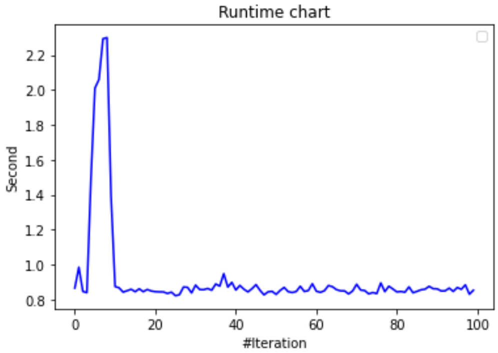
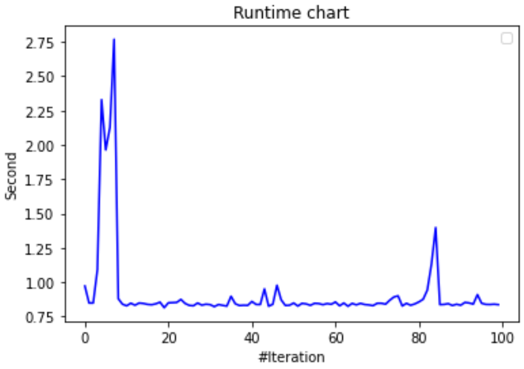
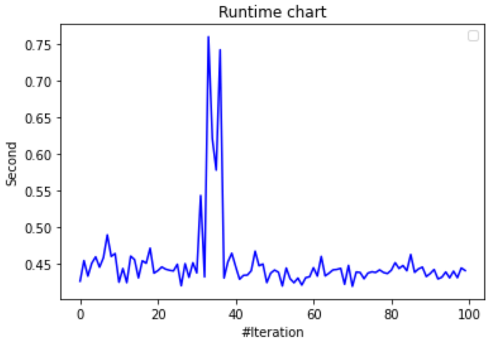
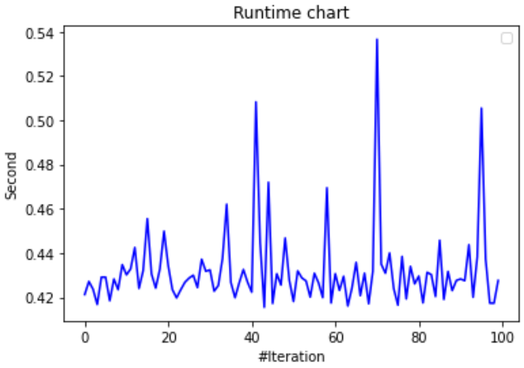
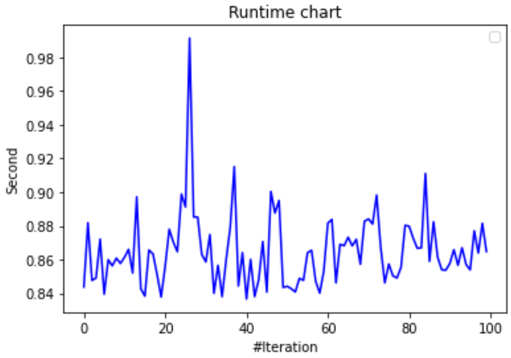
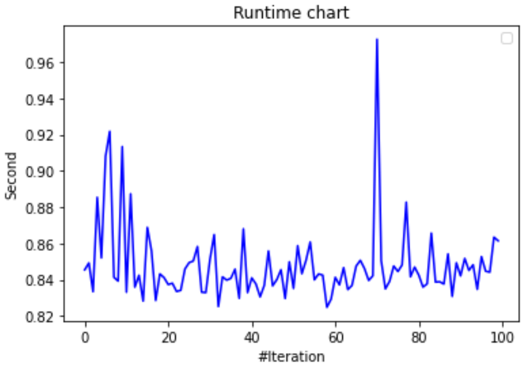
Findings-

When we increase the population size in both the algorithm we didn't saw any major change in the global best. Only when population size was less in 5 VM we saw some differences in EHO.

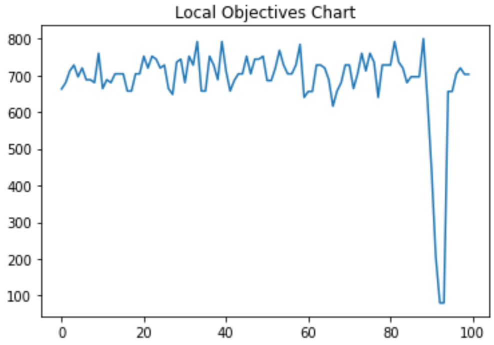
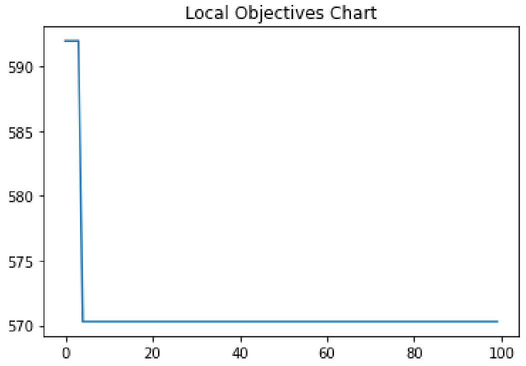
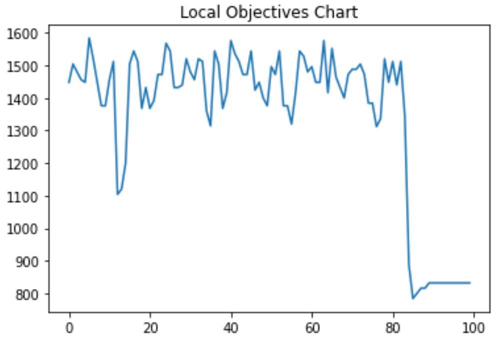
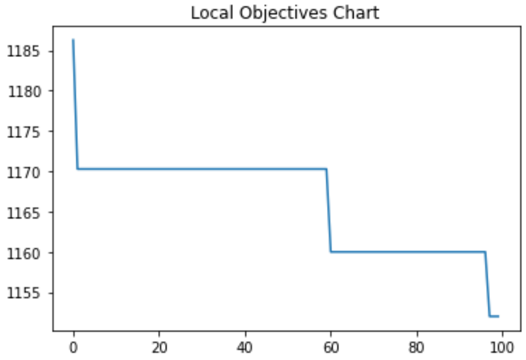
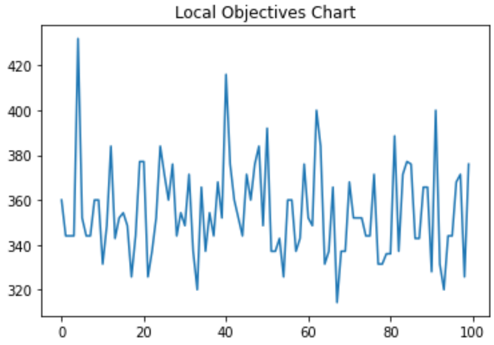
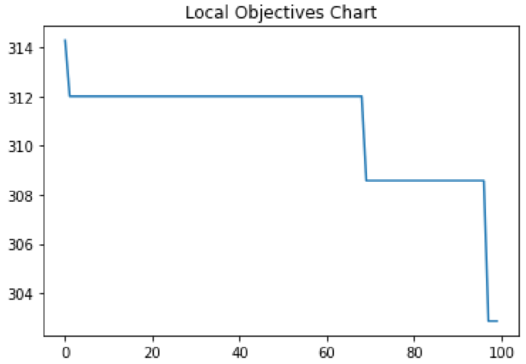
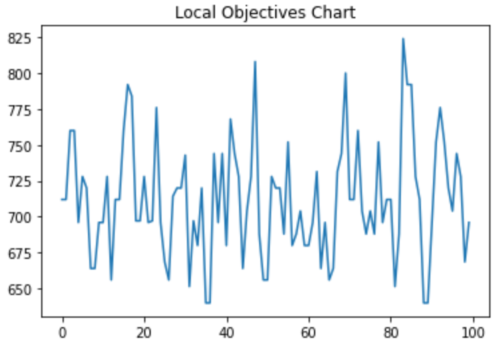
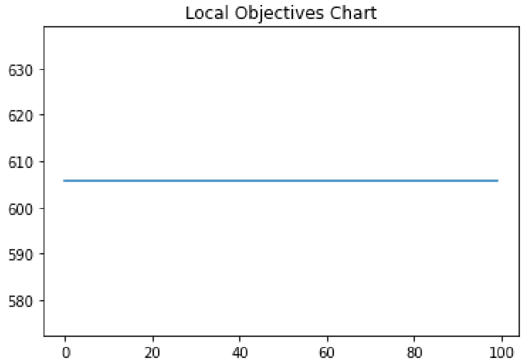
Comparing Elephant herd optimization VS Particle swarm optimization by Visualizing–

S No.	Epoch	Population Size	No. of Task	No. of VM	Global Best (EHO)	Global Best (PSO)
1	100	100	500	5	80	541
2	100	200	1000	5	784	1147
3	100	300	500	10	314.28	297
4	100	100	1000	10	639	634

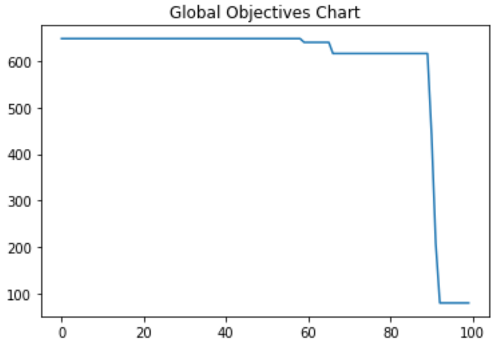
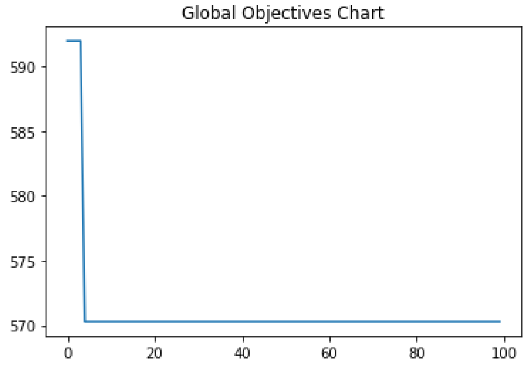
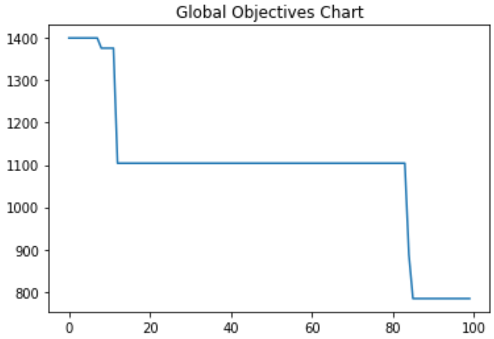
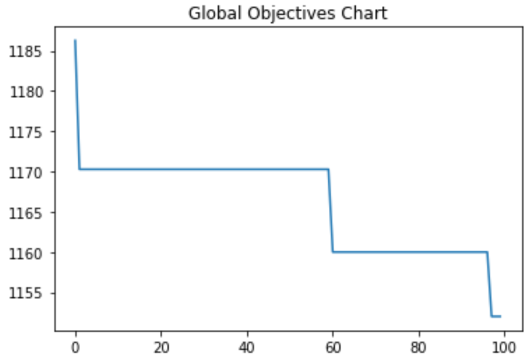
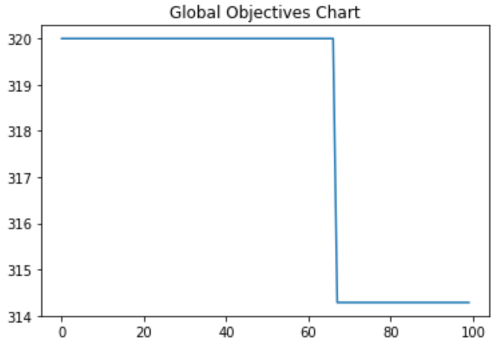
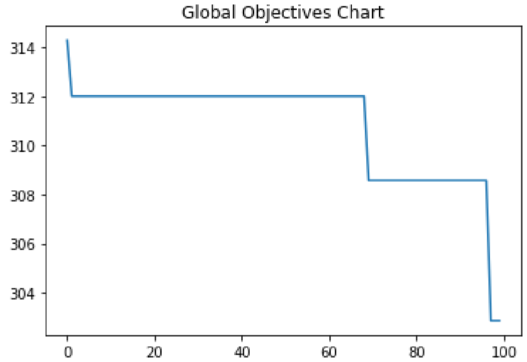
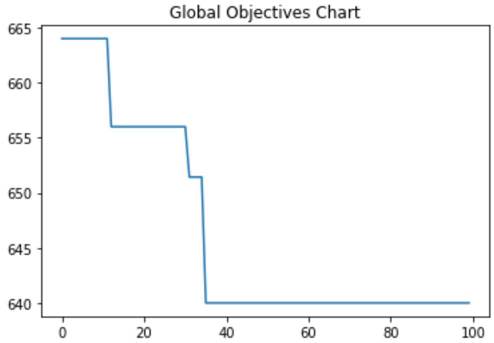
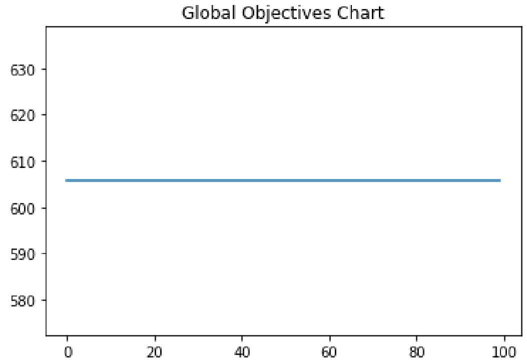
1. Comparing Runtime Per Epoch-

S No.	EHO	PSO
1		
2		
3		
4		

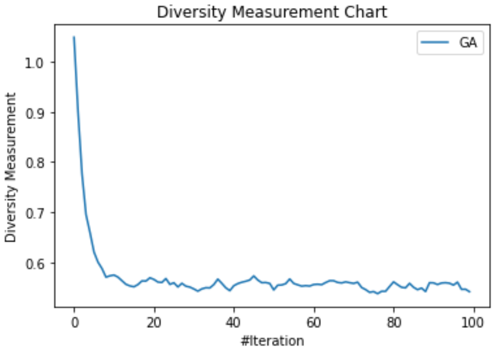
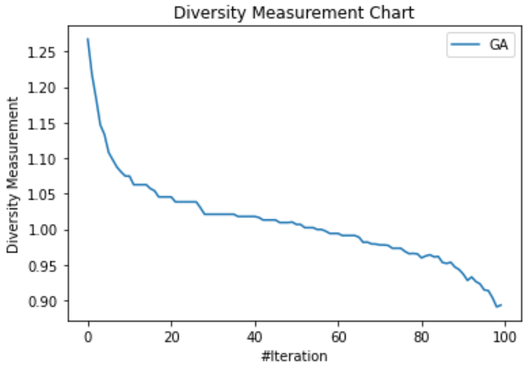
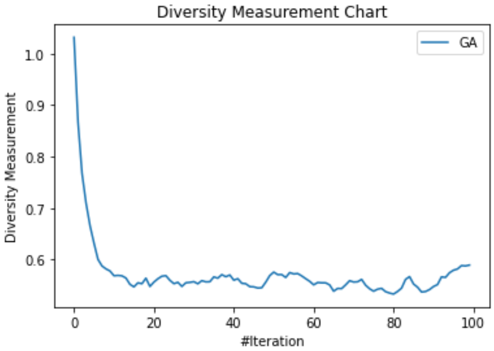
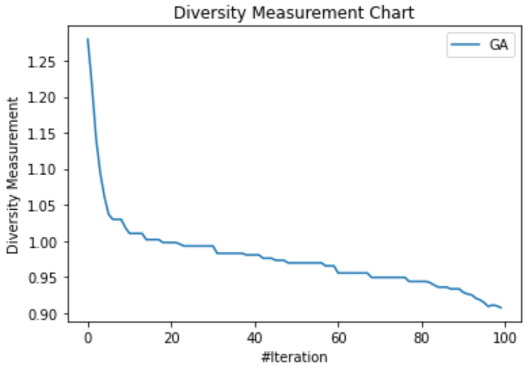
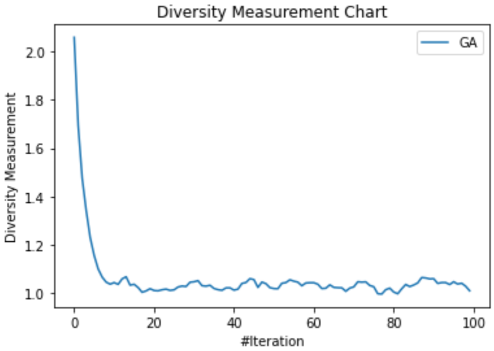
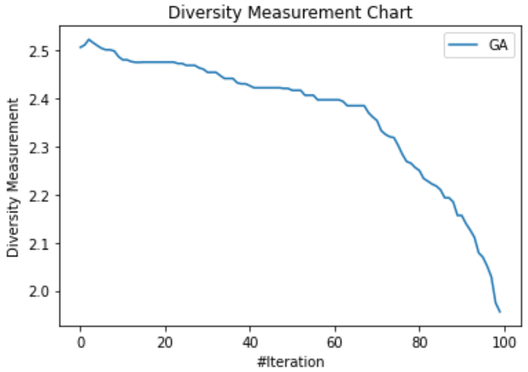
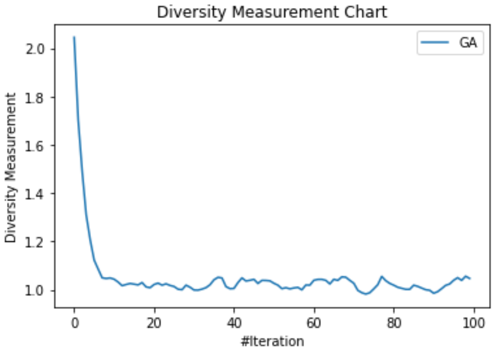
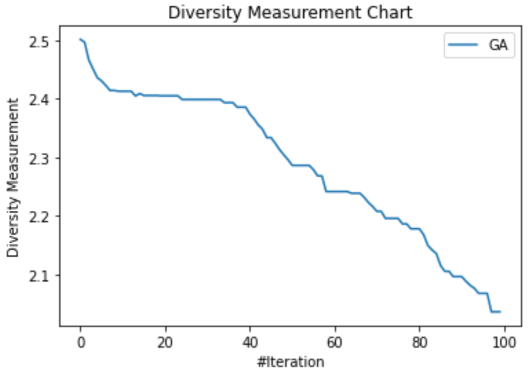
2. Comparing Local Best by Local Objective chart-

S No.	EHO	PSO
1		
2		
3		
4		

3. Comparing Global Best by Global Objective chart-

S No.	EHO	PSO
1		
2		
3		
4		

4. Comparing Diversity Measurement chart-

S No.	EHO	PSO
1		
2		
3		
4		

CONCLUSION

Task Scheduling in cloud is one of the most challenging things. So, we have proposed the Elephant Herd Optimization (EHO) as a method to solve cloud task scheduling problems in cloud. The EHO algorithm is inspired by behavior of elephant herd and its relationship.

In our project we had done various performance analysis of our proposed algorithm EHO and compared it with another nature inspired machine learning algorithm Particle Swarm Optimization (PSO). At first, we had solved the set of tasks ranging from 100 task to 3000 tasks. Then we had solved the by keeping number of tasks constant and varying population size. At last, we had done both of our first task with higher number of virtual machines. The performance of the proposed algorithm to solve the task scheduling problem was compared with other heuristic algorithms for data. It can be concluded that the proposed algorithm had provided better performance than the existing algorithm in many scenarios.

We conclude that EHO has a good characteristic as optimization algorithm and it can be used for solving complex optimizations problems.

In future we will be working on simulating this entire thing on CloudSim simulation environment and will enhance the throughput of cloud system by minimizing the execution time of the assigned with the minimum wastage of cloud resource from the virtual machines.