Multi Objective Optimization Project

Solving the Multi-Objective Problem of IoT Service Placement in Fog Computing Using Cuckoo Search Algorithm

Paper by Chang Liu, Jin Wang, Liang Zhou | Published: 22 January 2022

Work done by:
Prince Nwaekwu, Zied MAALEJ & Rajat Chauhan

Professor: Sonia YASSA
ADEO 2
2022/2023

Contents

Introduction

Due to recent advances in technology, the number of IoT devices has increased significantly.

Cloud data centers are geographically dense and often incur high delays, poor quality of service, and network congestion for distant requests.

Fog Computing (FC) technology emerged to overcome these constraints and effectively meet the requirements of IoT applications.

The purpose of this paper is to optimize the IoT service placement by considering the delay in allocating computing resources, where the heterogeneity of resources and applications is taken into account in the service placement procedure.

Difference between Cloud Computing and Fog Computing

Aspect	Cloud Computing	Fog Computing
Definition	• Centralized computing model provided as a service over the internet	• Decentralized computing model that brings computation and storage closer to end-users
Location	• Large data centers located at some distance from end-users	• Devices at the network edge and servers in closer proximity to end-users
Latency	• Often involves noticeable latency due to communication over the internet	• Designed to reduce latency by moving computation and storage closer to end-users
Scale	• Designed to handle large-scale data processing and storage for large number of users simultaneously	• Designed to support a smaller number of users but can still handle significant amounts of data processing and storage for local applications and services

"In summary, cloud computing is a centralized computing model that provides computing resources as a service over the internet, while fog computing is a decentralized computing model that extends the cloud to the edge of the network, closer to the end users".

The decisional multi objective problem

FSPP is modeled as a multi-objective optimization problem to achieve the best compromise between objectives.

According to the defined objectives, the problem can be expressed as follows:

Find the placement of the IoT service with the minimum of delay, response time, SLA violation, energy consumption, and cost, so that the maximum utilization of fog is achieved. Therefore, based on the six defined objective functions, the mathematical model of the fitness function is defined by Eq.

$$Fitness = min\{d, RT, SLAV, Energy, Cost\}, max\{FU\}$$

The decisional multi objective problem

Delay: The delay metric indicates the time required to execute all application services.

$$d^{A_k} = \sum_{a_l \in A_k} \left(\sum_{f^j \in Res^{a_l}(F^i)} \left[d\left(f^j, a_l\right) x_{a_l}^{f^j} \right] + d\left(F, a_l\right) x_{a_l}^{F} + d\left(N, a_l\right) x_{a_l}^{N} + d\left(R, a_l\right) x_{a_l}^{R} \right)$$

The decisional multi objective problem

SLA Violation: The service request mapping process must be performed without violating the Service Level Agreement (SLA).

Therefore, the SLA should not be violated to ensure meeting the QoS requirements presented in the fog landscape.

It is based on SLA Violation Time per Active Host (SLATAH) and (2) overall Performance Degradation caused by Migration (PDM).

$$SLAV = SLATAH \times PDM$$

The decisional multi objective problem

Energy Consumption: Two types of devices, namely thin and fat, are defined from IoT devices, and the type of fat consumes more energy than the type of thin. According to this assumption, the total energy of a colony is the sum of working energy and idle energy.

$$Energy = W_T + W_F + I_T + I_F$$

Where WT and WF indicate energy consumption of thin and fat type devices in working mode, respectively. Also, IT and IF signify energy consumption of thin and fat type devices in idle mode.

"These objectives include delay, response time, SLA violation, energy consumption, and cost that should be minimized. In addition, the utilization of fog resources is another objective that should be maximized."

How do we solve this problem?

CSA (Cuckoo Search Algorithm)

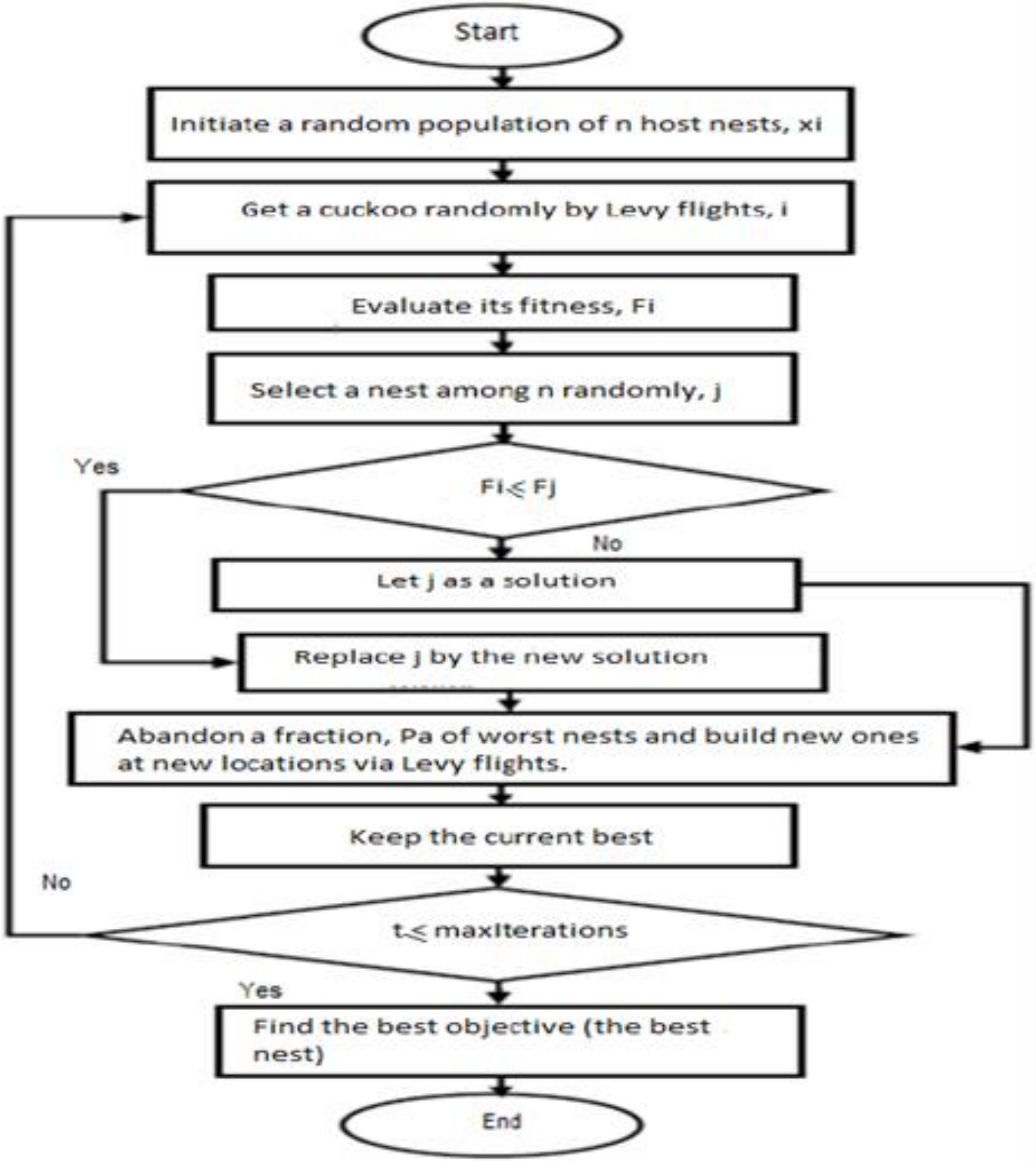
- is a metaheuristic optimization algorithm inspired by the behavior of cuckoo birds in nature proposed as a multi-objective algorithm based on the Pareto archive.

- The algorithm starts with a randomly generated initial population of "cuckoo" solutions, and then iteratively updates their positions based on the fitness of the solutions and the likelihood of discovering better solutions by replacing them with new ones.

- CSA has several parameters that need to be carefully tuned to ensure optimal performance, such as the population size, the step size of the Levy flight, and the rate of replacement of the existing solutions with new ones

Cuckoo Search Algorithm

The main advantage of CSA is allowing the search space to be examined efficiently and providing a suitable qualitative solution in polynomial time.



Evaluation and benchmarking between different metaheuristic algorithm

- In this paper we have seen several metaheuristic algorithms applied on FSPP and we have seen the comparison between them using specific measures such as : convergence, average fog utilization, the delay in executing service

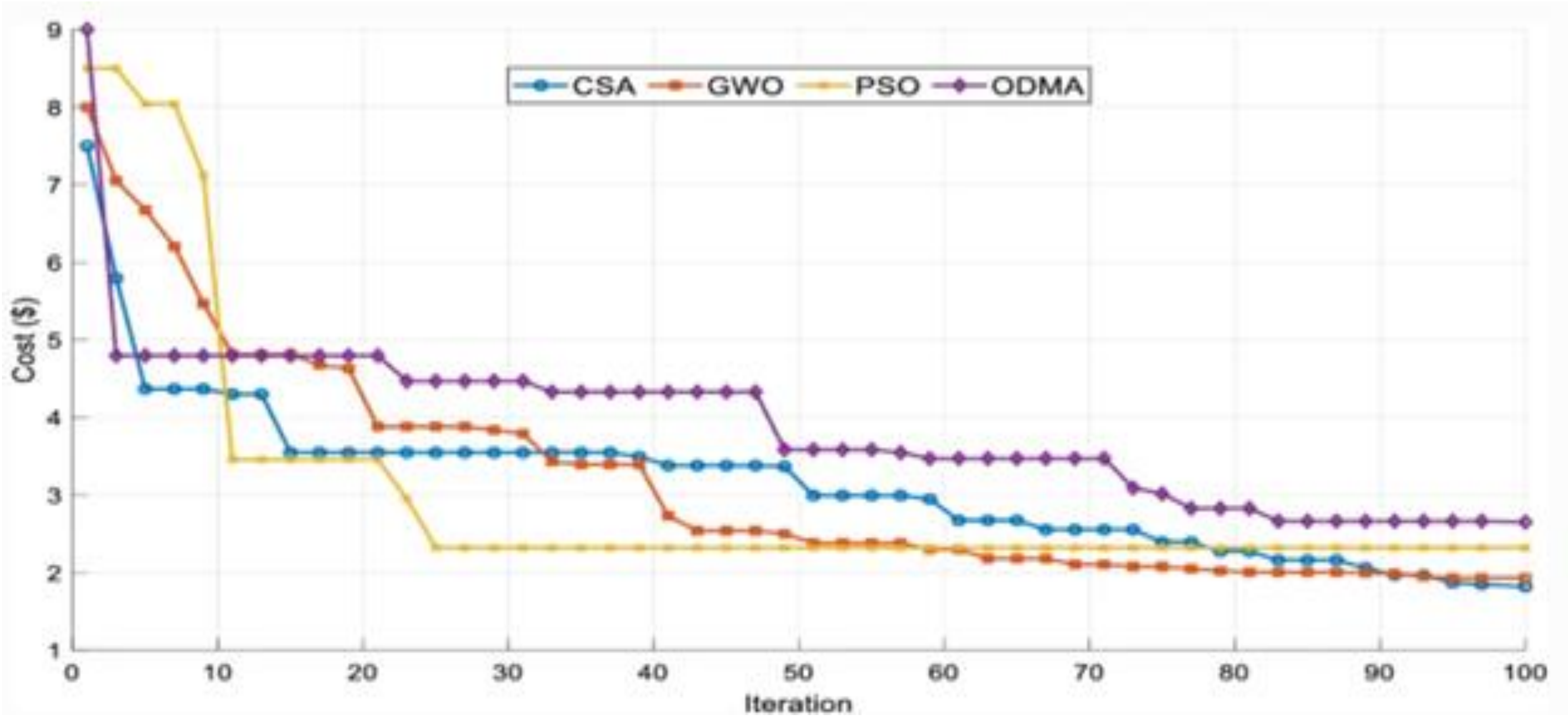


Figure 2 :
Comparison of convergence in different algorithms that are provided for each iteration of the cost of performing all services We listed the Performance of different algorithms based on evaluation metrics

Evaluation and benchmarking between different metaheuristic algorithm

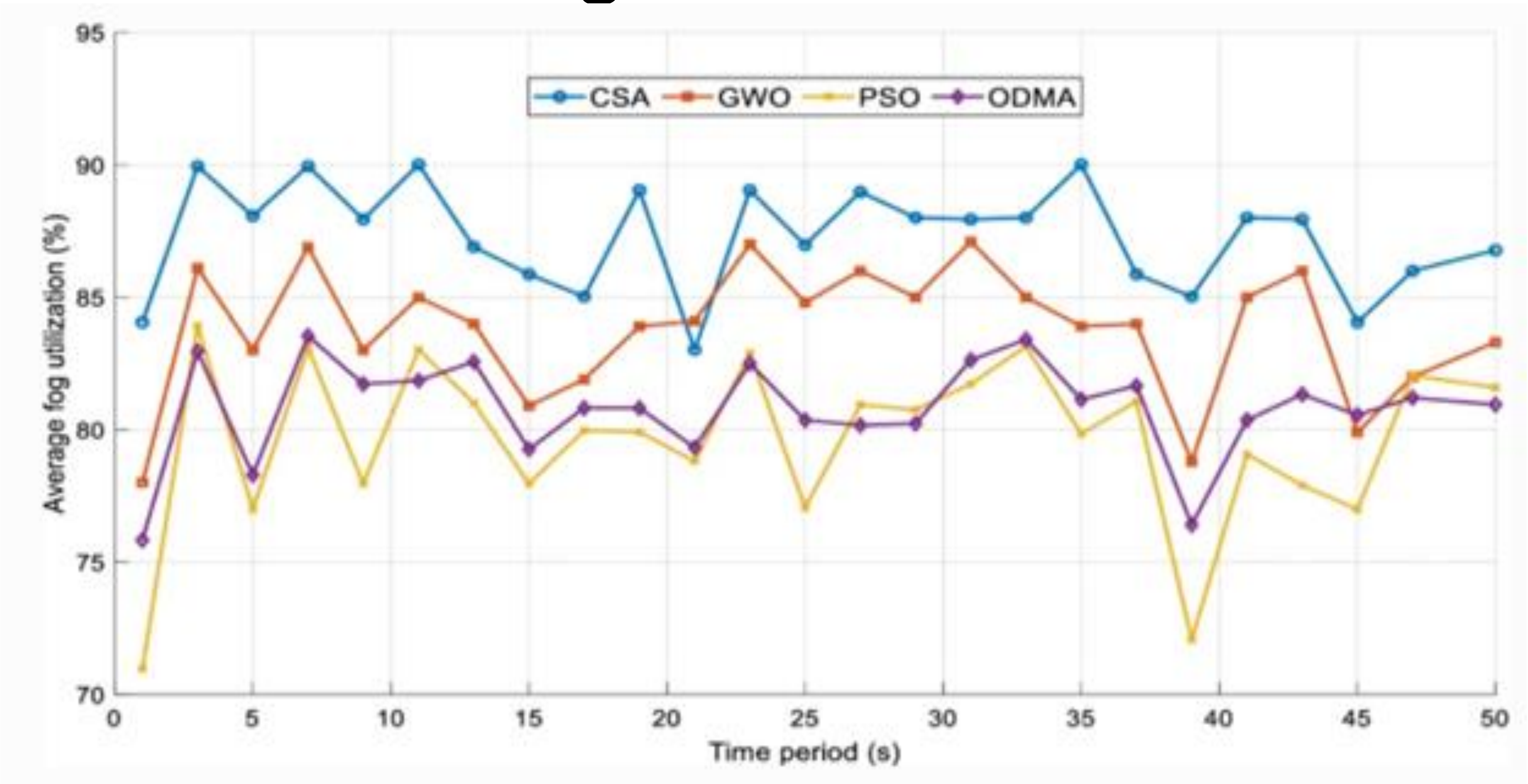


Figure 3 :
Comparison of the average fog utilization in different algorithms, such that for each time frame/period, planning is performed independently.

Evaluation and benchmarking between different metaheuristic algorithm

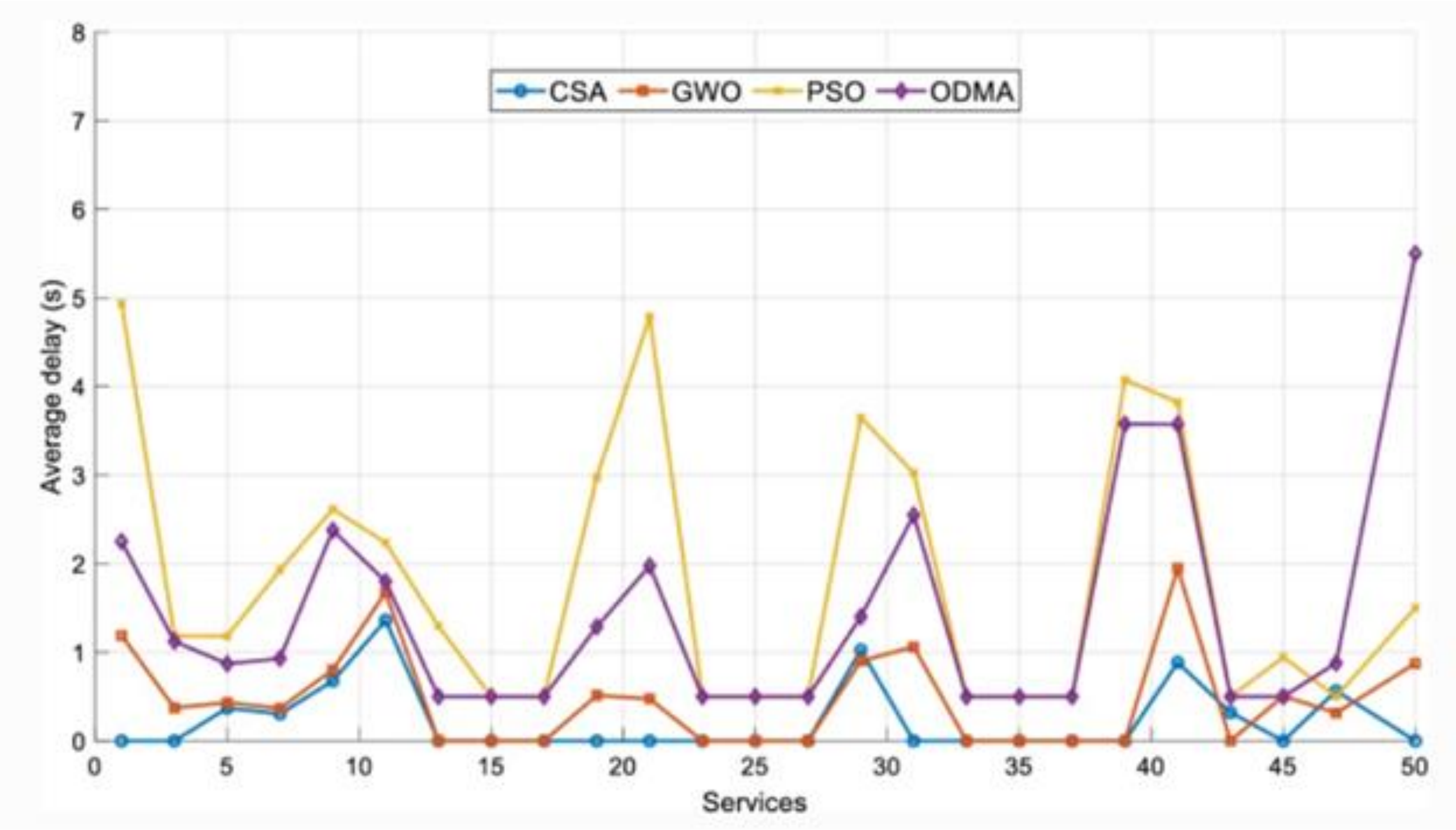


Figure 4:
Comparison of the delay in executing services for different algorithms after the first time period.

Evaluation and benchmarking between different metaheuristic algorithm

Algorithm	Cost ($)	Fog utilization (%)	Delay (s)	Response time (s)	SLAV	Energy consumption (J)	Runtime (s)
CSA	1.82	87.32	0.22	4.49	13	1.87	83.6
GWO	1.93	83.84	0.46	4.63	15	1.90	34.1
PSO	2.32	79.66	1.81	5.07	18	2.03	43.5
ODMA	2.65	80.92	1.42	4.99	17	1.96	98.3

We listed the Performance of different algorithms based on evaluation metrics.

Conclusion

- The results clearly show the superiority of CSA over other algorithms
- After CSA, the results show that GWO, ODMA and PSO algorithms are in the next ranks, respectively
- On average, CSA are 12.34%, 73.26% and 85.97% superior relative to GWO, ODMA and PSO, respectively