Dual-Objective Optimization in Federated Cloud: Profit Equalization and Ecosystem Maximization

Suman Kumar*, Byungkwan Jung*, Kazi Amanul Islam Siddiqui†
*Department of Computer Science, Troy University, Troy, AL 36081, USA
†Department of Computer Science, New Jersey Institute of Technology, Newark, USA
Email: {skumar,bjung}@troy.edu, ks225@njit.edu

***Abstract*—The increasing demand for cloud computing has led to the emergence of cloud federations, where multiple providers collaborate to pool resources and address large-scale demands. Such federations enable smaller providers to remain competitive in a market dominated by larger players. To sustain this cooperation, profit equalization becomes crucial for maintaining trust and long-term collaboration. Unequal profit distribution can lead to dissatisfaction and destabilize the federation. To address this, we formulate a dual-objective optimization problem aimed at maximizing federation profits while ensuring inter-cloud profit equalization. We propose heuristic algorithms, designed for both static and dynamic scenarios, that consider job and provider characteristics to promote equitable profit distribution and efficient resource utilization, solving the optimization problem effectively. Simulation results validate the feasibility of our approach across various cloud computing configurations and scenarios, demonstrating its potential to enhance trust, cooperation, and long term stability within cloud federations.**

***Index Terms*—Profit maximization, Heuristic Algorithms, Cloud Federation, Multi-Objective Optimization, Cloud**

I. Introduction and Motivation

The demand and popularity for cloud computing have led to the rise of numerous providers, resulting in both competition and collaboration. The economies of scale enabled by large providers makes it challenging for smaller cloud providers to compete. To survive and thrive in an increasingly competitive cloud computing landscape, smaller providers must adopt collaborative business models that leverage shared resources to meet growing market demands [1]. This need for cooperation has led to the emergence of cloud federation, where multiple providers join forces to collectively address large-scale demands [2], [3]. By pooling resources and expertise, cloud federation empowers smaller providers to remain competitive in a market dominated by larger players.

The federated cloud infrastructure, governed by common standards and policies [4], offers the potential to improve performance and service quality compared to traditional, non-federated systems. While maximizing profits is a primary objective, ensuring a fair and balanced distribution of profits remains a key research challenge [5]. A significant obstacle arises from the tendency of individual providers to prioritize their own interests, often leading to unequal profit sharing. To address this, researchers have proposed various strategies to promote fairness among participants [6]. However, perceptions of fairness are inherently subjective and can create disagreements, fostering distrust among providers [7]. This issue is exacerbated by uneven workloads across the federation, which can distort profit distribution and fuel dissatisfaction, ultimately risking trust, cooperation, and the long-term sustainability of the federation.

In this paper, we explore a dual-objective of profit maximization while achieving profit equalization within a collaborative federated cloud framework. Our approach draws inspiration from Fehr and Fischbacher's seminal work, "The Nature of Human Altruism" [8], which highlights the critical role of reciprocal behavior and fairness in fostering sustained cooperation. This study demonstrates participants often prefer to cooperate and share profits equally, prioritizing trust and long-term collaboration over short-term gains. Equal profit-sharing mechanisms have been demonstrated to align incentives, and enhance stability, particularly in high-stakes settings such as federated systems [9]. When profits are distributed equitably, collaboration is strengthened over time, fostering trust, group cohesion, and sustainable partnerships. Similarly, in a federated cloud, profit equalization can enhance cooperation by promoting both operational efficiency and financial stability. The dual objective fosters trust, and long-term collaboration.

In this work, we formulate a dual-objective, multi-constraint optimization problem that focuses on profit maximization and profit equalization in cloud federation systems. Heuristics algorithms are simple, efficient, and scalable, offering practical solutions to handle diverse, dynamic conditions. Therefore, we propose a set of such algorithms to effectively address the dual-objective problem for both static and dynamic cloud scenarios. In the static case, job assignments are made once, without further adjustments, addressing both no network cost and network cost situations. In the dynamic case, the system continuously updates and adjusts job assignments based on changing factors such as job arrivals, resource availability and profit distributions, with algorithms developed for both no network cost and network cost scenarios. The simulation results demonstrate the effectiveness of our proposals.

The remainder of this paper is organized as follows: Section II reviews related work, Section III presents the problem formulation and solutions, Section IV shows the performance results, and Section V concludes with future directions.

II. Related Work

The combined job processing capacity of a cloud federation can surpass that of individual clouds, leading to potentially greater overall profits [10]. However, achieving this requires overcoming key financial challenges such as pricing cloud resources across multiple locations, considering dynamic job arrivals, VM configurations, and fluctuating operational costs [11], [12]. In practice, cloud providers tend to prioritize their own profits, complicating the optimization problem and posing risks to long-term federation stability [13]. Various strategies for equitable profit distribution have been explored. For instance, a truthful, budget-balanced auction mechanism for trading VMs, ensuring individual rationality through dynamic resource bidding, is proposed in [14]. However, focusing solely on profit maximization can lead to strategic manipulation leading to uneven profit distribution [15].

A scoreboard mechanism is proposed in [16] to address the free rider problem in cloud federations. However, the presence of selfish cloud providers consumes resources without contributing, similar to the Tragedy of the Commons. A budget-balanced strategy using a reverse auction process is introduced as a multi-objective optimization problem in [17], but selfish clouds can manipulate bids and resource allocation. This challenge is tackled in [18] as a load-balancing problem in the context of an economic and non-cooperative game, aiming for equitable profit maximization within federations.

Previous research has not tackled long-term trust and stability through inter-cloud profit equalization. Additionally, traditional optimization and game-theoretic solutions are inefficient for large federated cloud systems. Our approach uniquely addresses selfish cloud providers with a dual-objective optimization problem and introduces heuristic algorithms that prioritize simplicity, efficiency, and scalability while optimizing both profit equalization and system performance.

III. Cloud Federation Model, Problem Formulation and Algorithms

The federation model consists of Cloud Service Providers (CSPs) [19], [20]. The i^{th} provider has the capacity to run c_i virtual machines. The federation handles n jobs, where the i^{th} job requests r_i virtual machines, generating d_i revenue per machine and requiring p_i network connectivity. Communication costs between providers are denoted by l_{ij}, and a_{ij} represents the allocation of job i to provider j's machines. The provider's profit is determined by revenue d_i minus network costs for inter-provider communication.

A. Optimization Problem Formulation

The federation of clouds is denoted as a graph $G = (V, E)$ where V is the vertex set such that every vertex $v \in V$ is a geo-distributed datacenter (DC) and E is the edge set where each edge has some weight W_{v1v2}. An edge represents the geographic distance between two data centers. Within the federation, the path for the inter-datacenter distribution of jobs forms a minimum spanning tree. The overall profit of the federation at time t, denoted by $P(t)$, is the accumulated sum of the profits of all the providers in the federation. The profit of an individual provider in the federation, denoted by $P_v(t)$, $v = 1, 2, 3...|V|$, is defined as:

$$\begin{aligned} P_v(t) = &\sum_{r\in R}\sum_{j\in J} a_{jvr}(t)k_{vr}(t) - c_v(t)\sum_{h\in H} N_v^h(t) \\ &- \frac{1}{2}\sum_{vj\in V} W_{vivj}\left(\sum_{r\in R} d_{vr}x_{vr}\right) \\ &- \frac{1}{2}N_v\left(\sum_{n\in N}(w_{nvr}\beta_{w_{nvr}})\right) \end{aligned} \tag{1}$$

where:

- W_{vivj} = weight of edge E. Geographic distance from DC v_i to DC v_j, $v_i \neq v_j$, $v_i, v_j \in V$.
- R = set of all job types.
- J = set of all customer types.
- $a_{jvr}(t)$ = number of type r jobs from zone j at t in v.
- $k_{vr}(t)$ = price charged to type-r job at time t in v.
- $c_v(t)$ = power cost of running one server at DC v.
- $N_v^h(t)$ = number of active servers of type-h in v at t.
- N_v = total number of servers at $v \in V$.
- H = number of VM types.
- d_{vr} = cost of one job r per unit distance for DC v.
- x_{vr} = number of r-jobs shared by DC v.
- w_{nvr} = number of VMs shared by PM n in DC v.
- $\beta_{w_{nvr}}$ = price of unit job by VM w in PM n of DC v.

Objective functions maximizing the federation's profit and minimizing cloud providers' profit standard deviation are:

$$\text{Max} : P(t) = \sum_{v\in V} P_v(t) \tag{2}$$

$$\text{Min} : \sum_{v\in V} P_v(t)^2 - \left(\sum_{v\in V} P_v(t)\right)^2 \tag{3}$$

subject to:

$$P_v(t) > 0 \quad \forall v \in \{1, 2, 3, ...|V|\}$$

$$N_v^h(t) \in \mathbb{Z}^+ \cup 0, \quad \forall h \in H, \forall v \in V, t \in [1, T]$$

$$W_{vivj} \geq 0, \quad v_i \neq v_j, v_i, v_j \in V$$

$$\sum_{n\in N_v} w_{nvr} + x_{vr} \leq \sum_{j\in J} a_{jvr}(t)$$

B. Heuristics Algorithms

The solution to the optimization problem discussed in the previous subsection is computationally expensive when using brute force. To address this, we propose an algorithm inspired by sorting-based placement strategies [21]. We introduce different heuristic approaches tailored for both static and dynamic scenarios, considering negligible (no) network cost and significant network cost variants for each. Negligible or no network cost is applicable for federated cloud located nearby each other and other one with long distance separated, therefore, incurring significant costs in terms of delays and data transfers. Next two subsections describe proposed heuristic algorithms providing efficient solutions.

1) Case 1: Static Allocation: In static scenarios, the job assignments are made once based on initial conditions, without further changes or adjustments. These algorithms assume that the environment remains constant after the initial allocation, and no subsequent reallocation or dynamic updates occur. Static approaches focus on maximizing and equlizing profit based on the information available at the time of assignment.

Algorithm 1: Static VM Placement, No Network Cost

Input: Set of jobs $J = \{J_1, J_2, \ldots, J_n\}$, each with VM requirements r_i;
Set of cloud providers $V = \{V_1, V_2, \ldots, V_m\}$, each with capacity c_j and initial profit π_j
Output: Final assignment of jobs to providers and updated profits π_j

while *there exists an unassigned job* J_i **do**
  Select job J_i with largest r_i;
  Find provider P_j with the smallest profit π_j such that $c_j \geq r_i$;
  if $c_j \geq r_i$ **then**
    Assign entire job J_i to provider V_j;
    Update $c_j \leftarrow c_j - r_i$ and $\pi_j \leftarrow \pi_j + \text{profit}(J_i)$;
  else
    Assign as many VMs from J_i as possible to V_j, i.e., $x = \min(r_i, c_j)$;
    Update $r_i \leftarrow r_i - x$, $c_j \leftarrow 0$, and $\pi_j \leftarrow \pi_j + \text{profit}(x)$;

Result: Final assignment of jobs to providers and updated profits π_j

a) Negligible network cost: Algorithm 1 sorts jobs in *decreasing order* based on VM requirements (r_i), handling larger jobs first to optimize resource allocation. By doing so, it prevents smaller jobs from limiting later assignments when capacity becomes scarce. Jobs are assigned to the provider with the smallest profit who can handle the job. This helps balance profits across providers. If a job can't be fully handled, it assigns as many VMs as possible and adjusts the remaining job size. Partial assignments prevent resource wastage and ensure fairness. To save computation, the algorithm uses only the $a_{ij}d_i$ part of the profit formula and avoids full job re-sorting after partial assignment.

b) Significant Network costs: Algorithm 2 sorts jobs based on profit per job rather than resource requirements, and subtracts a portion of the network cost from the revenue without tracking every provider involved in a job. The rationale is that sorting by profit prioritizes higher-yield jobs, potentially increasing overall revenue more efficiently. This approach simplifies profit calculations by reducing the complexity of network cost tracking.

2) Case 2: Dynamic Reallocation: In dynamic scenarios, the system continuously updates and adjusts job assignments based on changing conditions, such as resource availability, network performance, or job requirements. These algorithms involve reallocation strategies that react to evolving circumstances, aiming to optimize resource use or minimize costs over time. Dynamic approaches are suited for environments where job demands and provider capacities fluctuate.

Algorithm 2: Static VM Placement, Network Cost

Input: Set of jobs $J = \{J_1, J_2, \ldots, J_n\}$, each with VM requirements r_i;
Set of cloud providers $V = \{V_1, V_2, \ldots, V_m\}$, each with capacity c_j and initial profit π_j
Output: Final assignment of jobs and updated profits

while *there exists an unassigned job* J_i **do**
  Select job J_i with the largest π_i;
  Find provider V_j with the smallest profit π_j such that $c_j \geq r_i$;
  if $c_j \geq r_i$ **then**
    Assign entire job J_i to provider V_j;
    Update $c_j \leftarrow c_j - r_i$ and $\pi_j \leftarrow \pi_j + \text{profit}(J_i) - \text{network\_cost}(J_i, V_j)$;
  else
    Assign as many VMs from J_i as possible to P_j, i.e., $x = \min(r_i, c_j)$;
    Update $r_i \leftarrow r_i - x$, $c_j \leftarrow 0$, and $\pi_j \leftarrow \pi_j + \text{profit}(x) - \text{network\_cost}(x, V_j)$;

Result: Final assignment of jobs and updated profits

Algorithm 3: Reallocation of VMs, No Network cost

Input: Same as Algorithm 1
Output: Final assignment of jobs and updated profits after reallocation

Perform initial assignment using Algorithm 1;
foreach *provider* P_j **do**
  foreach *job* J_i *assigned to* V_j **do**
    if *job* J_i *size greater than* V_j*, reallocating* J_i *or part of* J_i *to another provider* V_k *improves* π_j **then**
      Reallocate VMs of J_i from V_j to V_k;
      Update c_j, c_k, and profits π_j, π_k;

Result: Final assignment and updated profits after reallocation

a) Negligible network cost: Algorithm 3 builds on the standard job allocation process by attempting to reallocate VMs between providers after the initial placement to improve performance. It re-evaluates each job to determine if a better configuration is possible. The rationale is that the initial placement may not be optimal due to localized decisions, and reallocation can help balance workloads and enhance profits. The migration cost of job is negligible. This approach should show how a second pass through the job assignments leads to improved system performance, particularly in terms of fairness and resource utilization.

Algorithm 4: Reallocation of VMs, Network cost

Input: Same as Algorithm 1
Output: Final assignment and updated profits after reallocation

Perform initial assignment using Algorithm 1;
for *each provider* V_j **do**
  for *each job* J_i *assigned to* V_j **do**
    if *job* J_i *size greater than* V_j*, reallocating based on different metric (e.g., cost, latency) improves* π_j **then**
      Reallocate part of J_i to provider V_k;
      Update c_j, c_k, and profits π_j, π_k;

Result: Final assignment and updated profits after reallocation

b) Significant Network cost: Algorightm 4 introduces a different VM reallocation technique, focusing on metrics like cost minimization or latency reduction instead of profit or resource utilization. The rationale is that reallocating based on real-world constraints, such as network cost or latency, can enhance overall system performance. This approach determines whether alternative reallocation methods lead to more efficient resource use and improved service quality for users.

IV. Simulation Result and Discussion

We evaluate the performance of the proposed heuristic algorithms in terms of profit in dollars as earning based on how much is assigned for each task, standard deviation in dollars, and computational complexity as in time complexity, following methods outlined in [22]. This section describes simulation setup and presents results, and discussion.

A. Simulation Set-up

We consider two environments: Static Cloud-Federation and Dynamic Cloud-Federation. In static environments, job allocation is done once without further adjustments, reflecting scenarios where resource conditions remain stable. Dynamic environments, on the other hand, continuously update job assignments as cloud conditions change, such as in resource availability or network performance. For each case, we evaluate two scenarios: no network cost (representing local clouds federations with little to no communication overhead), and with network cost (applicable to long distance geographically distributed clouds). In homogeneous environments, all cloud service providers within the federation have equal capacity (20 clouds forming federation each cloud has 50 VM), simplifying job allocation since all clouds can handle similar workloads. In heterogeneous environments, clouds have diverse capacities, reflecting a more complex, realistic scenario where the federation consists of clouds with varying resources. Heterogeneous cloud capacities and job capacities are modeled using the Zipf distribution. This distribution realistically reflects varying task sizes, revenues, and connectivity costs in cloud interactions. The Zipf distribution's applicability to cloud environments is well explored and experimentally validated in prior studies [23]–[25]. We ran 1,000 iterations per algorithm to ensure statistical significance, capturing randomness in job demands and revenues to simulate real cloud federation complexity.

B. Simulation Result

1) Static Cloud-Federation: Both algorithms Algorithm 2. Both methods begin by sorting jobs and assigning them to clouds with enough capacity but the lowest profit, continuing until the cloud can no longer meet the job requirements.

Figures 1(a) and 1(d) show the profits of federated clouds using both algorithms. In Figure 1(a), as more homogeneous clouds join, both methods see increased profits. Algorithm 1 shows a sharp rise initially but stabilizes as more clouds join, while Algorithm 2 shows steady, continuous growth. Algorithm 1 yields higher profits overall, as it does not account for network costs. In the heterogeneous case (Figure 1(d)), profits are initially lower but increase with the number of clouds. Similar to the homogeneous case, Algorithm 2 generates lower profits due to network latency, a key factor in task reallocation.

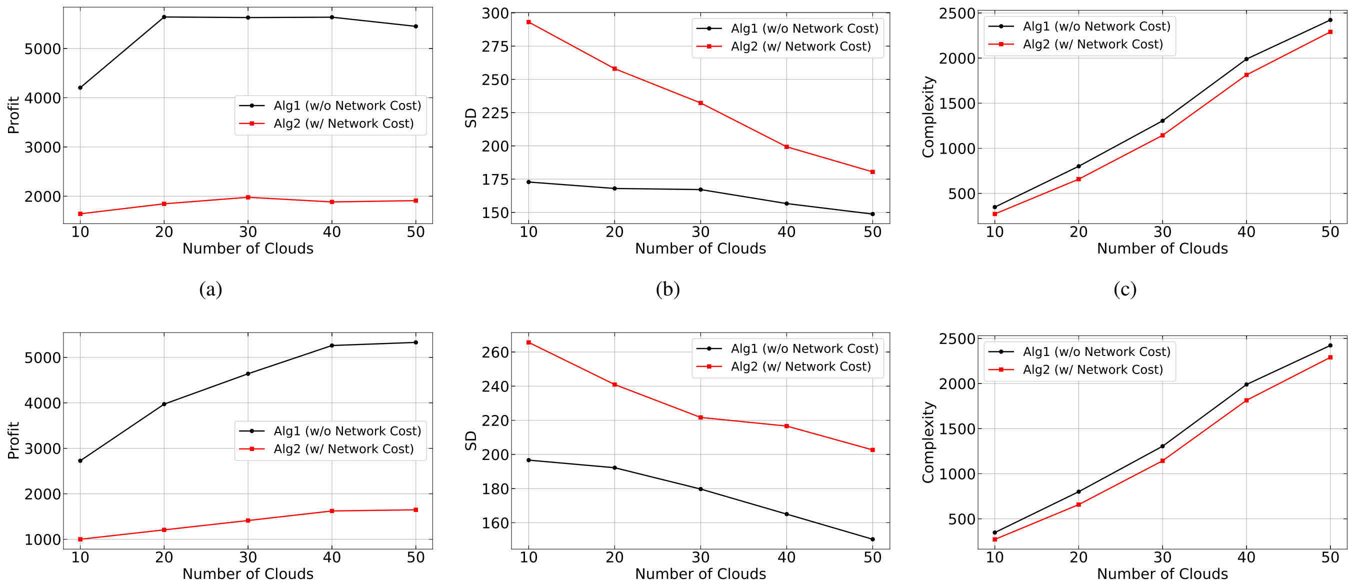
Figures 1(b) and 1(e) depict the standard deviation (SD) of profits. Both methods reduce SD as clouds join, with Algorithm 2 showing a more significant drop due to the inclusion of network costs. In Figure 1(b), Algorithm 1 reduces SD by 14%, while Algorithm 2 cuts it by 38%. In the heterogeneous case (Figure 1(e)), both algorithms reduce SD by 23%.

Finally, Figures 1(c) and 1(f) show the computational complexity of both methods, which increases steadily as more clouds participate. The complexity increase is similar for both algorithms, with no significant differences.

2) Dynamic Cloud-Federation: Algorithms 3 and 4 focus on re-evaluating jobs before allocation. A job can be assigned to multiple clouds when the initially allocated cloud lacks sufficient capacity.

Figures 2(a) and 2(d) show the profits of the federated cloud using Algorithms 3 and 4. In Figure 2(a), both algorithms exhibit stable profits as more homogeneous clouds join the federation, with no significant changes. In heterogeneous environments, Figure 2(d) shows an overall increase in profits, as the diversity in cloud sizes allows for more flexible job allocation, enhancing overall profit. The difference between the two algorithms lies in their use of additional metrics like network cost and latency. In homogeneous environments, profits increase gradually as jobs are allocated to clouds with available capacity. However, in heterogeneous environments, the wider variety of cloud capacities allows for better distribution, improving profits across all clouds.

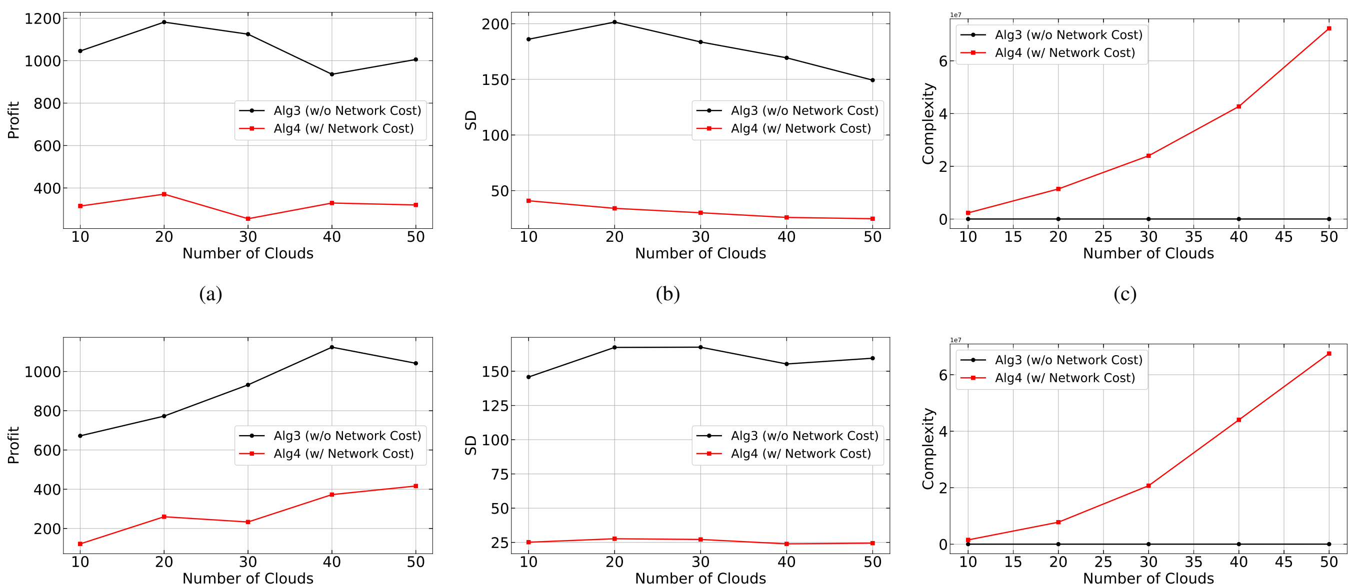
Figures 2(b) and 2(e) display the standard deviation (SD) of profits. In Figure 2(b), Algorithm 3 reduces SD by 19%, while Algorithm 4 shows a more significant 40% decrease. Algorithm 4 starts with a relatively balanced profit distribution, and as more clouds join, the distribution of jobs further reduces SD. In the heterogeneous environment (Figure 2(e)), SD shows



(a) (b) (c)

(d) (e) (f)

Fig. 1: Static Cloud-Federation Performance: Homogeneous (Top Row) and Heterogeneous (Bottom Row) Cloud



(a) (b) (c)

(d) (e) (f)

Fig. 2: Dynamic Cloud-Federation Performance: Homogeneous (Top Row) and Heterogeneous (Bottom Row) Cloud

a slight increase, indicating that profit equality among clouds is maintained as the federation grows.

Figures 2(c) and 2(f) highlight the computational complexity of the dynamic algorithms. In both environments, Algorithm 4's complexity increases significantly as new clouds join the federation. This is due to the need to account for additional factors, such as network cost and latency, during job allocation, which leads to higher computational demands compared to Algorithm 3.

C. Discussion

In both static and dynamic environments, algorithms without network costs (Algorithm 1 and Algorithm 3) outperform their network-cost counterparts (Algorithm 2 and Algorithm 4) in terms of profit. However, Algorithm 2 and Algorithm 4 achieve better profit equalization, shown by lower standard deviation (SD), balancing fairness and profitability. This trade-off demonstrates that these heuristic algorithms effectively promote profit equalization without drastically sacrificing total profit, ensuring fair distribution in federated clouds. Homogeneous cloud environments prove less attractive for profit maximization compared to heterogeneous environments, where adding diverse CSPs increases overall profit and maintains profit equality. Factors like network cost and latency are crucial considerations in resource allocation.

The heuristic algorithms presented in this context are reasonably effective in achieving profit equalization in cloud federations without significantly sacrificing overall profits. The results demonstrate that while the total profit may be lower in some algorithms, such as those accounting for network costs (e.g., Algorithm 2 and Algorithm 4), they achieve better balance and fairness in profit distribution. This is crucial for the long-term sustainability of cloud federations. Given that the underlying optimization problem is NP-hard, achieving an optimal solution is computationally infeasible. These heuristic algorithms offer a feasible alternative by balancing profit maximization and equalization. The algorithms, while not perfect, achieve a satisfactory level of equalization, making them a practical solution for federated cloud environments.

V. Conclusion

In this paper, we propose a dual-objective optimization problem that aims to maximize federation profits while equalizing inter-cloud profits in federated cloud systems. To achieve this, we introduce a set of heuristic algorithms tailored for both static and dynamic scenarios. These algorithms address resource allocation challenges under varying conditions, accounting for factors such as network costs, job requirements, and cloud capacities. Our simulations in both homogeneous and heterogeneous cloud environments demonstrate that these methods not only enhance overall federation profits but also ensure a balanced distribution of profits across clouds, promoting long-term collaboration and system stability. The results confirm the feasibility of profit equalization through heuristic approaches, offering a practical balance between computational efficiency and fairness, making them viable for cloud federation deployment. Future work could explore more advanced dynamic scenarios and real-time cloud traffic for improved scalability and applicability. In future work, comparison of proposed optimization methods and conventional profit maximization and more in depth discussion of the complexity analysis will be conducted.

References

[1] J. P. Romero Coronado and J. Altmann, "Model for incentivizing cloud service federation," in *Economics of Grids, Clouds, Systems, and Services: 14th International Conference, GECON 2017, Biarritz, France, September 19-21, 2017, Proceedings 14*. Springer, 2017, pp. 233–246.

[2] A. Hammoud, A. Mourad, H. Otrok, O. A. Wahab, and H. Harmanani, "Cloud federation formation using genetic and evolutionary game theoretical models," *Future Generation Computer Systems*, vol. 104, pp. 92–104, 2020.

[3] A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica, "Above the clouds: A berkeley view of cloud computing," *Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS*, vol. 28, no. 13, p. 2009, 2009.

[4] R. Moreno-Vozmediano, R. S. Montero, and I. M. Llorente, "Iaas cloud architecture: From virtualized datacenters to federated cloud infrastructures," *Computer*, no. 12, pp. 65–72, 2012.

[5] P. Cong, G. Xu, T. Wei, and K. Li, "A survey of profit optimization techniques for cloud providers," *ACM Computing Surveys (CSUR)*, vol. 53, no. 2, pp. 1–35, 2020.

[6] A. Badshah, A. Ghani, A. Daud, A. T. Chronopoulos, and A. Jalal, "Revenue maximization approaches in iaas clouds: Research challenges and opportunities," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 7, p. e4492, 2022.

[7] A. Angelovski, J. Brandts, and C. Solà, "Equal and unequal profit sharing in highly interdependent work groups: A laboratory experiment," *Journal of Economic Behavior & Organization*, vol. 184, pp. 232–252, 2021.

[8] E. Fehr and U. Fischbacher, "The nature of human altruism," *Nature*, vol. 425, no. 6960, pp. 785–791, 2003.

[9] C. Weber, H. Haugh, M. Göbel, and H. Leonardy, "Pathways to lasting cross-sector social collaboration: A configurational study," *Journal of Business Ethics*, vol. 177, no. 3, pp. 613–639, 2022.

[10] N. Mansouri and M. M. Javidi, "Cost-based job scheduling strategy in cloud computing environments," *Distributed and Parallel Databases*, vol. 38, no. 2, pp. 365–400, 2020.

[11] A. N. Toosi, R. N. Calheiros, R. K. Thulasiram, and R. Buyya, "Resource provisioning policies to increase iaas provider's profit in a federated cloud environment," in *High Performance Computing and Communications (HPCC), 2011 IEEE 13th International Conference on*. IEEE, 2011, pp. 279–287.

[12] C. Li, Q. Cai, and Y. Lou, "Optimal data placement strategy considering capacity limitation and load balancing in geographically distributed cloud," *Future Generation Computer Systems*, vol. 127, pp. 142–159, 2022.

[13] F. Ramezani, S. Abrishami, and M. Feizi, "A market-based framework for resource management in cloud federation," *Journal of Grid Computing*, vol. 21, no. 1, p. 3, 2023.

[14] R. Lu, Y. Liang, Q. Ling, C. Li, and W. Wu, "Double auction and profit maximization mechanism for jobs with heterogeneous durations in cloud federations," *Journal of Cloud Computing*, vol. 10, no. 1, p. 34, 2021.

[15] L. Mashayekhy, M. M. Nejad, and D. Grosu, "A trust-aware mechanism for cloud federation formation," *IEEE Transactions on Cloud Computing*, vol. 9, no. 4, pp. 1278–1292, 2019.

[16] M. R. M. Assis and L. F. Bittencourt, "Avoiding free riders in the cloud federation highways," in *International Conference on Cloud Computing and Services Science*, vol. 2. SCITEPRESS, 2017, pp. 197–207.

[17] G. Darzanos, T. G. Papaioannou, and G. D. Stamoulis, "Efficient and budget-balanced decentralized management of federated cloud and edge providers," in *2024 IEEE 24th International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*. IEEE, 2024, pp. 374–384.

[18] S. Mondal, G. Das, and E. Wong, "An economic and non-cooperative load-balancing framework among federated cloudlets," *Computer Networks*, vol. 207, p. 108847, 2022.

[19] Í. Goiri, J. Guitart, and J. Torres, "Economic model of a cloud provider operating in a federated cloud," *Information Systems Frontiers*, vol. 14, no. 4, pp. 827–843, 2012.

[20] M. Alicherry and T. Lakshman, "Network aware resource allocation in distributed clouds," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 963–971.

[21] X. Li, J. Wu, S. Tang, and S. Lu, "Let's stay together: Towards traffic aware virtual machine placement in data centers," in *INFOCOM, 2014 Proceedings IEEE*. IEEE, 2014, pp. 1842–1850.

[22] J. Zhao, H. Li, C. Wu, Z. Li, Z. Zhang, and F. Lau, "Dynamic pricing and profit maximization for the cloud with geo-distributed data centers," in *INFOCOM, 2014 Proceedings IEEE*. IEEE, 2014, pp. 118–126.

[23] R. Zhang and R. Zhou, "An efficient online auction for placing and pricing cloud container clusters," *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 4, pp. 2168–2179, 2023.

[24] M. Su, G. Wang, and K.-K. R. Choo, "Prediction-based resource deployment and task scheduling in edge-cloud collaborative computing," *Wireless Communications and Mobile Computing*, vol. 2022, no. 1, p. 2568503, 2022.

[25] Z. Wu, Y. Deng, H. Feng, Y. Zhou, G. Min, and Z. Zhang, "Blender: A container placement strategy by leveraging zipf-like distribution within containerized data centers," *IEEE Transactions on Network and Service Management*, vol. 19, no. 2, pp. 1382–1398, 2021.