# Towards Graph-based Cloud Cost Modelling and Optimisation

Akif Quddus Khan*, Nikolay Nikolov†, Mihhail Matskin‡, Radu Prodan§, Christoph Bussler¶, Dumitru Roman†, and Ahmet Soylu‖

*Norwegian University of Science and Technology, Norway
akif.q.khan@ntnu.no
†SINTEF Digital, Norway
‡ KTH Royal Institute of Technology, Sweden
§ University of Klagenfurt, Austria
¶ Robert Bosch LLC, CA, USA
‖ OsloMet – Oslo Metropolitan University, Norway

***Abstract*—Cloud computing has become an increasingly popular choice for businesses and individuals due to its flexibility, scalability, and convenience; however, the rising cost of cloud resources has become a significant concern for many. The pay-per-use model used in cloud computing means that costs can accumulate quickly, and the lack of visibility and control can result in unexpected expenses. The cost structure becomes even more complicated when dealing with hybrid or multi-cloud environments. For businesses, the cost of cloud computing can be a significant portion of their IT budget, and any savings can lead to better financial stability and competitiveness. In this respect, it is essential to manage cloud costs effectively. This requires a deep understanding of current resource utilization, forecasting future needs, and optimising resource utilization to control costs. To address this challenge, new tools and techniques are being developed to provide more visibility and control over cloud computing costs. In this respect, this paper explores a graph-based solution for modelling cost elements and cloud resources and potential ways to solve the resulting constraint problem of cost optimisation. We primarily consider utilization, cost, performance, and availability in this context. Such an approach will eventually help organizations make informed decisions about cloud resource placement and manage the costs of software applications and data workflows deployed in single, hybrid, or multi-cloud environments.**

***Index Terms*—cloud, cost, optimisation, graph**

## I. Introduction

Cloud computing has experienced exponential growth in recent years, and organizations have increasingly embraced cloud services to meet their computing needs. According to Gartner, most enterprises, 85%, are expected to adopt a cloud-first approach by 2025 [1]. The advantages of cloud computing, including scalability and flexibility, are well known, but the challenge of managing the costs associated with cloud computing remains a significant concern. As organizations continue to move a greater portion of their computing workloads to the cloud, cost optimisation becomes even more critical. The cost of cloud resources can rapidly accumulate, and organizations must proactively manage cloud costs to avoid unexpected and potentially costly expenses. The cost structure becomes even more complicated when dealing with hybrid or multi-cloud environments [2]. Managing cloud costs requires understanding resource utilization and the trade-offs between cost, performance, and availability. While challenging, cost optimisation can free up financial resources for other tasks and improve overall business performance.

The rapid growth of cloud computing has driven significant investment in research and development in the field of cloud cost optimisation [3]. Despite this investment, there is still a need for practical yet effective solutions to help organizations manage their cloud computing costs more effectively. In this respect, this paper explores a solution for modelling cost elements and cloud resources in the form of a graph and potential ways to solve the resulting constraint problem of cost optimisation. This presents the opportunity to consider a range of factors, including utilization, cost, performance, and availability. The significance of such an approach lies in its potential to help organizations make informed decisions about cloud resource placement and provide them with a solution for managing their cloud computing costs effectively for a wide range of software applications and data workflows [4]–[6] in single, hybrid, or multi-cloud environments.

The rest of the paper is organized as follows. Section II describes the elements of the cloud computing cost, while Section III presents the related work. Section IV describes cloud cost optimisation approaches in general, while Section V explores a preliminary solution. Finally, Section VI provides a discussion and concludes the paper.

## II. Cloud Computing Cost

Cloud computing is a model for delivering on-demand computing resources over the internet. Its cost can be divided into three main categories: compute, data transfer, and storage. Compute costs include the cost of virtual machines, containers, serverless functions, etc. Data transfer costs include transferring data within the cloud service providers' (CSP) network and to/from an external network. The cost structure for cloud storage is divided into four groups: data storage cost, network usage cost, transaction cost, and data replication cost. Data storage is the cost of storing data in the cloud, which is charged on a GB-per-month basis. Different storage tiers have different pricing, and some CSPs offer block-rate pricing, where the larger the amount of data, the lower the unit costs. Transaction costs are associated with managing, monitoring, and controlling a transaction when reading or writing data to cloud storage. Network usage costs are based on the amount of data transferred over the network. Data replication cost refers to replicating data from on-premises storage to the cloud or from one instance to another. By default, three copies are stored for each chunk of uploaded data to achieve high data reliability and better disaster recovery. In addition, there are several optional costs, including data management, data backup, and data security. Users can optimise mandatory cost elements, but they cannot avoid them.

In short, understanding the cost structure of cloud computing can be a difficult and intricate task due to the complex pricing models offered by various CSPs. Comparing costs and selecting the most suitable option for a particular application can be challenging. Researchers have attempted to simplify it to make it easier for users to comprehend the complexity of the cloud cost structure, e.g., [7]. Martens et al. [8] have observed that many cloud cost evaluations lack a systematic approach to cost estimation, which is necessary to understand the different pricing models of cloud services. When selecting a CSP, the cost is not the only factor to consider. There are other quality of service (QoS) elements, such as network performance, data availability, consistency, security, etc. This gives rise to certain trade-offs such as storage-computation, storage-cache, storage-network, availability-reliability, and cost-performance [7], which means balancing different factors to make decisions about resource allocation and use. These must be considered when deploying applications to the cloud. A potential solution should be able to find the optimal resource placement strategy for performance by quantifying the QoS elements.

## III. Related work

The field of cloud cost optimisation has received significant attention in recent years, with numerous studies exploring different approaches to reducing cloud computing costs. These approaches concerning the deployment phase can be divided into pre- and post-deployment. Regarding strategies, costs can be reduced by optimising compute costs, network costs, and storage costs or by optimising resource placement, i.e., choosing the most suitable resources and location based on cost and other QoS elements. Storage cost optimisation deals with the cost of storing data literally and the associated costs. It can be done before or after application deployment. Network cost optimisation deals with the cost of using the network to transfer data between different regions and transferring data from storage servers to compute resources. It is applied before application deployment. Lastly, compute cost optimisation deals with the cost of compute resources, such as virtual machines and GPUs. In the rest of this section, we provide an overview of the relevant research in the field, focusing on studies that investigated the use of optimisation algorithms for cloud cost management.

*a) Pre-deployment techniques:* Regarding network cost optimisation, Mansouri et al. [9] proposed an approach to minimize the cost of data placement for applications with time-varying workloads, while Zeng et al. [10] proposed a method for economically deploying edge servers in wireless metropolitan area networks. Shao et al. [11] proposed a data placement strategy for IoT services in wireless networks, which considers user distribution density to determine optimal edge server deployment locations and minimize deployment costs. For storage cost optimisation, Wang et al. [12] proposed a solution based on an architecture and Non-Dominated Sorting Genetic Algorithm II (NSGA-II) for multi-cloud storage. This solution aimed to reduce overall cost and maximize data availability simultaneously by using an entropy approach to find the best option from the set of non-dominated solutions known as the Pareto-optimal set. Moreover, multi-objective optimisation algorithms have been explored for resource placement (storage selection) to find an optimal solution that balances cost, performance, and availability. This approach has shown promise in finding solutions that trade-off between these objectives. Ilieva et al. [13] proposed a new approach for evaluating and ranking cloud services, which combines multi-criteria and fuzzy approaches to consider various factors. Oki et al. [14] presented selection models for cloud storage to satisfy data availability requirements, and Halimi et al. [15] proposed a QoS-focused approach for storage service allocation that considers various QoS objectives to improve the performance and scalability of cloud storage systems.

*b) Post-deployment techniques:* For storage cost optimisation, Liu et al. [16], [17] proposed an algorithm based on Markov decision processes and deep reinforcement learning to determine cost-effective tiers and evaluated on real-world traces and proved to achieve significant savings. Mansouri et al. [18] proposed an optimal offline dynamic programming algorithm and two practical online algorithms for determining the placement of objects in hot and cold tiers in tiered cloud storage to minimize monetary costs and improve the quality of service. Erradi et al. [19] also proposed a cost-optimising approach for tiered cloud storage. Liu and Shen [17] proposed a method for efficient storage resource distribution, which includes three enhancement strategies to reduce payment cost and service latency. Compute costs optimisation area of research focuses on cloud resource provisioning, where optimisation algorithms are used to find the optimal allocation of cloud resources to ensure necessary levels of performance and availability while minimizing costs. For example, Traneva et al. [20] presented a method for cost optimisation by resource provisioning in cloud computing by considering uncertainty in resource usage. Zheng et al. [21] proposed an algorithm for cost optimisation for scheduling scientific workflows on clouds under deadline constraints.

*c) Discussion:* In summary, the literature review demonstrates a growing body of research focused on cloud cost optimisation, with various optimisation algorithms proposed and evaluated. However, these algorithms do not address industry-specific requirements, and most of them are not evaluated in

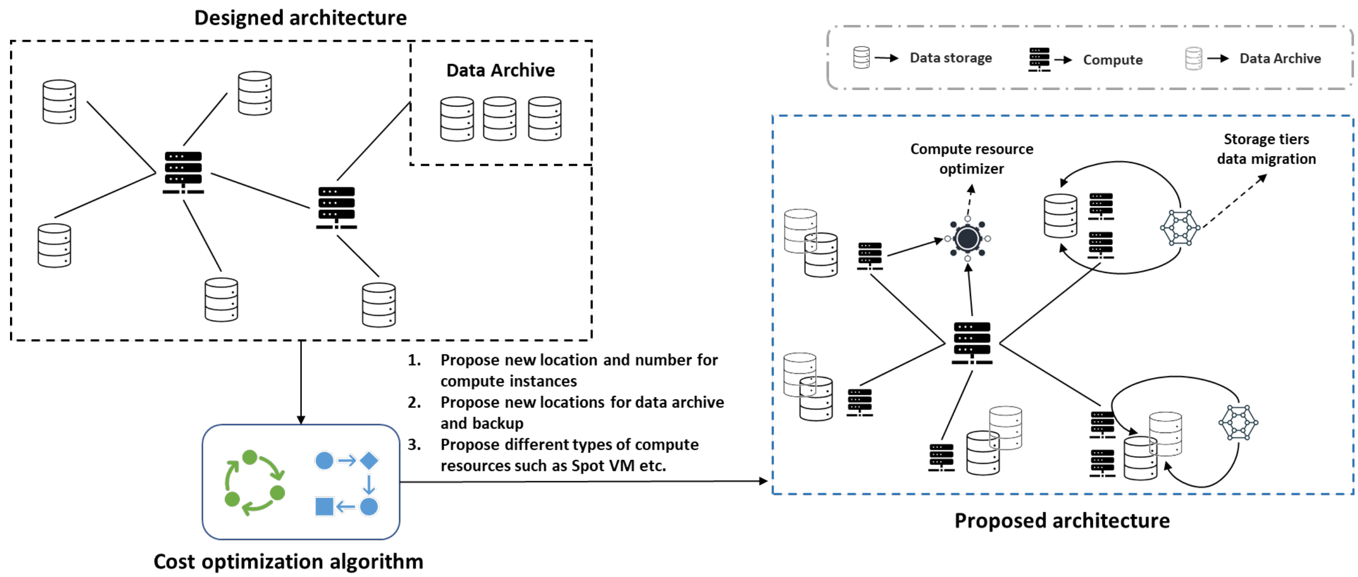


Fig. 1. High-level diagram illustrating an ideal cost optimisation scenario.

real-world scenarios. We explore a new solution to cloud cost optimisation by modelling cost elements and cloud resources in the form of a graph and quantifying QoS elements to address the trade-offs. Graph theory has been proven effective in finding optimized solutions for problems in a wide range of domains, including the cloud environment, such as data replication [22] and caching [23]; therefore, we find it promising for cloud cost modelling and optimisation as well. The solution should be applicable to new and existing applications in the cloud; be able to address multiple challenges, such as compute and storage resource placement and network cost optimisation; and have the ability to incorporate QoS requirements.

## IV. Cost optimisation

We break down the complete process of cloud cost optimisation into six paths described in what follows.

1) *Optimisation of the network cost by proposing new locations for cloud resources*: This path focuses on identifying the most suitable locations for compute and storage resources to ensure optimal performance and cost efficiency. By analyzing data access patterns and workload requirements, the solution shall suggest new locations for the resources that can achieve the best possible balance between performance and cost. However, if it is done post-deployment, migration costs will apply.
2) *Optimisation of storage cost by proposing the optimal number of storage instances*: This path expands the optimisation approach to include storage, backup, and archiving. By proposing new locations and an optimal number of storage instances, the system can ensure that data is stored cost-effectively and efficiently. In a hybrid cloud, a storage instance is a server and for the public, a storage bucket in a zone/region.
3) *Optimisation of compute cost by proposing the optimal number of compute resources (VMs, GPU):* By carefully selecting the number of virtual machines and GPUs needed for a specific workload, the overall cost of compute resources can be minimized, while still ensuring that the application runs smoothly and efficiently.
4) *Optimisation of compute cost by scaling compute resources*: This path focuses on optimising the system's compute resources to ensure they are appropriately scaled to meet the workload requirements. Analyzing the workload patterns and scaling the compute resources accordingly can reduce costs and improve performance.
5) *Optimising storage costs through data migration between storage tiers*: This path involves migrating data between storage tiers to save cost by identifying opportunities to move data to lower-cost storage tiers without compromising performance. It can be done by analysing data access patterns and workload requirements.
6) *Proposing more efficient and cost-effective resource alternatives*: This path involves identifying alternative solutions that can provide the same or better performance while reducing costs. By analyzing the system's current configuration and workload requirements, the proposed solution can suggest better and cost-effective alternatives, such as different CSP or hardware options, such as spot VMs instead of regular VMs, containerization instead of VMs, serverless computing instead of fixed provisioning, etc. However, this may require changes in the architecture/implementation of the system.

An illustration of an ideal cost optimisation scenario containing all the six paths is exemplified in Figure 1. The goal is to show how each optimisation path will work when put into action. The *proposed architecture* is not the actual representation of the optimised scenario. The *designed architecture*, which could be for a large software application or a multi-step big data workflow, has data storage in five locations, with compute resources placed at two locations near all storage servers. Additionally, a separate data lake is deployed for the data archive. After passing through the cost optimisation techniques, this architecture gets modified based on user requirements, such as required resources and their location, and data access patterns (i.e., improved architecture). This scenario can offer several benefits; firstly, assuming compute resources only access one data storage, moving them closer to the data store will significantly reduce data transfer costs. Secondly, centralizing the system status and moving the data archive to the same location as the actual storage will reduce overhead costs and improve overall efficiency.

## V. Graph-based cloud cost modelling and optimisation

The proposed preliminary solution, shown in Figure 2, targets paths 1, 2, and 3 as described in Section IV and aims to find the most suitable placement and the number of cloud service instances in terms of storage and compute resources, hence optimising compute, storage and most importantly network costs. This involves developing a model that considers the number and location of cloud services deployed, the data access patterns, and an algorithm to suggest the most efficient number and location for cloud service instances.

Followings are the steps to implement the proposed solution:

*a) Purpose clarification:* It is an essential first step that involves the identification of both the functional and non-functional requirements that are specific to the industry. Functional requirements include the specific tasks that the application should be able to perform, such as data storage, data processing, data retrieval, and data analysis. On the other hand, non-functional requirements include the quality attributes that the application should possess, such as latency, availability, and durability. Considering these requirements makes it possible to determine the necessary resources required, including storage servers and compute resources.

*b) Designed architecture:* In this solution proposal, designing an architecture for the software application is critical. The architecture outlines the geographic locations where cloud resources will be deployed, their type, and the data flow within the application. It will define the constraints regarding which resources can be moved and to what extent (within a country, region, or continent) and the resource requirements (the amount of storage space, CPU, and RAM required). This also must consider various factors, such as the number and location of cloud resources, data flow, QoS requirements, etc. The output of this step will be used as a basis for graph creation in the next step. To ensure the feasibility of the architecture, it is essential to consider several key factors during the design phase such as the verification of cloud resource availability in designated regions and specifying data flow in the application for efficient processing and storage.

*c) Graph creation:* Creating a graph based on the information gathered from the previous steps presents a few challenges. Cloud resources can be treated as graph nodes, and the network usage cost can be specified as the edge. However, multiple cost elements must be considered, such as storage, compute, security, and other cloud resource costs. As seen in Figure 2, the nodes in yellow represent compute resources, while the blue nodes represent storage resources. $C_{a1}, C_{a2}$... are the compute resource costs, and $S_{a1}, S_{a2}$... are the storage resource costs. We denote the edges as $e_{ij}$, which represent the weights of the graph and in this case the costs of services. For finding the cost-optimal solution, storage and compute costs are not the only factors to be considered. To ensure that all costs are accounted for, we must also consider the costs of services, such as network usage costs. To solve this problem, we can treat $n_{ij}$ as the network cost between the nodes $C_i$ and $S_j$, hence $e_{ij}$ be as follows: [1]

$$e_{ij} = C_i + n_{ij} + S_i \tag{1}$$

To incorporate QoS elements, we assume latency, availability, and durability as $l_{ij}$, $a_{ij}$ (quantified value based on the SLAs), and $d_{ij}$ (numerical representation of the redundancy model), respectively. Similarly, taking $w_l$, $w_a$, and $w_d$ as weights in percentage to define the importance of each factor as per requirements and $N$ as the normalising constant. Using a cost-effectiveness ratio (CER), we can quantify $l_{ij}$, $a_{ij}$, and $d_{ij}$ as:

$$f(CER_l|CER_a|CER_d) = \frac{Cost}{(1-(l_{ij}|a_{ij}|d_{ij}) \times N)} \times w_{(l|a|d)} \tag{2}$$

For example, if the cost of *Server A* is \$150 and its latency performance is 0.3 [2], the cost-effectiveness ratio would be: $\frac{150}{(1-0.3)\times 10} = 21.42\frac{\$}{sec.}$. Similarly, if *Server B* has a cost of \$200 and a latency performance of 0.2, then its cost-effectiveness ratio would be: $\frac{200}{(1-0.2)\times 10} = 25\frac{\$}{sec.}$. In this case, *Server A* would be more cost-efficient than *Server B*.

Hence, $f(QoS)$ will be:

$$f(QoS)_{ij} = f(CER_l)_{ij} + f(CER_a)_{ij} + f(CER_d)_{ij} \tag{3}$$

Putting Equations 1, 2 and 3 together, we will get:

$$e_{ij} = C_i + n_{ij} + S_i + f(QoS)_{ij} \tag{4}$$

This way, additional costs, such as security and encryption, can also be included in the calculation. Once the weights are specified on the edges, all possible resource combinations in a graph can be formed. For example, as per Figure 2, *Location A* has three compute and two storage instances; similarly, *Location B* has five compute and four storage instances. Assuming only one storage and one compute instance are required in each location, the number of resource combinations in the graph for just these two locations will be $Num(S)_A \times Num(C)_B + Num(S)_B \times Num(C)_C$, which in this case will be $(2 \times 5) + (3 \times 4) = 22$. The total

[1] Equation 1 refers to a general formula, multiple variations might be used to avoid redundancy in the cost calculation.

[2] Latency performance of 300ms divided by 1000.

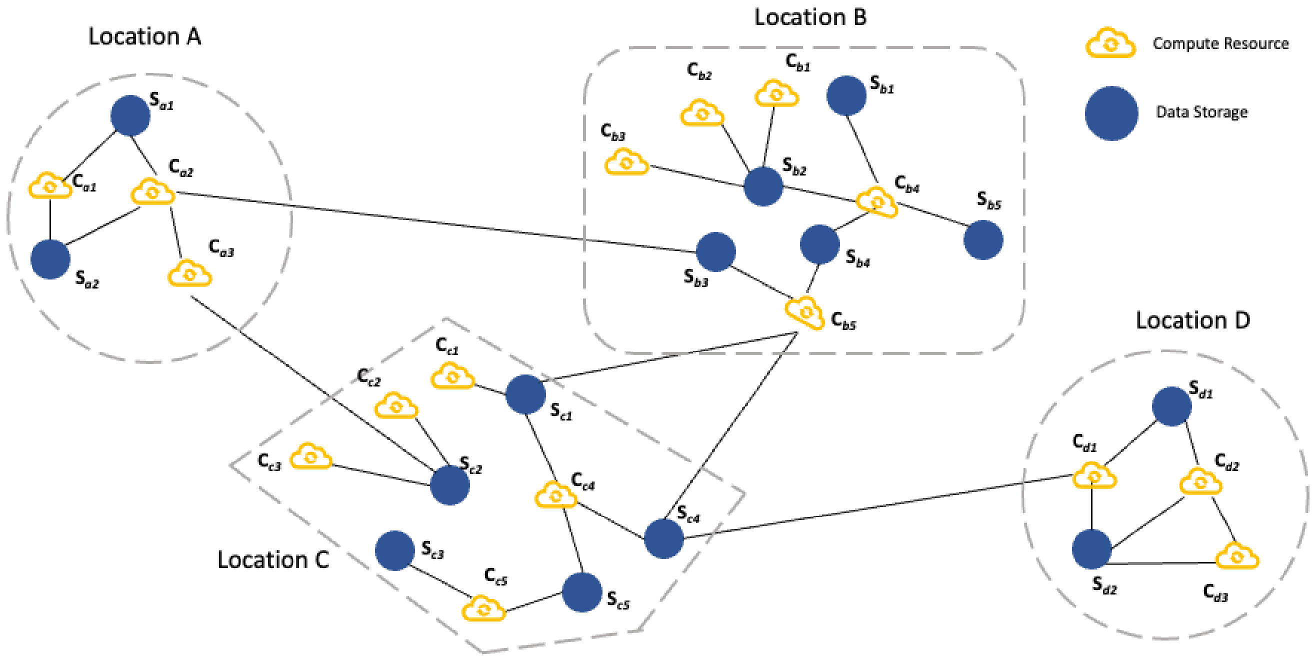


Fig. 2. Illustration of directed graph data structure based on required cloud resources for big data applications deployed in a cloud environment.

number of combinations can be calculated using Equation 5. The complexity and the total number can be increased when more resources are required in each location.

$$Total = \Sigma_{x=1}^{n}(Num(S)_x \times Num(C)_{x+1}) \qquad (5)$$

*d) Implementation:* A potential implementation could begin with the collection and normalization of input data. The following are the detailed execution steps:

1) Obtain a list of regions where the desired services are available within the specified geographic location. This information will be utilized to create the graph nodes.
2) Using the CSPs' "Billing API", such as the one provided by Google[3], for cost estimations of the requested services, the weights of the edges will be set.
3) Nodes and edges obtained in the previous steps will be combined and transformed into a graph. This data structure will serve as the foundation for the next step.
4) Find the optimal solution that minimizes the cost and satisfies the necessary performance requirements by addressing the constraint problem. These include, for example, trade-offs, a limit on the amount of storage or compute resources available within a specific cloud provider or location, or QoS requirements of the applications running on the cloud resources, which could impact the selection of suitable cloud resources for cost optimization.

*e) Possible algorithms:* One or more shortest-path algorithms or graph-neural network (GNN) [24] could be used to find the optimal solution for big data application deployment in the cloud environment. Shortest path algorithms are a class of algorithms used to find the shortest path between two vertices in a graph. Some of the most popular algorithms are: Dijkstra's algorithm [25], Bellman-Ford algorithm [26], A* algorithm [27], Floyd-Warshall algorithm [28], Johnson algorithm [29]. GNN [30], on the other hand, is a type of artificial neural network designed to operate on graph-structured data.

Such a solution is currently missing in the literature and presents a significant challenge for the cost-effective and efficient use of cloud resources. Our solution proposal offers several benefits, including increased efficiency, and cost savings. However, it is important to note that implementing this solution may require changes to the architecture and behaviour of the system, which could be a trade-off depending on the changes made. The solution uses a graph structure, conceptualizing each region with a cloud service instance as a node for storage or processing. Graph theory is then used to identify the optimal path between nodes, reducing costs by optimizing cloud resource placement. To minimize the cost of network usage, the flow of information between the nodes will be pre-defined, and the data transfer cost from one node to another will be treated as the weight and specified on the edge connecting the two nodes. The *designed architecture* may include resources selected for each location, for example, some in Europe West, others in Europe East, and perhaps some in US East. However, multiple options are available from CSPs for each location, such as EuWest1, EuWest2, EuEast1, EuEast2, and so on. Based on the resources selected in the designed architecture, the algorithm will determine the optimised path between the selected locations.

[3]https://cloud.google.com/billing/docs/reference/rest

## VI. Discussion & Conclusions

In this paper, we explored a graph-based solution for the cloud cost modelling and optimisation problem, as a first step by modelling cost elements and cloud resources in the form of a graph and potential ways to solve the resulting constraint problem. The proposed solution has the potential to address industry-specific needs and provide tailored solutions for particular industries that may have unique requirements or challenges. This is due to its ability to take into account several cloud resources, to define information flow due to the directed nature of graphs, and its applicability to single or multi-cloud environments. However, this solution is not without its challenges. For example, placing compute resources at multiple locations requires careful planning and coordination. Despite these, the proposed solution is a material step in optimising cloud costs. Optimising cloud resources can minimize costs and achieve the maximum performance possible within specific cost constraints. Developing such a solution would be notable in the field and have significant implications for practical applications and the potential for further research.

For future work, we aim to start by developing the proposed graph-based model, which can optimise resource placement and utility in cloud environments for big data workflows. To measure the cost-effectiveness of the proposed solution, it is necessary to simulate the real operations of the application. Hence, existing software applications that have already been deployed on the cloud can be used as input and redeployed based on the optimised solution proposed by the new graph-based solution. This can provide a more realistic evaluation of the proposed solution and its potential effectiveness.

**Acknowledgments**. The work in this paper was partially funded by the projects DataCloud (H2020 101016835), enRichMyData (HE 101070284), Graph-Massivizer (HE 101093202), and BigDataMine (NFR 309691).

## References

[1] K. Robinson, "Why companies are flocking to the cloud more than ever," https://www.businessinsider.com/cloud-technology-trend-software-enterprise-2021-2, 2021, accessed: 2023-02-20.

[2] B. S. Rajeshwari, M. Dakshayini, and H. S. Guruprasad, *Workload balancing in a multi-cloud environment: challenges and research directions*. Springer, 2022, pp. 129–144.

[3] R. Donida Labati, A. Genovese, V. Piuri, F. Scotti, and S. Vishwakarma, *Computational intelligence in cloud computing*. Springer, 2020, pp. 111–127.

[4] N. Nikolov, Y. D. Dessalk, A. Q. Khan *et al.*, "Conceptualization and scalable execution of big data workflows using domain-specific languages and software containers," *Internet of Things*, vol. 16, p. 100440, 2021.

[5] D. Roman, R. Prodan, N. Nikolov *et al.*, "Big data pipelines on the computing continuum: Tapping the dark data," *Computer*, vol. 55, no. 11, pp. 74–84, 2022.

[6] A. Q. Khan, N. Nikolov, M. Matskin *et al.*, "Smart data placement using storage-as-a-service model for big data pipelines," *Sensors*, vol. 23, no. 2, p. 564, 2023.

[7] Y. Mansouri, A. N. Toosi, and R. Buyya, "Data storage management in cloud environments: Taxonomy, survey, and future directions," *ACM Computing Surveys*, vol. 50, no. 6, pp. 1–51, 2017.

[8] B. Martens, M. Walterbusch, and F. Teuteberg, "Costing of cloud computing services: A total cost of ownership approach," in *Proceedings of the 45th Hawaii International Conference on System Sciences*. IEEE, 2012, pp. 1563–1572.

[9] Y. Mansouri, A. N. Toosi, and R. Buyya, "Cost optimization for dynamic replication and migration of data in cloud data centers," *IEEE Transactions on Cloud Computing*, vol. 7, no. 3, pp. 705–718, 2017.

[10] F. Zeng, Y. Ren, X. Deng *et al.*, "Cost-effective edge server placement in wireless metropolitan area networks," *Sensors*, vol. 19, no. 1, p. 32, 2018.

[11] Y. Shao, Z. Shen, S. Gong *et al.*, "Cost-aware placement optimization of edge servers for IoT services in wireless metropolitan area networks," *Wireless Communications and Mobile Computing*, vol. 2022, 2022.

[12] P. Wang, C. Zhao, W. Liu *et al.*, "Optimizing data placement for cost effective and high available multi-cloud storage," *Computing and Informatics*, vol. 39, no. 1-2, pp. 51–82, 2020.

[13] G. Ilieva, T. Yankova, V. Hadjieva *et al.*, "Cloud service selection as a fuzzy multi-criteria problem," *TEM Journal*, vol. 9, no. 2, p. 484, 2020.

[14] E. Oki, R. Kaneko, N. Kitsuwan *et al.*, "Cloud provider selection models for cloud storage services to satisfy availability requirements," *IEICE Transactions on Communications*, vol. E100-B, no. 8, pp. 1406–1418, 2017.

[15] O. H. Milani, S. A. Motamedi, S. Sharifian *et al.*, "Intelligent service selection in a multi-dimensional environment of cloud providers for internet of things stream data through Cloudlets," *Energies*, vol. 14, no. 24, p. 8601, 2021.

[16] M. Liu, L. Pan, and S. Liu, "RLTiering: A cost-driven auto-tiering system for two-tier cloud storage using deep reinforcement learning," *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 2, pp. 73–90, 2022.

[17] G. Liu and H. Shen, "Minimum-cost cloud storage service across multiple cloud providers," *IEEE/ACM Transactions on Networking*, vol. 25, no. 4, pp. 2498–2513, 2017.

[18] Y. Mansouri and A. Erradi, "Cost optimization algorithms for hot and cool tiers cloud storage services," in *Proceedings of the 11th International Conference on Cloud Computing (CLOUD 2018)*. IEEE, 2018, pp. 622–629.

[19] A. Erradi and Y. Mansouri, "Online cost optimization algorithms for tiered cloud storage services," *Journal of Systems and Software*, vol. 160, p. 110457, 2020.

[20] V. Traneva, S. Tranev, and V. Atanassova, "Index matrices as a cost optimization tool of resource provisioning in uncertain cloud computing environment," in *Recent Advances in Computational Optimization*. Springer, 2020, pp. 155–179.

[21] W. Zheng, B. Emmanuel, C. Wang *et al.*, "Cost optimization for scheduling scientific workflows on clouds under deadline constraints," in *Proceedings of the 5th International Conference on Advanced Cloud and Big Data (CBD 2017)*. IEEE, 2017, pp. 51–56.

[22] J. Liu, H. Shen, H. Chi *et al.*, "A low-cost multi-failure resilient replication scheme for high-data availability in cloud storage," *IEEE/ACM Transactions on Networking*, vol. 29, no. 4, pp. 1436–1451, 2020.

[23] X. Xia, F. Chen, Q. He *et al.*, "Graph-based data caching optimization for edge computing," *Future Generation Computer Systems*, vol. 113, pp. 228–239, 2020.

[24] F. Scarselli, M. Gori, A. C. Tsoi *et al.*, "The graph neural network model," *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.

[25] A. Goldberg and T. Radzik, "A heuristic improvement of the bellman-ford algorithm," Stanford University - Computer Science Department, Tech. Rep., 1993.

[26] J. Bang-Jensen and G. Z. Gutin, *Digraphs: theory, algorithms and applications*. Springer Science & Business Media, 2008.

[27] S. J. Russell, *Artificial intelligence a modern approach*. Pearson Education, Inc., 2010.

[28] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, "Introduction to algorithms," 1990.

[29] D. B. Johnson, "Efficient algorithms for shortest paths in sparse networks," *Journal of the ACM*, vol. 24, no. 1, pp. 1–13, 1977.

[30] A. Micheli, "Neural network for graphs: A contextual constructive approach," *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 498–511, 2009.