International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 8, 2025
https://theaspd.com/index.php

Optimizing Data Distribution And Privacy Protection Over Multi-Cloud Storage For Pdts Scheduling

K.H.Vani[1*], Dr. P Balamurugan[2]
[1,*]Assistant professor, Department of Computing, Coimbatore institute of technology, Coimbatore, Tamilnadu- 641014 vanikhassistantprofessor@gmail.com
[2]Associate professor, Department of computer science, Government arts college (Autonomous), Gopalapuram, Coimbatore, Tamil Nadu 641018, spbalamurugan@rediffmail.com

Abstract: *Privacy of electronic data that is of utmost priority is even in cloud computing environment. Putting data on a specific cloud service is quite risky since it may be leaked or compromised by an insider threat. Spreading some cloud storage providers (CSP) can be the solution to the preceding and will be of benefit by boosting the safety and file uploading by the dispersion of information on separate platforms. In such multi-cloud environments to reduce the time taken in uploading data, a suitable storage model was proposed. More intelligent erasure coding and task scheduling of partially dependent tasks (PDTs) forms the basis of this scheme and the task of this challenge will be perceived as a multi-objective optimization problem which falls in the category of NP-hard problems. A solution to this is provided by generating answers to this issue using a Non-dominated Sorting Genetic Algorithm (NSGA), which forms a collection of Pareto-optimal solutions of the conflicting objectives i.e. access time reduction, and increase in the level of data availability. When a user cannot make the best alternative that is proposed in Pareto front, entropy based decision-making process can be adopted to facilitate to make optimal decision. The fact of successful interaction between erasure coding and smart CSP selection matters to the privacy of the data and successful utilization of the multi-cloud storage. In order to enhance on this more the Muddy Soil Fish Optimization Algorithm (MSFOA) has been utilized together with a privacy-aware multi-objective optimization framework. This strategy together with a superior method of Pareto solution using NSGA transpires to outdo conventional security algorithms in several assessment aspects. The framework proposed is actually experimented in a variety of real life applications of the cloud storage and the same verifies the effectiveness of the framework.*
Keywords: *Multi-cloud storage, Pareto-set, NSGA, erasure coding, MSFOA, security.*

1. INTRODUCTION

Organizations are relatively turning to multi-cloud strategies to influence innovation and avoid being tied to a sole provider in addition to not limiting their service options. Nonetheless, security is also the main issue when adopting cloud systems. Transferring sensitive business data to a third-party cloud services provider (CSP) poses a certain degree of danger [1]. Thus, enterprises need to selectively, and carefully, evaluate and select providers who will ensure that data protection is extensive. Cloud storing of data predisposes organizations to possible external threats since the absolute security of data on the internet is hard to obtain. This creates opportunities of lack of authorize or concealed gaps [2].

With the developments of this trend of cloud computing, the usage has spread to cloud computing, file sharing and distributed data storage functions, which have become mandatory [3]. Multi-cloud systems should allow the fast and secure access to the data to help organizations maintain efficiency and meet the needs of users [4]. Other solutions such as scalable parallel file systems are now preferred where data is chopped into regular sized pieces and further stored in a cluster using several small servers [5]. This will not only bring efficiency in processing but also on data security via distribution and redundancy. In addition, the outsourcing of data is becoming a norm [6], but issues related to poor authentication procedure are also still on the rise. Due to almost infinite storage space, business, state organizations, and educational institutions are becoming more inclined toward using cloud platforms [79]. However, using only one CSP can result in the problems of vendor lock-in and system-scale failures [10]. Efforts to curb these threats have put a move towards multi-cloud flows where the data is kept in redundancy across several CSPs [11]. The approach enhances fault tolerating and creates flexibility, which can be of significance in data-intensive systems such as digital archives, electronic health records and data backups.

The strong points of cloud storage is that it encrypts data prior to posting content so that a stranger can not decode the data. Only those users who have the decryption key can unlock content and have safe access to the original content where no one can decrypt. Since managing the storage and security of a complex multi-

cloud environment is a complex problem that belongs to the NP-Hard problem categories, heuristic and metaheuristic algorithms are quite useful when it comes to obtaining a near-optimal solution. The term metaheuristics denotes that the heuristics have a more general applicability than domain-specific heuristics indicated by the prefix meta [12]. They are common in performance boosting of the combinatorial and numerical optimization problems by finding attractive solution spaces [13]. The majority of the privacy regimes in multi-cloud systems consist of two general stages: data sanitization and recovery [14]. In this framework, Muddy Soil Fish Optimization Algorithm (MSFOA) is proposed to create the best encryption keys in line with given sanitization techniques. Further, the erasure coding [15] is instrumental to the low storage cost alternative to full replications schemes, as well as minimizes the bandwidth resource requirement of data passable on the network [16]. A fast non-dominated sort is also in use to reduce solution complexity and increase efficiency of the system. The integration of the PDT mechanism, which is based on the NSGA and can be applied to solve nonlinear optimization issues, guarantees the diversity of the set of solutions, which would fit different tastes of users. Finally, the Pareto front-based approach is employed in the locating of cloud configurations that present non-dominated and user-relevant trade-offs [17]. This procedure deals with some of the weaknesses of the previous genetic algorithms; great weaknesses, such as weak robustness, dimensional differences, and weight assignment bias. The long-term goal is to develop cost and privacy-sensitive data placement policies to provide users the high availability and security of data in the environments of complex cloud storage.

2. RELATED WORKS

In the context of both common individuals and academic researchers, it has become quite crucial to store the information on different cloud storage platforms. Massive interest in the field has lately been ignited by the tremendous development of cloud computing technology. Although a significant part of the available literature is devoted either to the encryption of data in multi-cloud systems or optimization of the data transfer process, the combination of both objectives is rather scarce. One of the most prominent contributions in this regard is the approach that was suggested in [18] that focuses on the issues of privacy in terms of data sharing among several cloud service providers (CSPs). The data slicing protocol they use is index-based, cryptographically and protects the data as it is sent over. In particular, it encrypts both private keys and entire the dataset using RSA and 3DES respectively and stops unauthorized access by malicious insiders. Furthermore, the machine also offers more protection by way of dynamic file slicing, which means that a user would be able to specify the extent to which a file in the system is to be sliced into by an interactive interface. In a different experiment, [19] presented Triones, a methodical solution that in correlation with erasure coding to aid in structured data preservation on a number of cloud environments. Triones addresses the issue of optimization of the placements of data through the use of geometric modeling and nonlinear programming processes. It allows it to deal with multiple, frequently inconsistent, optimization objectives. Besides, it is a scalable framework, which can accommodate an increasing number of users without being over-optimized. Triones efficacy has been proven by its track record in the field testing on a broad real-world cloud service environment. An algorithm [20] tried to give to enhance the security of data storage and an architecture that minimizes the threats of hostile insiders and files threats on multi-clouds storage functions. Such approach avoids the situation that insider attacks make to get good data, thus provides a secure environment in which the data owner can secure and have access to the data in a multiple cloud environment without the fear of having conflicts in merging files. The proposed architecture decreases the chances of malicious insider attack, and the plan protects the resources of the provider against malicious files. All kinds of data including video files can be encrypted with the use of the index- based cryptography. The common key encryption algorithm satisfies the secrecy and privacy security standards in securing user data [21].

The attacks also get extended in different scenes such as in state, societal, at a personal level, within a social network, with the identification of certain forgeries and so forth [22]. Prior studies of deep learning steganographic application are still limited in a number of ways. The authors bring a novel start-to-finish channel attention methods in picture steganography. Through channel interdependencies, it is possible to generate a special channel attention module that is incredibly adjustable to dynamically change channel-specific properties in the deep picture representation [33]. Much research is being concentrated on developing the security system of huge amounts of data storage. In the recent research [24], it was focused on how to

develop the encryption scheme of large data storage in multiple cloud storage. It is a low cost way to avoid vendor lock-in, it is also suitable in mission critical application. The paper does not make a reference to latency or storage limitations. Safe data storage on a multi-cloud environment via slice based safe storage means that a file is divided into several parts. Manipulating the data slices will take excessively much RAM in case these slices are too vast. Hence, the variety of slice size should be used depending on the file size. The data slicing rules may be used to designate the slice size during uploading a file of a user [25]. Besides, cost and time, in most cases are the two most important details of interest to most customers when their activities are done in the data centre of the computing CSP. The topic of one constrained area to obtain an extreme value in another has had many studies about the topic. It may not be practical to request specific restrictions to users but rather the CSP may be well placed to provide a number of options to users to enable them to select that which is most suitable to them in the context of time constraints on both sides. Nevertheless, such studies tend to ignore the possibility that customers might not make any firm conclusions about their time or financial budget.

3. PROPOSED METHOD

The scheduling strategy is more elaborate because cloud PDTs incorporates workflow tasks together with independent jobs. To solve this problem with the best possible solutions the conventional NSGA has been applied and with this a superior algorithm with the use of PDT, chromosome coding and the NSGA-III can be recommended. The outcomes of simulation tests reveal that the augmented version of the NSGA algorithm is much more successful than the classical one provided that the variant is being tested in the same conditions in respect of the population size and the number of generations. In particular, the resulting Pareto-optimal front of the improved algorithm has more unique, and not redundant solutions, it is also closer to optimal one, and its individuals are more densely distributed around it. Nevertheless, the bandwidth latency and other possible interference which can possibly occur during the task scheduling will be another factor that the cloud service providers should also consider, in order to have a real life application. It is necessary to mention that the given work is concerned mostly with optimization of the task computing in terms of time and money spent, but does not include wider relatively network concerns.

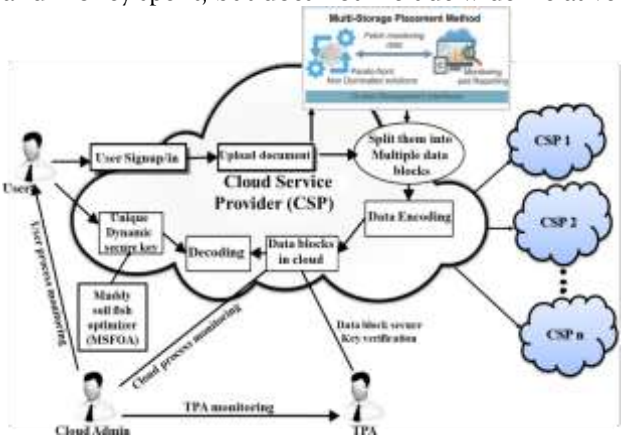

Figure. 1 Outline of effective Multi-storage scheme

The second procedure leads to generation of dynamic unique key. The key obtained and optimised with the help of the MSFOA approach is also checked with the help of third-party authority (TPA). It sends the administrator the key whenever an operation such as a delete or an append operation is performed. Meanwhile another dynamic key will be introduced and enhanced. One of the processes can be accomplished in each block, and therefore in case the operation is completed in this way by using the key, the TPA is expected to authorise it. The section called the cloud admin will continually monitor the TPA whilst the user admin would compare the keys which have been optimized and approved by the TPA with the keys generated. Admin is a go-between between the user and the TPA. Upon verification of optimized keys by the TPA following the actions of the user, the administrator monitors all actions carried out by a user and a TPA. The administrator keeps a track in each course of the cloud service process. The user chooses between a Pareto-

optimal cloud storing region while utilizing the set of non dominated storage nodes manually. The decision-making process produces a list of the storage facilities that are not Pareto dominated. Figure 2 illustrates the inner and the exterior encoding procedures. These two procedures are optimized individually with the help of MSFOA. They are performed in the multiple phases of erasure coding.

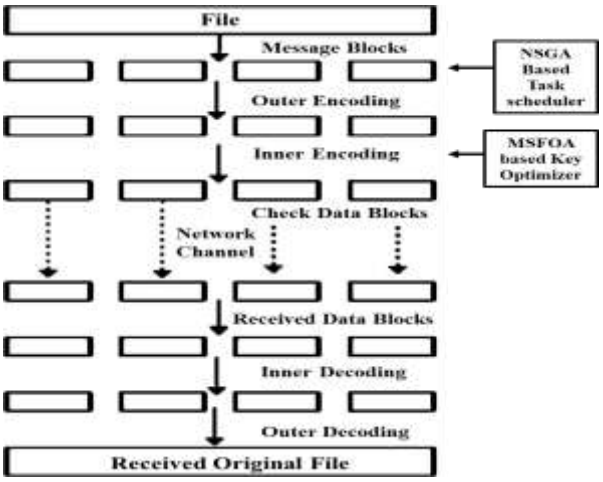

Figure 2 The NSGA and MSFOA-based operators in data processing

The users are charged as per the multiple time periods in blocks of lease period in pricing model of PDT scheduler and it follows a pay-as-you go mode of billing. As the user reserves a block of at least an hour then the cost of the entire hour is charged even when the user has used the block briefly. The reason is that the partial utilization is rounded off to whole unit time. Moreover, as in the majority of cloud settings, data transport is free, the related aspect is not included into the model. Also, a workflow is expected to rent an unlimited number of blocks. In the occasion when a block is leased soon before it is prepared to accomplish a task the retardation of block acquisition happens at that initial boot-up duration. There is also the need to consider the performance variation created through virtualization of a resource, the distributive quality of the assets and others. Time of terminations does not influence the workflow organizing process; therefore it is not taken into consideration when a block is released. In a cloud, the assumption made is always the average bandwidth between blocks has a fixed value. And moreover, the volume of data transmitted defines the time spend on transferring the data between two tasks; in case of having two tasks in one and the same block, the total period of time is considered to be equal to zero.

An outer encoding procedure is the process which separates an input file into several data block components which are both transparent and functionally identical by using PDT and NSGA modules. Since any given fragment in some file will be a fragment in the outer encoding, it is extremely hard to list the count of fragments in that file. In this way, basically, as far as the user perspective is concerned, the Pareto set is very important as it provides the user control over the optimisation process. The user can then state their criteria of service quality then proceed to select a storage infrastructure of choice out of a list of pareto optimum storage infrastructures. The reason behind this is normally one or more quality features are highly likely to change frequently over the passage of time like performance. This mapping of the storage nodes is no longer a satisfactory one in as much as one of the Key MSFOA because of the same motive and so the re- computation of the Pareto set has to start its course again.

Pseudocode: NSGA-based Multi-Cloud Task Scheduling with Erasure Coding and Security

```
Algorithm: Secure Multi-Cloud Scheduling with Enhanced NSGA and Erasure Coding

Input:
   T = {t1, t2, ..., tn} : Set of Partially Dependent Tasks (PDTs)
   C = {c1, c2, ..., cm} : Set of Cloud Storage Providers (CSPs)
   D : Dataset to be sliced and stored
   N : Population size
```

```
   G : Number of generations
   (n, k) : Erasure coding parameters

Output: Optimized task allocation and secure data distribution

Begin
   // Step 1: Encode workflow into chromosomes
   Initialize population P with N chromosomes
   For each chromosome in P:
      Encode scheduling strategy, CSP selection, encryption settings

   // Step 2: Fitness Evaluation
   For each generation g = 1 to G:
      For each individual in population P:
         Evaluate objectives:
            - Cost (minimize)
            - Execution time (minimize)
            - Availability (maximize)
            - Data security (maximize)
         Apply erasure coding (n, k) on data D
         Slice and distribute encrypted blocks to CSPs

      // Step 3: Selection, Crossover, Mutation (with MSFOA-based tuning)
      Select parents based on Pareto dominance
      Apply perturbation-based multipoint crossover
      Apply dynamic range mutation
      Generate offspring population P'

      // Step 4: Combine and Sort
      Combine P and P' into R
      Sort R using fast non-dominated sorting
      Select N best individuals for next generation P using crowding distance

   // Step 5: Output the final Pareto front
   Return final non-dominated set (Pareto-optimal task allocation)
End
```

4.3 Mathematical Analysis

This section presents the mathematical formulation of the proposed multi-objective optimization model for secure and efficient task scheduling in a multi-cloud environment. The objective is to optimize several conflicting goals—such as minimizing execution time and cost while maximizing availability and data security. The approach is based on enhanced erasure coding and an NSGA-based evolutionary algorithm. The model defines key decision variables, multiple objective functions, and realistic constraints that reflect the dynamics of cloud service providers, task dependencies, and resource limitations.

Objective Functions

The optimization problem aims to balance multiple objectives:

{1. Minimize Total Cost}

$$Z_1 = \sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij} \cdot Cost(t_i, c_j) \quad (1)$$

{2. Minimize Execution Time}

$$Z_2 = \sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij} \cdot Time(t_i, c_j) \quad (2)$$

{3. Maximize Availability}

$$Z_3 = \sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij} \cdot A(c_j) \quad (3)$$

{4. Maximize Data Security}

$$Z_4 = \sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij} \cdot S(t_i, c_j) \quad (4)$$

Constraints

{1. Task Assignment Constraint}

Each task must be assigned to exactly one cloud provider:

$$\sum_{j=1}^{m} x_{ij} = 1, \forall i = 1, \dots, n \quad (5)$$

{2. Cloud Resource Capacity Constraint}

The total demand from all assigned tasks should not exceed a cloud's capacity:

$$\sum_{i=1}^{n} x_{ij} \cdot R(t_i) \leq Cap(c_j), \forall j = 1, \dots, m \quad (6)$$

{3. Task Dependency Constraint (for PDTs)}

A task must start only after its predecessor has completed:

$$Start(t_j) \geq Finish(t_i), \text{ if } t_i \rightarrow t_j \quad (7)$$

Erasure Coding Analysis

Let (n, k) be erasure coding parameters:

{4. Storage Overhead}

$$Overhead = \frac{n}{k} \quad (8)$$

{5. Availability Probability}

If each cloud has availability p, and data is retrievable from any k of n clouds:

$$P_{avail} = \sum_{i=k}^{n} \binom{n}{i} \cdot p^i \cdot (1-p)^{n-i} \quad (9)$$

Final Multi-Objective Optimization

$$\text{Optimize } \{Z_1, Z_2, -Z_3, -Z_4\} \text{ subject to constraints (5)–(7)} \quad (10)$$

The mathematical model addresses the complexities of scheduling partially dependent tasks (PDTs) across diverse cloud platforms while ensuring data confidentiality and system resilience. It incorporates four core objective functions: minimizing cost and time, and maximizing both availability and data security. Constraints are included to ensure valid task assignments, resource limits, and execution order. Additionally, erasure coding analysis quantifies storage overhead and availability, enhancing system robustness. This analytical framework supports the development of a secure and optimized cloud scheduling mechanism, suitable for real-world multi-cloud environments.

4 RESULTS AND DISCUSSIONS

The MATLAB / Simulink platform is used to implement the recently presented model of the NSGA-based multi-cloud framework for the security and storage enhancement. A top-notch platform is utilized for multi-cloud data security and storage using fore-front data preservation methods.

a. Comparisons of CSP datasets

In addition, the proposed approach utilizes four distinct cloud storage providers—Google Drive, PCloud, MediaFire, and MEGA—as the initial dataset sources for the experimental setup. The experiment measures the time required for encoding and decoding data blocks, each with a size of 20 MB, under varying erasure

coding configurations denoted by parameters *(n, k)*. These parameter combinations are selected based on prior experimental considerations, and the results are illustrated in Figure 3**.**

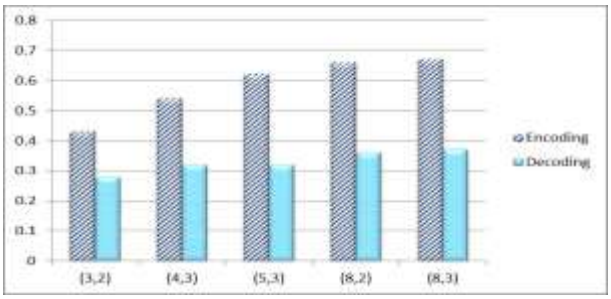

Figure. 3. Enhanced erasure Coding

Comparatively to this overhead, the overall latency for obtaining these kinds of data items housed on the multi-cloud storage is relatively minimal. Erasure coding's overhead lets multi-cloud storage enable data to be striped unlike single cloud storage. Two aspects including upload time, access latency, and energy efficiency are considered for multi-cloud optimization in the proposed approach. Find the upload times for every cloud. Sometimes the qualities of the providers of original cloud storage will vary. The first stage in lowering the tabulated access latency time in Table 1 is the optimizing process. This is acquired for the four separate data as shown earlier (Figure 4a). Figure 4b likewise shows the expected energy. Energy efficiency gains thereafter are generally positive.

Table 1. Access Latency (in ms) Across Multiple Cloud Storage Providers Using Various Methods

Methods	Data			
	Google Drive	MediaFire	MEGA	PCloud
Adaptive Data Slicing	92	87	205	124
Erasure Coding + MSFOA	86	83	197	118
Erasure Coding + MSFOA + PDT	83	80	196	114

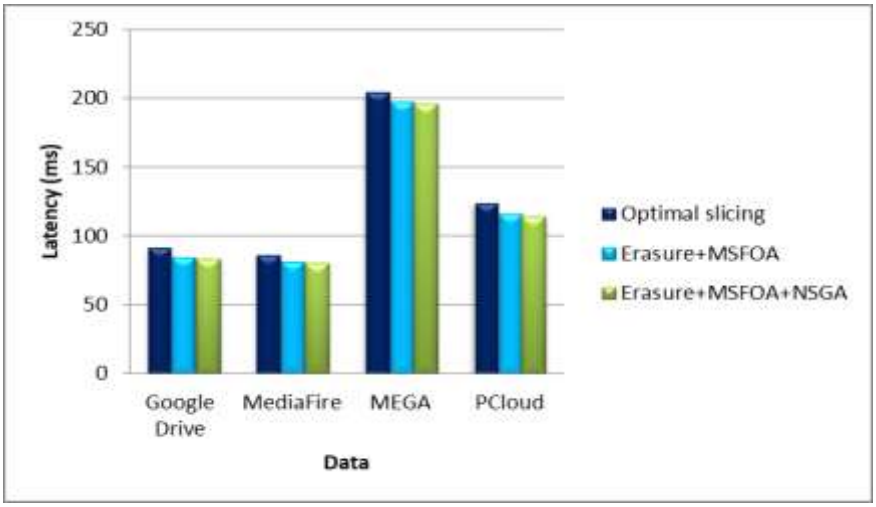

Fig. 4a. Comparisons of latency Process for Multi-Cloud Storage

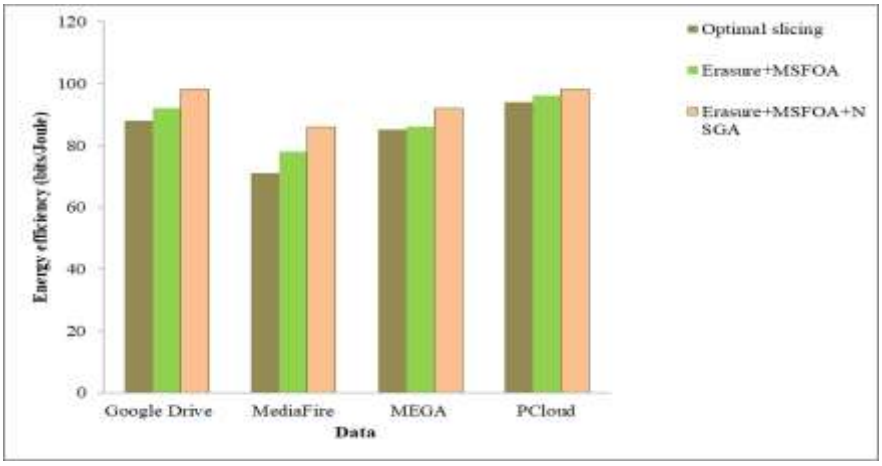

Fig. 4b. Comparisons of Energy efficiency for Multi-Cloud Storage

After that, one considers the upload times of every cloud storage vendor. Table 2 shows tested sizes of 10, 20, 50, and 100MB with optimization for a maximum number of four clouds. Consequently, the data were split into four groups based on the experiment's used cloud storage service providers' count. Figures five show the file slicing with optimization results (in MB).

Table 2. Upload Times for Optimized Data Slices Across Multi-Cloud Platforms

Methods	File size (MB)			
	10	20	50	100
Optimal slicing	8	21	50	84
Erasure+MSFOA	7	18	46	81
Erasure+MSFOA+NSGA	7	16	42	78

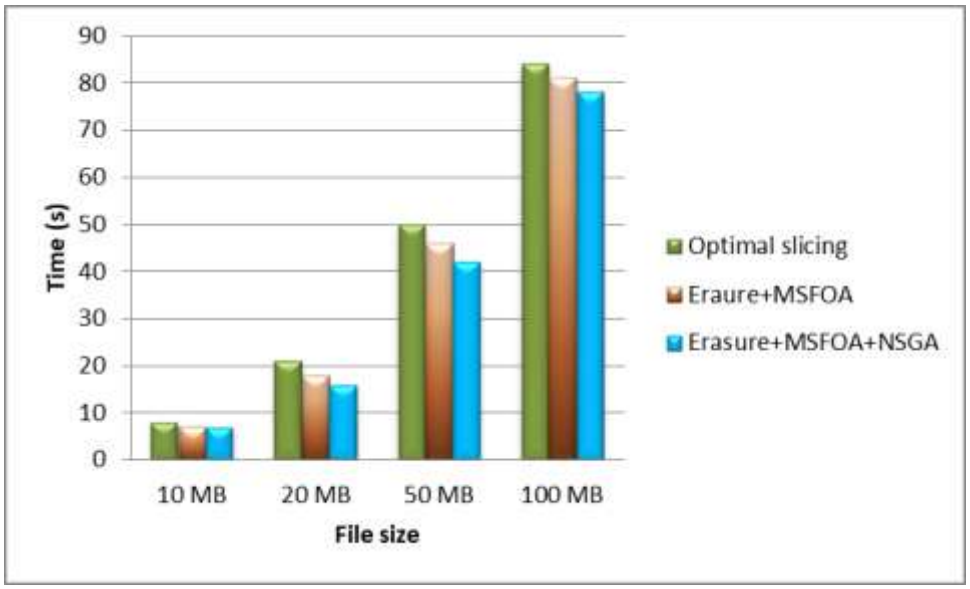

Fig. 5. Comparisons of Multi-Cloud Storage with optimization Process

The effects on system delay, time taken, energy efficiency, and network latency of changing the new coding rate. The replication code regarding the lowest delay and most energy efficiency is at one extreme. This is a result of submitted model simply waiting for a single server to complete the task. Low latency results in faster data throughput and, hence, higher energy efficiency for storage nodes. The other extreme is GA code that has no storage overhead and no fault tolerance (i.e. maximum). Because additional servers are available to handle the same operation simultaneously, expanding the number of repeated requests decreases the service

time. Each job's servicing time is shortened as a result. Because more requests may be processed by the servers in a given amount of time, energy efficiency is increased. However, when the quantity of redundant requests rises, fewer servers are available to handle the next job in the queue, which causes the queue size at the servers to grow. As a result, throughput is lost, which may lead to a possible decline in energy efficiency.

b. Analysis of Pareto-optimal sets

In a bid to determine how effective the modified NSGA-III is in solving the scope of Partially Dependent Tasks (PDTs) scheduling, it is subjected to comparative analysis against traditional versions of NSGA. The two important performance measures in this evaluation are the capability of finding Pareto-optimal solutions and quality of the distribution of such solutions over the Pareto front. As far as benchmarking is concerned, both algorithms are tested at equal conditions: the same population size and quantity of evolutionary generations are used. The number of different, not repetitive solutions, criterion of Pareto-optimality, is considered as the main indicator, defining the optimization ability of an algorithm. The greater number of counts shows the better capability to find the best trade-offs. The average and standard deviation of the Euclidean distances among the neighboring solutions on the Pareto front are computed in order to measure the uniformity of the selection of the solutions. The reduced levels of the metrics imply a shallower and more distributed distribution of people, denoting high quality of solutions and convergence properties of the Pareto-optimal set. It follows that NSGA-III is an evolutionary algorithm, and in the case when the evolutionary algorithm will be applied to the task to be scheduled, both the generation of the evolution and the population size can influence the performance of the method. This part presents the example of the revised algorithm which can demonstrate how the algorithm works when the fixed variation of population size is within the range of 100-500 with the step of 100 and with the evolutionary game large enough. Table 3 averages the mean value, the deviation and the number of not repeated Pareto-optimal solution between each pair of neighbouring solutions.

Table 3 Influence on the population size performances

Population size	The average number of solutions	Mean		Standard deviation	
		NSGA-II	Proposed NSGA-III	NSGA-II	Proposed NSGA-III
100	73.4	103.908	99.412	84.592	81.245
200	123.8	72.489	66.742	68.624	64.217
300	150.4	64.726	60.211	66.062	63.144
400	149.6	65.226	61.2011	66.076	62.426
500	141.4	64.785	59.413	68.454	64.218

From the table, it is clear that population size has a significant impact on routine progress. Rather than assuming that the larger the population, the risk of atavistic behaviour and the time and computer resources wasted are worse. Figure 6 shows the mean distance between nearby solutions and its standard deviation.

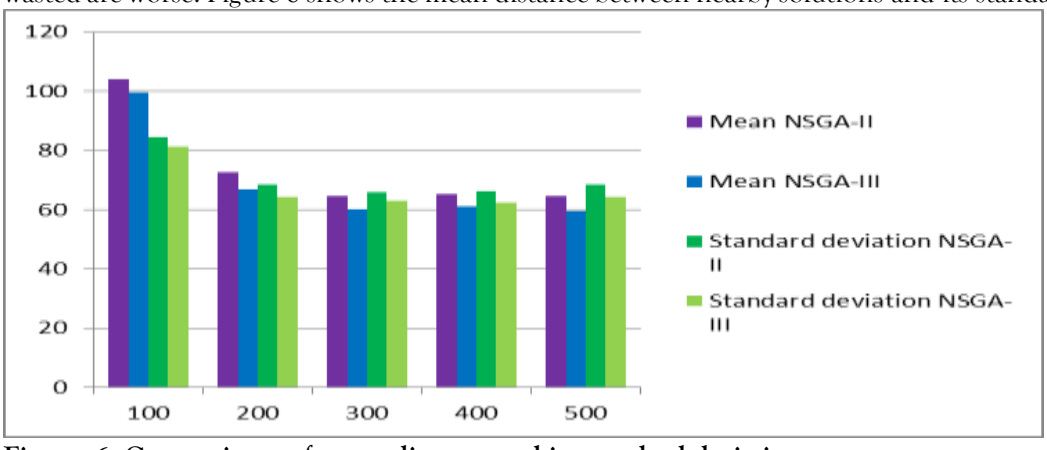

Figure. 6. Comparisons of mean distance and its standard deviations

Instead, one should decide on a population size depending on the PDTs model's scale and make sure the evaluation generation is adequate. It will therefore be much better to select a suitable population size and evaluation generation, or to establish some limit values to have the computation process finish automatically. When the population size and evolution generation are equal, the updated algorithm performs noticeably better than the traditional method.

The effect of raising the number of servers (*n*) on energy efficiency and latency with all other system parameters remain unchanged. We found that raising *n* improves energy efficiency for lower levels of n. This is due to the fact that there have more servers accessible to handle the workload, which lowers average latency and boosts throughput. Increasing *n* may result in increased energy consumption, while lower latency leads to higher throughput. Thus, there is an overall gain in energy efficiency. Nevertheless, increasing n causes latency and throughput to drop for high levels of *n*. This is so because effective service rate, not the number of servers, is what limits the amount of latency improvement. The energy usage increases significantly for very big *n*. The energy efficiency starts to decline with big *n*. Thus, we deduce that there exists an ideal number of *n* that achieves both near-minimal latency and maximum energy efficiency.

c. Comparisons of sports datasets

The chess dataset used in this study is derived from the second dataset referenced in [19, 21]. Figure 6 illustrates the restoration performance prior to data sanitization, comparing various models—GA-CSA, traditional GA, MSFOA-based, and the proposed NSGA-based approach—based on their respective confidence values. Performance is further validated by comparing the Precision Rate (PR), Hit Rate (HR), False Rate (FR), and Data Masking (DM) metrics of the proposed model against those of existing approaches, as shown in Figures 6(a) to 6(d).

For the chess dataset, the proposed model achieves optimal configurations across four different confidence levels for all relevant performance parameters. These comprehensive evaluations clearly indicate the superiority of the suggested method. Consequently, the key strengths of this approach are integrated and benchmarked against current models to highlight its enhanced effectiveness.

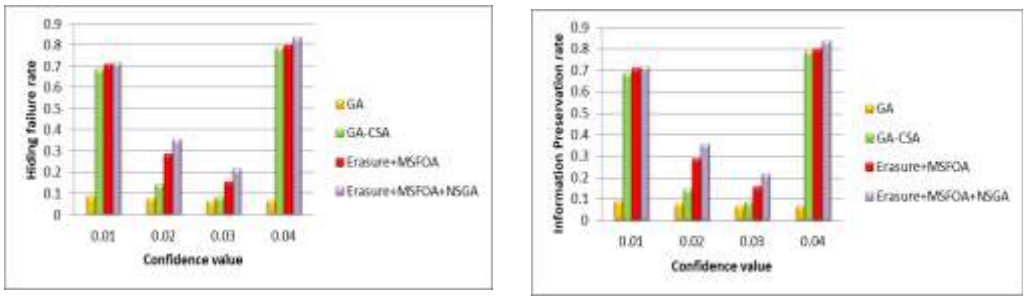

Fig. 7. Simulation Analysis of several models (a) PR (b) Comparisons of HR

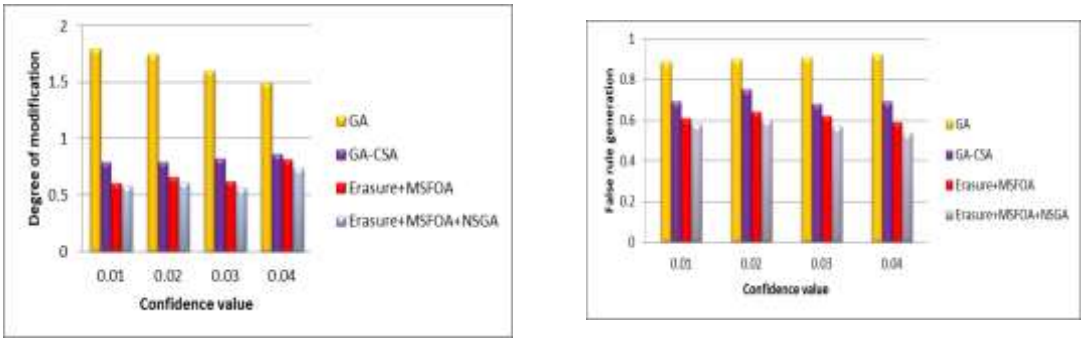

Fig. 7 (c) Comparisons of FR (d) Comparisons of DM

Furthermore, based on the performance metrics discussed above, the proposed model consistently achieves higher confidence values compared to conventional methods. This demonstrates the effectiveness of the newly developed approach in terms of both data sanitization and restoration processes. The robustness of the model is further validated through simulation results under various types of attacks, including the Known

Ciphertext Attack (KCA) [22], Chosen Plaintext Attack (CPA) [24], and Known Plaintext Attack (KPA) [23]. As detailed in Table 4, KPA measures the correlation between the original dataset and the restored version, while KCA evaluates the relationship between the restored data and the reconstructed dataset. These evaluations highlight the enhanced performance and reliability of the proposed framework.

Table 4 Comparison Analyses of KPA and KCA

Methods	Chess	
	KPA	KCA
Erasure+MSFOA+NSGA	0.00115	0.0012
Erasure+MSFOA	0.00117	0.00124
GA-CSA	0.00114	0.0013
GA	0.0078	0.0052

Table 5 provides the CPA analysis for the suggested data preservation plan. The suggested chess dataset scheme is based on the analysis. As a result, the proposed method containing attack-related improvement has been examined, and its effectiveness has been successfully validated.

Table 5 Comparison Analysis of CPA

Methods	Sub 1	Sub 2	Sub 3	Average of correlation
Erasure+MSFOA+NSGA	0.0021	0.0032	0.0042	0.003167
Erasure+MSFOA	0.0025	0.0043	0.0052	0.003868
GA-CSA	0.0026	0.0046	0.0055	0.0042
GA	0.0078	0.0084	0.0086	0.00835

The adopted strategy also performs better than the traditional models when applied to the chess dataset. In terms of multi-cloud storage and security analysis, the results of the proposed MSFOA scheme thus give superior improvement.

5 CONCLUSIONS

In this paper, an efficient and secure multi-cloud storage architecture is suggested, which incorporates the Muddy Soil Fish Optimization Algorithm (MSFOA) and improved erasure coding, as well as the Non-dominated Sorting Genetic Algorithm (NSGA), to simultaneously optimize the positioning of information and the scheduling of a task. The suggested framework depicts a viable use of a Pareto-based NSGA solution to obtain the optimal nodes to be used in storage that will provide an optimal balance between performance, cost, and availability. MSFOA is also used to create encryption keys to facilitate sound data sanitization and restoration so that provision of data to authorized users can be done without being exposed to insider threats. One more, the methodology uses Partially Dependent Tasks (PDTs) slicing-based approach, which considers the upload time, access latency, and storage parameters of different cloud platforms to enhance the efficiency of the transmission. The strategy also uses smart methods of selecting and combining, to reduce the time required in computation when cloud is partitioned and the tasks deployed. With the means of suggesting the most suitable cloud services to the user regarding his/her requirements, the system allows ensuring efficient data migration and provides improved usability. The conducted experiments prove that the suggested solution is superior to the current methods in the sphere of computational efficiency, data security, and the quality of task implementation. The direction of future research will be the widening of the range of goals to address and bringing in more multi-objective optimization plans to enhance and justify the model further.

REFERENCES

[1]Rizvi, S. S., Bolish, T. A., & Pfeffer III, J. R. (2017, March). Security evaluation of cloud service providers using third party auditors. *Proceedings of the Second International Conference on Internet of Things, Data and Cloud Computing*, 1–6.

[2]El Sibai, R., Gemayel, N., Bou Abdo, J., & Demerjian, J. (2020). A survey on access control mechanisms for cloud computing. *Transactions on Emerging Telecommunications Technologies*, *31*(2), e3720.

[3]Li, W., & Karame, G. (2020). *Secure and efficient cloud storage with retrievability guarantees* [Patent]. Google Patents.

[4]Sumathi, M., & Sangeetha, S. (2019). *Survey on sensitive data handling–Challenges and solutions in cloud storage system*. Springer Singapore.

[5]Thakare, V. R. S., & John, K. (2020). Cloud security architecture based on fully homomorphic encryption. In *Architecture and security issues in fog computing applications* (pp. 83–89). IGI Global.

[6]Ali, M., et al. (2019). Distributed file sharing and retrieval model for cloud virtual environment. *International Journal of Engineering and Advanced Technology*, *9*(2), 4062–4065.

[7]Peacock, A. L., et al. (2020). *Systems and methods for monitoring globally distributed remote storage devices* [Patent]. Google Patents.

[8]Fazio, M., et al. (2016). Open issues in scheduling microservices in the cloud. *IEEE Cloud Computing*, *3*(5), 81–88.

[9]Subashini, S., & Kavitha, V. (2011). A survey on security issues in service delivery models of cloud computing. *Journal of Network and Computer Applications*, *34*(1), 1–11.

[10]Wang, P., Zhao, C., Liu, W., Chen, Z., & Zhang, Z. (2020). Optimizing data placement for cost-effective and high available multi-cloud storage. *Computing and Informatics*, *39*(1–2), 51–82.

[11]Srisakthi, S., & Shanthi, A. P. (2015). Towards the design of a secure and fault tolerant cloud storage in a multi-cloud environment. *Information Security Journal: A Global Perspective*, *24*(4–6), 109–117.

[12]Boussaïd, I., Lepagnot, J., & Siarry, P. (2013). A survey on optimization metaheuristics. *Information Sciences*, *237*, 82–117.

[13]Angadi, B. M., Kakkasageri, M. S., & Manvi, S. S. (2021). Computational intelligence techniques for localization and clustering in wireless sensor networks. In *Recent Trends in Computational Intelligence Enabled Research* (pp. 23–40). Elsevier.

[14]Azizi, M. (2021). Atomic orbital search: A novel metaheuristic algorithm. *Applied Mathematical Modelling*, *93*, 657–683.

[15]Vedavathi, D., & Ramaganesh, B. (2016). Efficiency and probing the data hosting scheme by combining replication and erasure coding in the multi-cloud.

[16]Latha, V. P., Reddy, N. S., & Babu, A. S. (2021). Enhancing performance of multi-cloud storage environment using modified erasure coding technique. *Webology*, *18*(6).

[17]Subramanian, K., & John, F. L. (2018). Dynamic and secure unstructured data sharing in multicloud storage using the hybrid crypto-system. *International Journal of Advanced and Applied Sciences*, *5*(1), 15–23.

[18]Su, M., et al. (2016). Systematic data placement optimization in multi-cloud storage for complex requirements. *IEEE Transactions on Computers*, *65*(6), 1964–1977.

[19]Erradi, A. M., & Yaser. (2020). Online cost optimization algorithms for tiered cloud storage services. *Journal of Systems and Software*, *160*, 110457.

[20]Singh, L., Malhotra, J., & Narkhede, S. (2017). Secure data storage in multi-cloud environment using Apache Hadoop. *International Journal of Engineering Sciences & Research Technology*, *6*(9), 656–666.

[21]Nelder, J. A., & Mead, R. (1965). A simplex method for function minimization. *The Computer Journal*, *7*(4), 308–313.

[22]Celesti, A., et al. (2019). Towards hybrid multi-cloud storage systems: Understanding how to perform data transfer. *Big Data Research*, *16*, 1–17.

[23]Wang, P., Chen, Z., Zhou, M., Zhang, Z., Abusorrah, A., & Ammari, A. C. (2021). Cost-effective and latency-minimized data placement strategy for spatial crowdsourcing in multi-cloud environment. *IEEE Transactions on Cloud Computing*.

[24]Alaei, M., Khorsand, R., & Ramezanpour, M. (2021). An adaptive fault detector strategy for scientific workflow scheduling based on improved differential evolution algorithm in cloud. *Applied Soft Computing*, *99*, 106895.

[25]Viswanath, G., & Krishna, P. V. (2021). Hybrid encryption framework for securing big data storage in multi-cloud environment. *Evolutionary Intelligence*, *14*(2), 691–698.