Soft Computing (2020) 24:14503–14530
https://doi.org/10.1007/s00500-020-04802-1

METHODOLOGIES AND APPLICATION

A review of data replication based on meta-heuristics approach in cloud computing and data grid

Najme Mansouri[1] · Mohammad Masoud Javidi[1]

Published online: 11 March 2020

Abstract

Heterogeneous distributed computing environments are emerging for developing data-intensive (big data) applications that require to access huge data files. Therefore, effective data management like efficient access and data availability has become critical requirement in these systems. Data replication is an essential technique applied to achieve these goals through storing multiple replicas in a wisely manner. There are replication algorithms that address some metrics such as reliability, availability, bandwidth consumption, storage usage, response time. In this paper, we present different issues involved in data replication and discuss the key points of the recent algorithms with a tabular representation of all those features. The focus of the review is the existing algorithms of data replication that are based on the meta-heuristic techniques. This review will enable the readers to see that previous studies contributed response time to the data replication, but the contribution of the energy consumption and security improvement has not been considerable well. Moreover, the impact of meta-heuristic algorithms on data replication performance is investigated in a simulation study. Finally, open issues and future challenges have been presented in this research work.

Keywords Data replication · Cloud computing · Data grid · Meta-heuristic

1 Introduction

In some scientific application areas like earth observations, huge amounts of data are becoming a significant part of the shared resources. Such large-scale datasets are usually distributed in different data centers. Data replication technique is usually applied to manage large data in a distributed manner. The replication algorithm tries to store multiple replicas to achieve efficient and fault-tolerant data access in the systems (Mansouri et al. 2019; Dinesh Reddy et al. 2019). Although data management has been previously investigated (Mansouri 2014; Alghamdi et al. 2017; Pitchai et al. 2019; Mansouri and Javidi 2018a), very few of the available algorithms take a holistic view of the different costs and benefits of replication. Many of them usually adopt replication process to enhance data availability and efficiency. These metrics are improved when the number of copies in the system increases. But, the most critical fact they ignored is that data replication leads to energy consumption and financial costs for the provider. Consequently, introducing a data replication technique that considers the balancing of a variety of trade-offs is necessary. In recent times, optimization is a booming field to determine an optimal solution for complex problems. One of the well-known economical techniques for solving optimization problem is meta-heuristic approach due to its (1) simplicity of implementation, (2) prevention of local optima, and (3) independent of problem specific. Consequently, researchers have focused on meta-heuristic technique to solve replication questions (Nanda and Panda 2014; El-Henawy and Abdelmegeed 2018).

The highlighted contributions of the review are:

- Presenting the main challenges of data replication algorithms in the distributed environments.
- Comparing the meta-heuristic-based data replication techniques according to the some attributes like security, energy consumption, bandwidth consumption, performance, storage usage, response time, etc.

Communicated by V. Loia.

✉ Najme Mansouri
najme.mansouri@gmail.com

Mohammad Masoud Javidi
javidi@uk.ac.ir

[1] Department of Computer Science, Shahid Bahonar University of Kerman, Kerman, Iran

- Implementing a data replication algorithm based on several meta-heuristic techniques, i.e., GA, ACO, FA, BA, PSO, WOA, GWO, and ABC to investigate the impact of meta-heuristic techniques on the simulation results.
- Identifying some open research problems in the field of data replication and management in cloud and grid environment.

The rest of the survey is structured as follows. In Sect. 2, a background of cloud computing has been presented. Section 3 covers data replication classification and main challenges in data management. Section 4 introduces the main concept of meta-heuristic techniques. Section 5 presents some challenges motivating meta-heuristics use. Section 6 discusses about meta-heuristic-based replication algorithms in grid and cloud environments. Section 7 summarizes the reviewing results. Section 8 evaluates the performance of data replication algorithms based on different meta-heuristic techniques. Section 9 presents some of the main research challenges and future works. Finally, the conclusion is given in Sect. 10.

2 Background of cloud computing

In this section, we firstly introduce the historical evolution of cloud computing and then we present the main technologies of cloud environment. Finally, the comparison of service models in cloud is discussed.

2.1 History and emergence of cloud computing

In cloud computing, the term "cloud" is applied as a metaphor for "the Internet," so the phrase cloud computing means a type of Internet-based computing. In addition, another reason for representing network infrastructures by an iconized "cloud" is hiding the complexity of the facilities from users (Shojaiemehr et al. 2018). The additional terms that come with "cloud" show the scope of that "cloud," and it can be, for instance, mobile computing and networking. The historical evolution of cloud computing since 1960s to 2011 is presented in Table 1 (Moura and Hutchison 2016; Nadh Singh and Raja Srinivasa Reddy 2017).

It is obvious that cloud computing evolution is currently related to the increasing big data popularity. Recently, there are interesting research areas such as grids (Mansouri et al. 2011), Internet of Things (IoT), and network functions virtualization (NFV) for future networks with a strong relation to cloud computing.

Various review works on data replication can be found in the literature, as presented in Table 2. The novelty of the work we present here, in relation to other surveys, is to focus on a study on meta-heuristic techniques for data replication in data grid and cloud commuting environments.

2.2 Foundations of cloud computing

Main foundation elements of cloud computing are virtualization, distributed computing like grid, Internet, and network management (Mansouri and Javidi 2019). Figure 1 indicates the common elements of cloud environment. Virtualization has a main role in cloud computing and can offer significant benefits such as maximizing resources, flexibility, availability, scalability, hardware utilization, load balancing, and security. In addition, cloud computing system needs reliable access, management automation, and self-service provisioning. The major complexity of cloud system is related to the management automation that automatically optimizes resource usage and adapts based on the dynamic demands of users and system status (Michael et al. 2010).

2.3 Cloud computing service models

Services of cloud have three specific features that differentiate them from traditional hosting: (1) on-demand service which means resources are made available to the user on an "as-needed" basis; (2) rapid elasticity that refers to the ability of a system in adapting to workload changes by increasing or decreasing the amount of system capacity (e.g., CPU, storage, and input/output bandwidth); and (3) self-service that means customers are able to provision cloud computing resources without interaction with service providers (Aznoli and Jafari Navimipour 2017). Generally, there are three service models to describe cloud services (see Fig. 2): Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS).

2.4 Cloud architecture

Cloud computing is not a completely new concept; it is based on several other computing areas such as utility computing and cluster computing. Table 3 gives an end-to-end comparison in various perspectives (Hashemi and Khatibi Bardsiri 2012).

Foster et al. (2008) presented the four-layer structure of cloud computing in comparison with the traditional five-layer grid architecture (see Fig. 3).

- The main responsibility of fabric layer is managing low-level resources such as computing units, storage, and network bandwidths.

Table 1 Historical evolution of cloud environment

Year(s)	Project/organization	Main achievement
1960s	IBM	Technology of mainframe time sharing
1995	MicronPC	Provider of web services for small users and organizations
1999	Salesforce	Enterprise-level applications
2002	Amazon	An online marketplace (Mechanical Turk)
2006	Amazon	Cloud infrastructure service (Elastic Compute Cloud-EC2)
2007	Academic Cloud Computing Initiative	Students can explore the new potential cloud capabilities
2007	Google	Google Docs
2008	Eucalyptus, OpenNebula	Open-source toolkits for cloud management
2010	Microsoft	Windows Azure with focusing on cloud
2011	IBM	The Smarter Computing framework that includes cloud computing as a relevant tool

Table 2 Data replication-surveyed contributions

References	Year	Environment	Main goal
Amjad et al. (2012)	2012	Grid	A comprehensive discussion on dynamic replication in grid according to their nature
Kingsy Grace and Manimegalai (2014)	2014	Grid	Present a comparison among various replica placement and selection algorithms
Tos et al. (2015)	2015	Grid	Study the impact of grid architecture on the performance of data replication
Hamrouni et al. (2016)	2016	Grid	Discuss replica selection algorithms based on data mining techniques
Malik et al. (Tziritas et al. 2015)	2015	Cloud	Characterize data replication algorithms that tackle the resource usage and QoS provisioning
Alami Milani and Navimipour (2016)	2016	Cloud	Present the taxonomy of data replication algorithms (i.e., static and dynamic) and highlight their features

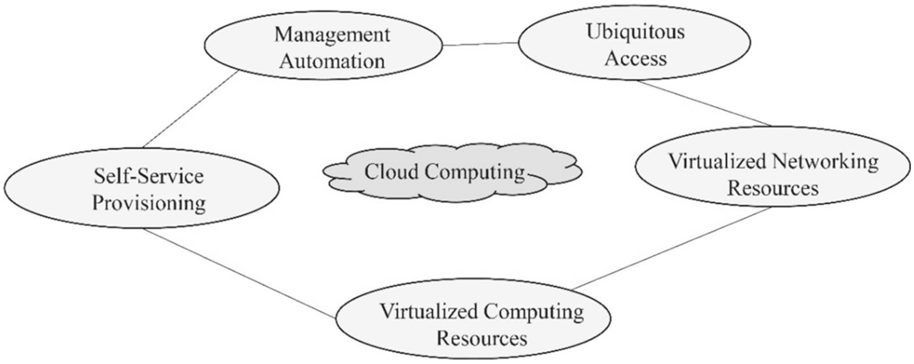

Management Automation
Ubiquitous Access
Self-Service Provisioning
Cloud Computing
Virtualized Networking Resources
Virtualized Computing Resources

Fig. 1 Main features of cloud computing

- The platform layer presents a development and running platform for cloud workflows and controls middleware services, scheduling service, programming languages, and so on.
- The unified resource layer is composed of both software services and hardware services that are necessary for executing cloud applications.
- The final layer consists of cloud workflows like social networking tools that operate within virtual environment.

All resources such as storage are shared among the tenants in cloud environments, and so an elastic management of these resources leads to the satisfaction of both tenant requests and the profit of provider. In addition, quality of service (QoS) requirements are very critical for the dynamic nature of the cloud environment and for this regard a service-level agreement (SLA), a legal contract between the tenant and the provider, is defined (Limam et al. 2019). The general overview of SLA is presented in Fig. 4 (Aljoumah et al. 2015).

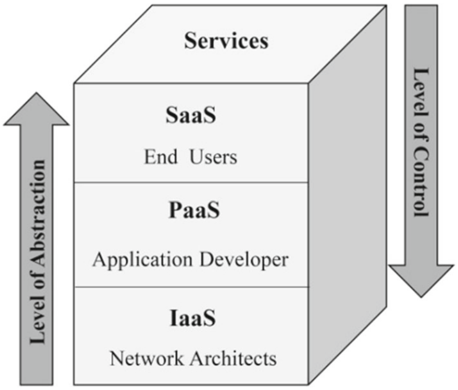


Fig. 2 Service models in cloud

Table 3 Comparisons of grid and cloud

Parameter	Grid	Cloud
Goal	Resource sharing	Present anything as a service
Abstraction	Low	High
Time to run	Not real time	Real time
Failure management	Limited	Strong
User-friendly	Low	High
Number of users	Few	More
Transparency	Low	High
Resource management	Distributed	Centralized/distributed

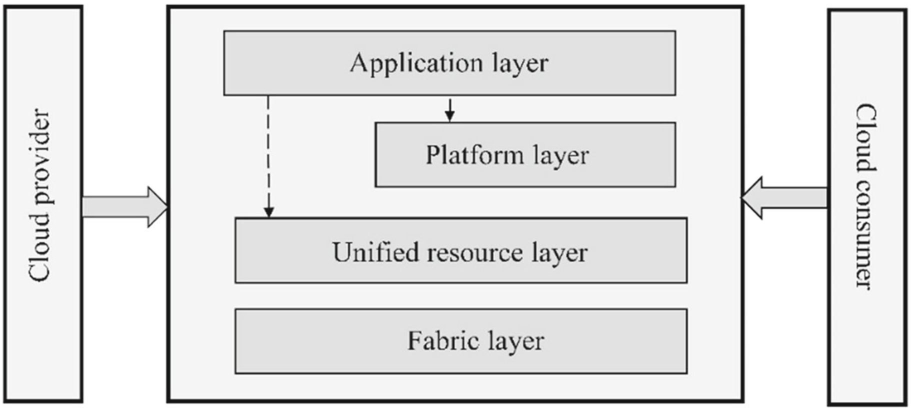
3 Data replication

Nowadays, large amount of data is being created and processed by many scientific and engineering researches like bioinformatics, earth science, and astronomy. In dynamic and distributed environments such as cloud and grid, many challenges revolve around data management and data transfer. Data replication is a general technique to overcome these challenges and enhance the performance of system and reliability (Mansouri 2016; Tos et al. 2018; Mansouri and Dastghaibyfard 2013; Boru et al. 2015). Mansouri and Buyya (2018) discussed about a vital role obtaining the cheapest network and storage resources in suitable time with considering various prices of storage types and time-varying workload. To optimize the costs of replica creation, storage, and potential migration, they investigated the following main questions. (1) Which storage type should host the replica (i.e., placing), (2) which replica should select for task execution, and (3) when the replica should be migrated from a storage to another one.

Generally, data replication algorithms are categorized into two types: static replication and dynamic replication as shown in Fig. 5. Static replication algorithms predetermine the location of replicas. Therefore, there are no more copies generated or adjusted. One of the main benefits in static replication strategy is low overhead during execution (Mansouri et al. 2017), while dynamic replication algorithm dynamically generates and removes replicas based on the access pattern and system status. Dynamic replication algorithms are suitable for dynamic environments like cloud. Nevertheless, frequent huge data transfer that is a consequence of dynamic method leads to a strain on resources of network. Hence, a well-defined data replication algorithm that can create replicas in appropriate time and location is essential to avoid unnecessary replication.

Static and dynamic replication techniques can be categorized further into categories as distributed and centralized strategies (Sun et al. 2012; Mansouri and Javidi 2018b). In centralized replication techniques, a central element manages all aspects of data replication. All necessary information is either collected or distributed by this central element. The main problem of the centralized method is if the nodes of distributed system frequently enter and leave, then the overload of central decision element significantly grows. On the other hand, in the decentralized method, no central element exists to keep the information of system and so nodes themselves decide about data replication. The main issue of the decentralized manner is the synchronization process. Hence, designing a

Fig. 3 Cloud architecture



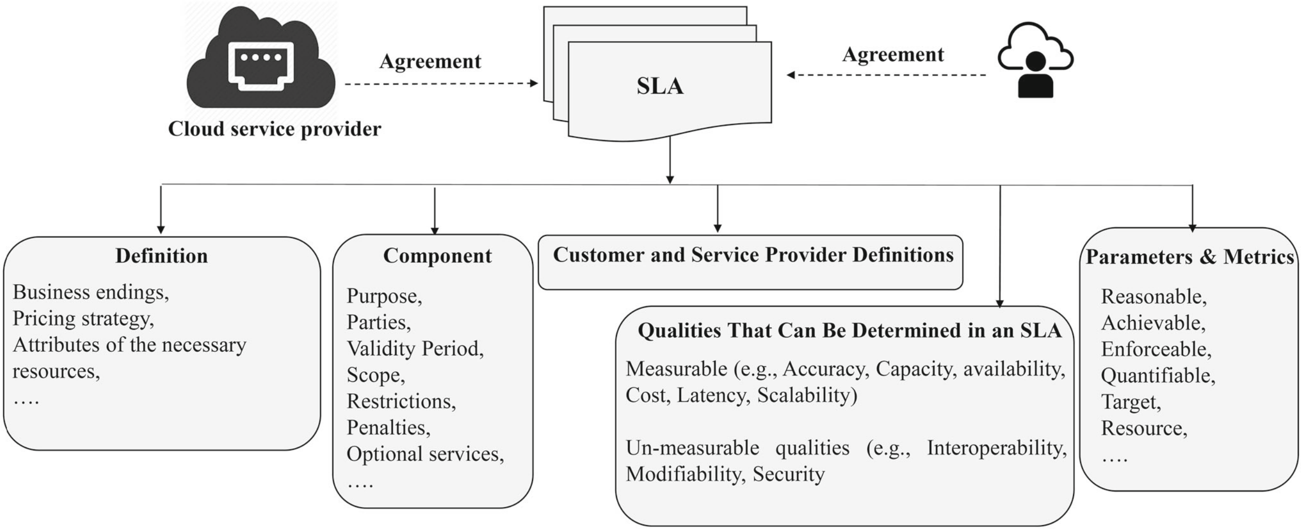

Cloud service provider
Agreement
SLA
Agreement
Definition
Business endings,
Pricing strategy,
Attributes of the necessary resources,
….
Component
Purpose,
Parties,
Validity Period,
Scope,
Restrictions,
Penalties,
Optional services,
….
Customer and Service Provider Definitions
Qualities That Can Be Determined in an SLA
Measurable (e.g., Accuracy, Capacity, availability, Cost, Latency, Scalability)
Un-measurable qualities (e.g., Interoperability, Modifiability, Security
Parameters & Metrics
Reasonable,
Achievable,
Enforceable,
Quantifiable,
Target,
Resource,
….

Fig. 4 SLA main concepts

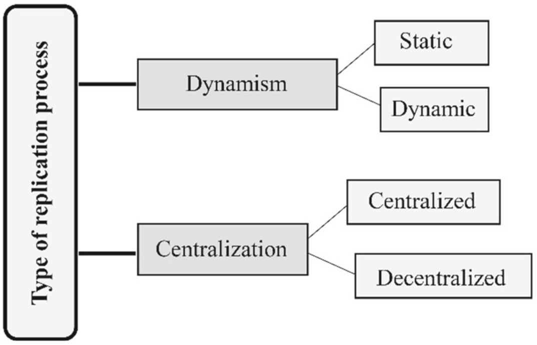


Fig. 5 Types of replication process

data replication algorithm, which adapts to the environment and requests of user, becomes a critical issue, especially as datasets of interest tend to be large.

3.1 Main challenges in data replication

Data replication process reveals the following challenges:

- *Replica decision* Which data should be replicated to meet the requirements of users for reducing waiting time and data access time? It is not necessary to create replicas for all files, particularly for nonpopular files.
- *Number of replicas* How many suitable new replicas should be generated to provide the reasonable system availability? When the number of new replicas is increased, then cost of the system maintenance will significantly increase. Moreover, too many replicas may not enhance system availability, but lead to the waste of resources in the network.
- *Replica placement* Where the new replicas should be stored to reduce job execution time and network latency? It is hard to select optimal locations so that the workload of system is balanced.
- *Replica selection* Which replicas should be selected to access the required data for job execution? One of the vital parameters in replica selection is response time.
- *Replica replacement* Which replicas should be deleted to provide sufficient storage space for new replica? Usually, the algorithm must check the benefits of storing the new replica with the cost of deleting the files.
- *Replica consistency* When a replica changes, how to make other copies and original file quickly consisting? It is necessary to ensure synchronous update when modifying for any replica.

One of the main factors that play a main role in replication process is the architecture of environment. Generally, there are four architectures: (1) multitier, (2) hybrid, (3) P2P, and (4) general graph, as shown in Fig. 6 (Tos et al. 2015). In Tos et al. (2015), the analyzing simulation results based on response times proved that data replication algorithms were performed best in general graph architecture.

4 Meta-heuristic techniques

In the last 30 years, an approximate technique has been introduced that combines standard heuristic strategies with higher-level frameworks to explore a search space, effectively. These techniques are nowadays named meta-heuristic. Meta-heuristic techniques have been widely conveyed in the literature (i.e., algorithms, applications, comparisons, and analysis) due to simplicity and flexibility

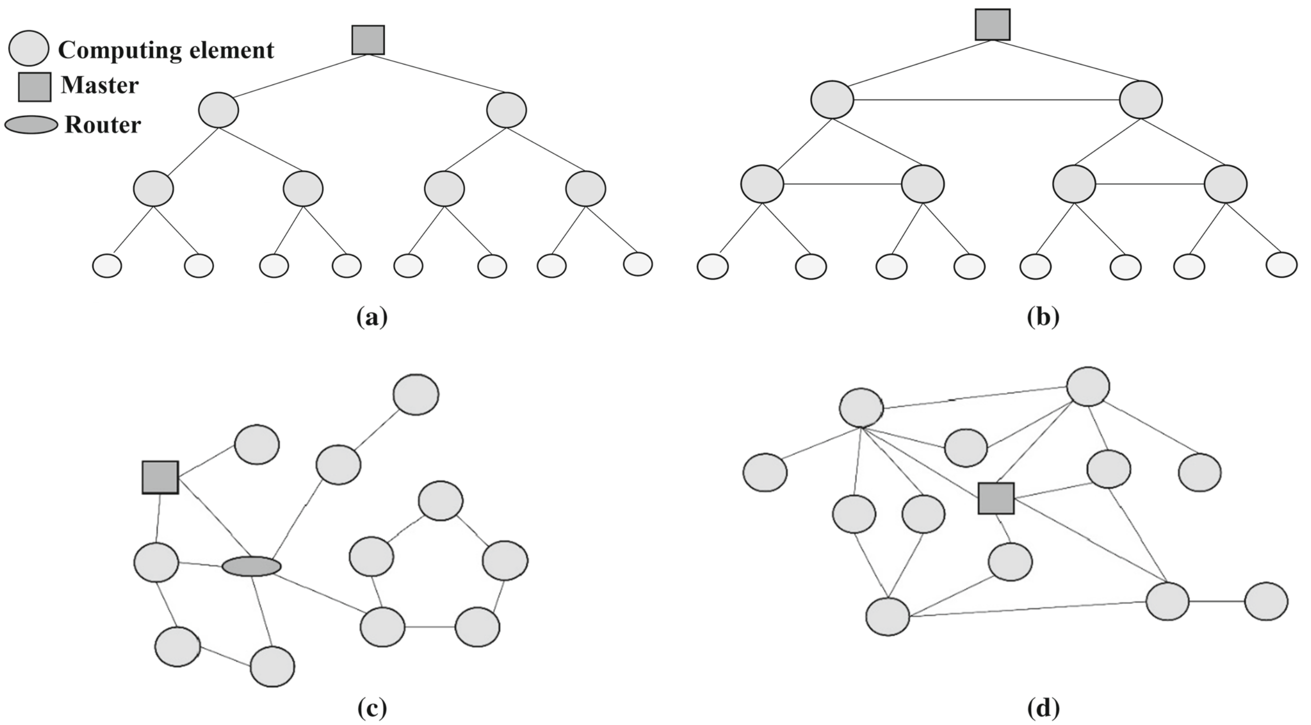


Fig. 6 Architectures considered in evaluations. **a** Multitier, **b** hybrid, **c** P2P, **d** graph (Tos et al. 2015)

(Singh Kushwah et al. 2018; Ebrahimzade et al. 2020; Mirzai et al. 2017; Farzampour et al. 2019; Ebrahimzade et al. 2018). Four advantages of meta-heuristic algorithms are as follows:

(1) *Broad applicability* Meta-heuristic algorithms are not problem specific and can be used to any problems that can be formulated as a function optimization. Hussain et al. (2019) sorted different fields that meta-heuristic techniques are used based on the number of publications during the period of 1983–2016. Table 4 indicates that meta-heuristic algorithms mostly applied on numerical problems (e.g., continuous and discrete, constrained, and unconstrained, etc.), and data mining (e.g., optimization in classification, prediction, and clustering, etc.). Other applications among top ten fields are communications (e.g., networking, and telecommunication), transportation (e.g., routing), engineering (e.g., mechanical designs), civil engineering (e.g., bridge design), and information and communications technology—ICT (e.g., cloud and grid computing).

(2) *Hybridization* Meta-heuristic algorithms can be combined with traditional approaches. Raidl and Roli (Dokeroglu et al. 2019) presented a review on the strategies of hybrid meta-heuristics and indicated that combining various algorithms is one of the most successful techniques for optimization problem. However, the process of designing hybrid meta-heuristics is rather complex and needs knowledge about a broad spectrum of algorithmic concepts, programming, and data structures.

(3) *Ease of implementation* They are simple to develop and understand. Meta-heuristic algorithms can solve problems with different objective functions with minor changes in codes.

(4) *Efficiency* They can solve large problems in acceptable time. In other words, they lead to a significant reduction in software development time.

Figure 7 indicates an outline of how conceptual meta-heuristic algorithm works (Tsai and Rodrigues 2014). It consists of initialization, transition, evaluation, and determination operators that are performed in a number of iterations. Firstly, it randomly creates initial solutions like the greedy and deterministic strategies. Secondly, it performs a transition operator to modify the current solution and can keep partial good subsolutions. Thirdly, it uses the evaluation operator to evaluate the solution based on the value of objective function in the optimization problem. Fourthly, it considers the determination operator to guide the search directions to move toward better solutions. The performance of meta-heuristic algorithm is controlled by this operator. For example, an intensification search leads to converge very quickly but very easy to trap into local optimum in initial iterations. On the other hands, a diversification search will not converge to a particular solution very quickly, but the quality of final solution will be good.

Table 4 Applications of meta-heuristic algorithms

Rank	Meta-heuristic applications
1	Numerical problems
2	Data Mining
3	Electrical and Electronics
4	Scheduling
5	Combinatorial Optimization Problems
6	Communications
7	Transportation
8	Engineering Design Problems
9	Civil Engineering
10	ICT
11	Business, Finance, Economic
12	Image Processing
13	Oil and Energy
14	Emergency and Disaster Management
15	Control Engineering
16	Agriculture
17	Biology
18	Medical
19	Water Management
20	Chemical Process Engineering
21	Manufacturing and Production
22	Civil Aviation
23	Mining
24	Traffic Control
25	Military and Defense
26	Music

```
Input data
Create initial solution
While the termination condition not reached do
{
    Transition ()
    Evaluation ()
    Determination ()
}
Output solution
```

Fig. 7 Overall framework of meta-heuristic algorithm

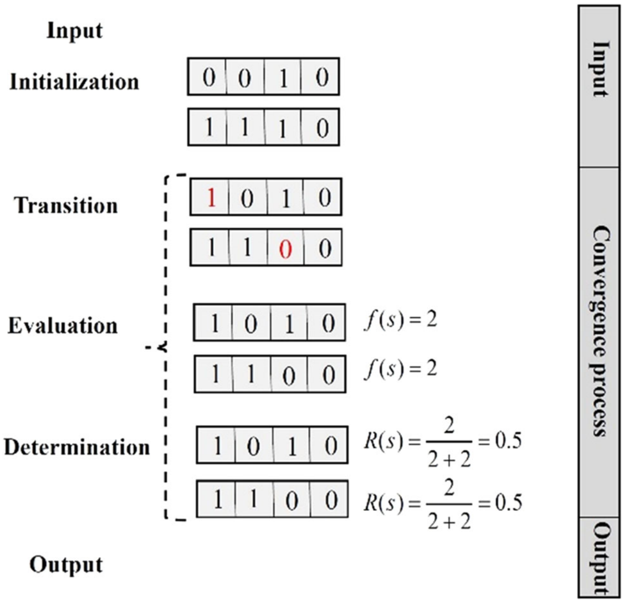


Fig. 8 Example of meta-heuristic procedure

Figure 8 shows an example that illustrates how meta-heuristic algorithms work in general (Tsai et al. 2015). Consider the population size is two and each solution has four subsolutions. Firstly, two solutions are randomly generated. Then, the solutions are transformed by the transition process. After that, the evaluation operator evaluates the fitness of solutions and determination operator selects suitable solutions for the next iteration. Finally, after the termination criterion is satisfied, the final solution obtained is considered as an approximate solution or the optimal solution.

Most of the state-of-the-art meta-heuristic algorithms (classical) such as genetic algorithms (GA) (Goldberg and Holland 1988), particle swarm optimization (PSO) (Eberhart and Kennedy 1995), ant colony optimization (ACO) (Dorigo et al. 2006) have been introduced before the year 2000. Nevertheless, new evolutionary algorithms such as artificial bee colony (ABC) (Basturk and Karaboga 2006), firefly algorithm (FA) (Yang 2009), whale optimization algorithm (WOA) (Mafarja and Mirjalili 2018), bat algorithm (BA) (Yang 2010), and gray wolf optimization (GWO) (Mirjalili et al. 2014) also are developed successfully in the last two decades. In many cases, these new meta-heuristic algorithms achieve the best solutions for some of the benchmark problems. Figure 9 represents the search result of the number of publications for some classical and new generation meta-heuristic algorithms on Google Scholar web site (in May 2019). We can see that GA and ABC have the largest number of papers.

Cloud computing has become a buzzword in the scope of high-performance distributed system since its unique features such as on-demand self-service, compatibility, and elasticity. To achieve its full advantages, much research is necessary across broad areas (Mansouri et al. 2019). One of the main challenges, which must be focused for its efficient performance, is data replication. Meta-heuristic-based algorithm can determine near-optimal solutions in reasonable time for such hard problems.

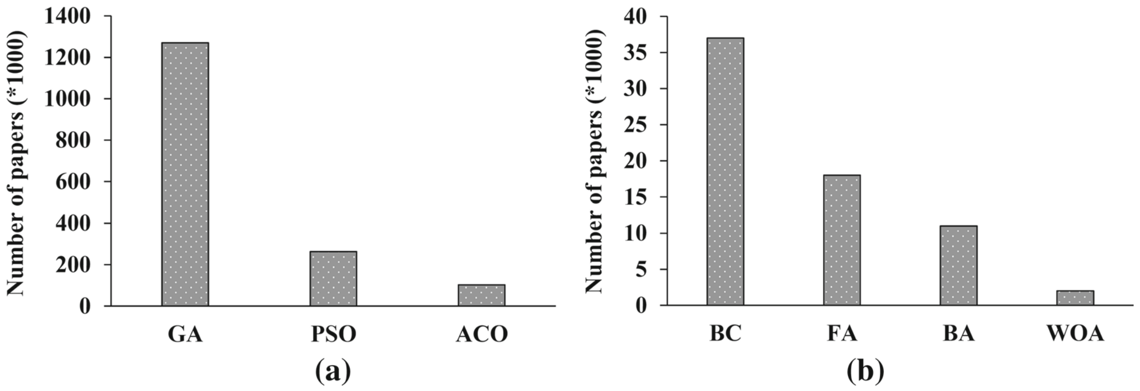


Fig. 9 The number of papers for the classical and new meta-heuristics

5 Motivation

The distributed computing environments have undergone several changes. The emergence of cluster is the early change that connects a set of loosely or tightly systems to work together. Then, grid computing is developed to solve the issues of cluster systems being only able to use local resources. Therefore, grid combines all the available heterogeneous resources from geographically distributed systems. Now, cloud computing is used that leverages the strengths of cluster and grid systems (Foster et al. 2008). Generally, the vision for cloud and grid is similar and they are struggling with many of the same issues. Nevertheless, the details and technologies used may differ. Data replication is one the common problems that are applied in the distributed systems such as P2P and grid systems. In such systems, a data replication algorithm must specify what to replicate? when to create/remove replicas? where to store them?, and how many replicas to store? (Ranganathan and Foster 2001). But, most of the developed data replication algorithms in the above environments are difficult to adapt to clouds since they focus on improving the performance of system without considering cloud goals such as the economic cost. In other words, data replication algorithms for grid environments usually try to present low response time and the most significant point that they neglected is the profit of cloud providers (Mokadem and Hameurlain 2020). In fact, the performances of these systems are improved as the number of replicas increases. However, creating as many replicas as possible in cloud environments cannot be economically feasible since data replication does not come free and it can result in wasteful resource (i.e., storage and energy) and so reduces the provider profit (Long et al. 2014). Therefore, designing data replication techniques for cloud should also take into account other points such as: (1) considering a reliable quality of service (QoS) according to the service-level agreement (SLA) (Mokadem and Hameurlain 2020) and (2) adjusting the cloud resource based on the pay as you go' pricing model (Pitchai et al. 2019). It is obvious that proposing a dynamic data replication algorithm for cloud should be involved balancing a variety of trade-offs. For example, the response time decreases as the number of replicas increases. Nevertheless, the number of replicas should be as small as possible for minimizing the energy.

While the fundamental data replication aspects remain unchanged, various optimization objectives ballooned the replication literature in the past decade. Because of new dynamic properties of cloud (e.g., diversity of user requests, dynamic load on data centers, and changing resource capacity), the traditional data replication cannot work effectively in cloud computing. Users are using the cloud in a dynamic manner, and so, the resource must be automatically allocated and de-allocated to maximize agility. With increasing the number of cloud users and size of files each day, data replication turns into an imperative issue to manage.

Guarantees of performance (e.g., response time) are often not considered as a part of the SLA by providers of cloud because of the heterogeneous workloads in cloud environment. For instance, Google Cloud SQL only offers error guarantees without response time guarantee (Mansouri and Javidi 2018a). So the "tenant performance/provider profitability" trade-off should be solved for data replication algorithms in cloud especially when they are proposed for OLAP (online analytical processing) applications. As a consequence, a data replication algorithm seeks to find the near-optimal solutions by balancing the trade-offs among the several optimization objectives [e.g., availability (Wei et al. 2010), reliability (Li et al. 2017), low latency (Ma and Yang 2017), data durability (Liu et al. 2018), security (Ali et al. 2018a) and energy efficiency (Séguéla et al. 2019), availability, load balancing (Long et al. 2014), and cost (Limam et al. 2019)].

Meta-heuristic techniques are preferred to effectively deal with its complexity in the data replication of grid and cloud, and accordingly several algorithms have been introduced. Since meta-heuristics apply iterative processes to find solutions in a reasonable time and can provide better results than traditional methods (Shojaiemehr et al. 2018). Consequently, meta-heuristic technique plays a vital role for data replication problem in the distributed environment. To the best of our knowledge, despite the importance of meta-heuristic-based data replication algorithms, there is not any detailed and comprehensive review of these techniques. The aim of this study is to review the available strategies and compare the differences among the explained algorithms.

6 Meta-heuristic-based data replication algorithms

In this section, we classify the dynamic replication algorithms into different categories with respect to the meta-heuristic techniques. There are many performance criteria for cloud computing, so it is infeasible to cover all aspects in one paper. In this review, the scope is narrowed to main parameters related to data-intensive systems. Table 5 indicates a summary of meta-heuristic-based replication algorithms (between the periods of 2006 and 2019) for grid and cloud environments. The sign (+) shows that the replication algorithm considers the concerned option, and the sign (−) indicates that the replication algorithm does not focus on that option. Finally, the sign (NM) indicates that the option is not mentioned in the paper.

6.1 Data replication strategies based on genetic algorithm

The genetic algorithm (GA) is one of the most popular evolutionary techniques that introduced in 1970s by Henry Holland (1992). The standard GA is very generic, and many parts such as solution representation, selection process, crossover, and mutation operators can be developed differently based on the problem. Algorithm 1 indicates the pseudo-code of genetic algorithm.

Cui et al. (2018) proposed a new replica placement strategy based on GA for cloud environment. The authors designed a tripartite graph with three kinds of vertices that represent jobs, replicas, and nodes. The mappings from jobs to replicas are indicated by the edges between job and replica vertices. This mapping is used in scheduling step. The mappings among replicas and nodes are represented by the edges between replica and node vertices. This mapping determines the locations of data replicas. The proposed strategy can determine the mappings with the best performance as a near-optimal solution by using genetic algorithm. The performance evaluation indicated that the proposed replica placement strategy has lower data transmission time than random data placement algorithm applied in Hadoop distributed file system (HDFS) (Shvachko et al. 2010).

Junfeng and Weiping (2016) presented a pheromone-based genetic algorithm for selection strategy to enhance the performance of cloud system. The proposed strategy determines the probabilistic information of replica in cloud by virtue of group constant overlapping, realization of mutation operators, and the feedback information. Genetic algorithm indicates two main benefits in replica selection step. The first advantage is that genetic algorithm finds the optimal solution at end. The second one is that it is a distributed optimization technique and so can be adapted to the distributed environment. For fitness calculation, the proposed algorithm considers network condition and transmission time. The simulation results demonstrated that the presented algorithm leads to good performance in terms of mean job execution time.

Junfeng and Weiping (Al Jadaan et al. 2010) enhanced replica selection process in data grid to ensure the satisfaction of the users. The proposed strategy considers availability to download necessary data from that node even if there are some difficulties such as malfunctioning and degradation in network. The authors combined various parameters (i.e., response time, availability, and security) that are not aggregated with each other in replica selection.

Algorithm 1. Genetic Algorithm

```
1 Initialization: Create initial population of chromosomes.
2 Fitness: Compute fitness value for each chromosome based on the fitness function.
3 Selection: Select the best chromosomes for reproduction.
4 Crossover: Apply crossover operator on the chromosomes determined in step 3.
5 Mutation: Apply mutation operator on the chromosomes.
6 Fitness: Evaluate the fitness of these newly created chromosomes known as offsprings.
7 Replacement: Replace least-fit population with better chromosomes from offsprings.
8 Repeat steps 3 to 7 until termination condition is met.
9 Output: Return the best chromosome as a solution.
```

Table 5 Comparison of data replication algorithms-cont'd

Strategy	Cui et al. (2018)	Junfeng and Weiping (2016)	Al Jadaan et al. (2010)	Almomani and Madi (2014)
Year	2015	2016	2010	2014
Environment	Cloud	Cloud	Grid	Grid
Type	Dynamic	Dynamic	Dynamic	Dynamic
Architecture	Graph	Graph	NM	NM
Replica decision	–	–	–	–
Replica placement	+	–	–	+
Replica selection	–	+	+	–
Replica replacement	–	–	–	–
Replica consistency	–	–	–	–
Number of replicas	–	–	–	–
Heuristic technique	GA	GA	GA	GA
Availability	–	–	+	–
Security	–	–	+	–
Load balancing	–	–	–	+
Response time	+	+	+	+
Storage usage	+	–	–	+
Bandwidth consumption	+	+	+	–
Energy Consumption	–	–	–	–
Replication cost	–	–	–	–
Algorithm evaluation	Java code	OptorSim	NM	NM
Experiment size	10–30 jobs, 5–13 nodes	10 nodes	0–100 requests	NM
Compared with	Random strategy of HDFS	Ant colony, No replication	Round robin, random	NM
Main idea	Design tripartite graph model	Use probabilistic selective formula	Consider security factor	Consider site throughput

Strategy	Grace et al. (2014)	Xu et al. (2015)	Liu et al. (2017)	Wu (2017)
Year	2015	2015	2017	2017
Environment	Grid	Cloud	Cloud	Cloud
Type	Dynamic	Dynamic	Dynamic	Dynamic
Architecture	P2P	NM	P2P	Graph
Replica decision	–	–	–	–
Replica placement	–	+	+	+
Replica selection	+	–	–	–
Replica replacement	–	–	–	–
Replica consistency	–	–	–	–
Number of replicas	–	–	–	+
Heuristic technique	GA	GA	GA	GA
Availability	+	–	–	–
Security	+	–	–	–
Load balancing	+	–	+	–
Response time	+	+	+	+
Storage usage	–	–	–	+
Bandwidth consumption	–	–	+	+
Energy consumption	–	–	–	–
Replication cost	–	–	–	+
Algorithm evaluation	GridSim	Real environment	Java code	Real environment
Experiment size	0–20 requests	0–20 nodes	5–13 nodes	40 nodes

Table 5 (continued)

Strategy	Grace et al. (2014)	Xu et al. (2015)	Liu et al. (2017)	Wu (2017)
Compared with	ACO	Monte Carlo	Random, k-means	NREP, RAND, and MCRP
Main idea	Compare genetic with ant colony	Consider number of file access	Group high similarity tasks together	Define cost models

Strategy	Chunlin et al. (2019)	Zhang et al. (2014)	Huang and Wu (2018)	Wang et al. (2013)
Year	2019	2014	2018	2013
Environment	Cloud	Cloud	Cloud	Cloud
Type	Dynamic	Dynamic	Dynamic	Dynamic
Architecture	Multitier	NM	NM	NM
Replica decision	–	–	–	–
Replica placement	+	+	+	–
Replica selection	–	–	–	+
Replica replacement	–	+	–	–
Replica consistency	+	–	–	–
Number of replicas	+	+	+	–
Heuristic technique	GA	GA	GA	ACO
Availability	+	–	–	–
Security	+	–	–	–
Load balancing	+	–	–	–
Response time	+	+	+	+
Storage usage	+	+	+	–
Bandwidth consumption	+	–	–	+
Energy consumption	–	–	–	–
Replication cost	–	–	–	+
Algorithm evaluation	Java	CloudSim	MATLAB	NM
Experiment size	1–10 nodes, 100–500 jobs	NM	10–50 nodes	NM
Compared with	DRAS, ARDS	–	GA	–
Main idea	Use migration model	Establish p-center model	Define data support degree	Consider pricing models

Strategy	Sun et al. (2005)	Yang et al. (2013)	Jafari Navimipour and Alami Milani (2016)	Shojaatmand et al. (2011)
Year	2005	2013	2016	2011
Environment	Grid	Grid	Cloud	Grid
Type	Dynamic	Dynamic	Dynamic	Dynamic
Architecture	P2P	P2P	Graph	P2P
Replica decision	–	–	–	–
Replica placement	–	–	–	–
Replica selection	+	+	+	+
Replica replacement	–	–	–	–
Replica consistency	–	–	–	–
Number of replicas	–	–	–	–
Heuristic technique	Ant colony	Ant colony	Ant colony	Ant colony
Availability	–	–	–	–
Security	–	–	–	–
Load balancing	+	+	–	–
Response time	+	+	+	+
Storage usage	–	–	–	–

Table 5 (continued)

Strategy	Sun et al. (2005)	Yang et al. (2013)	Jafari Navimipour and Alami Milani (2016)	Shojaatmand et al. (2011)
Bandwidth consumption	+	+	+	+
Energy consumption	–	–	–	–
Replication cost	–	–	–	–
Algorithm evaluation	OptorSim	OptorSim	Java code	OptorSim
Experiment size	NM	100–3000 jobs, 11 nodes	5000 nodes	500–2000 jobs
Compared with	No replication	No replication	RTRM	No replication
Main idea	Consider disk I/O transfer	Consider available bandwidth	Consider read accessibility of file	Consider file size

Strategy	Khojand et al. (2018)	Khalili Azimi (2019)	Taheri et al. (2013)	Salem et al. (2019)
Year	2018	2019	2013	2019
Environment	Grid	Cloud	Grid	Cloud
Type	Dynamic	Dynamic	Dynamic	Dynamic
Architecture	Multitier	Multitier	P2P	P2P
Replica decision	–	–	–	+
Replica placement	+	+	+	+
Replica selection	–	+	–	+
Replica replacement	+	+	–	–
Replica consistency	–	–	–	–
Number of replicas	–	–	+	+
Heuristic technique	ACO	ABC	ABC	ABC
Availability	–	–	–	+
Security	–	–	–	–
Load balancing	–	–	+	+
Response time	+	+	+	+
Storage usage	+	–	+	+
Bandwidth consumption	–	+	+	+
Energy consumption	–	–	–	–
Replication cost	–	–	–	–
Algorithm evaluation	NM	MATLAB	NM	CloudSim
Experiment size	100 nodes	100–1500 jobs	5–40 nodes and 9652 jobs	1–15 nodes, 1000 jobs
Compared with	PHFS, FS, Cascading	LRU, LFU, and BHR	DIANA, FLOP MinTrans, MinExe	GA, DCR2S, EFS
Main idea	Apply fuzzy system	Consider three level structure	Combine with job scheduling	Consider the knapsack problem

Strategy	Sadeghzadeh and Navaezadeh (2014)	Jayasree and Saravanan (2018)	Ebadi and Jafari Navimipour (2018)	Muñoz and Carballeira (2006)
Year	2014	2018	2018	2006
Environment	Grid	Cloud	Cloud	Grid
Type	Dynamic	Dynamic	Dynamic	Dynamic
Architecture	NM	Multitier, Flat, Dcell (Bilal et al. 2013)	Multitier	P2P
Replica decision	–	–	–	–
Replica placement	+	+	+	+
Replica selection	–	–	+	–

Table 5 (continued)

Strategy	Sadeghzadeh and Navaezadeh (2014)	Jayasree and Saravanan (2018)	Ebadi and Jafari Navimipour (2018)	Muñoz and Carballeira (2006)
Replica replacement	–	–	–	+
Replica consistency	–	+	–	–
Number of replicas	–	–	–	–
Heuristic technique	FA	PSO	PSO	PSO
Availability	–	–	–	–
Security	–	+	–	–
Load balancing	–	–	–	–
Response time	+	+	+	+
Storage usage	–	+	+	+
Bandwidth consumption	–	+	–	+
Energy consumption	–	–	+	–
Replication cost	–	–	–	–
Algorithm evaluation	MATLAB	NM	MATLAB	OptorSim
Experiment size	10–150 nodes	NM	10–40 nodes	7 nodes
Compared with	PSO, GA	Greedy, GRA, DROPS	TS, PSO, ACO	NM
Main idea	Consider writing cost	Apply centrality concept	Use tabu search	Consider latency and hit ratio

To overcome this problem, they employed GA-clustering algorithm. Then, the best replica is the one that provides good response time, availability, and a reasonable level of security at the same time. If more than one site shows the best possible combination, then the proposed strategy will randomly choose one of them by default. The experimental results conclusively demonstrated that it is more secure than previous strategies such as round robin (RR) strategy, and at the same time, it is more reliable and efficient.

Almomani and Madi (2014) presented a genetic-based replica placement algorithm to determine the best location for new replicas. The proposed algorithm with optimization technique has two main achievements. The first improvement is that it reduces data access time by considering the read cost of files. The second improvement is that it avoids network congestion by considering the load of sites. The fitness function is defined based on the read cost and storage cost. Therefore, it can determine the appropriate site for replica placement such that it satisfies the user requirement and resource provider.

Grace et al. (2014) used two meta-heuristic strategies (i.e., genetic and ant colony) to select appropriate location from many available sites. The main parameters in ant colony-based algorithm are response time and the size of file. In genetic-based algorithm, response time, data availability, security, and load balancing are considered. The simulation results with GridSim indicated that the efficiency of genetic algorithm is 30% more than the ant colony strategy.

Xu et al. (2015) introduced a novel data placement algorithm based on GA to improve data access in cloud environment. Firstly, the authors defined a mathematical model for data scheduling in cloud system and then used the fitness function based on the number of data access to compute the fitness of each individual in a population. They employed roulette-wheel selection technique for choosing the individuals with high fitness value. The experimental results indicated that genetic algorithm could find a near-optimal data placement matrix in an acceptable time and performance result is better than Monte Carlo strategy (Shijie et al. 2010).

Liu et al. (2017) presented a new replica placement strategy by GA for cloud computing. The proposed strategy considers two main facts. The first one is that the dependency between initial files is important in reducing the data transmission time. It stores files with high dependency together, and so data movement among data centers is decreased. The second fact is that the transmission cost is related to the file size. In addition, there is much probable that a task requires small input files but creates large files.

Therefore, if the created files are necessary for other tasks in various data centers, then the large amount of data must be moved. In summary, the proposed strategy after constructing the data interdependency matrix uses the genetic algorithm to determine the best replica location. It uses the total transmission time as fitness function. The experimental results proved that the proposed algorithm could reduce the size of data movement compared to the *K*-means algorithm (Yuan et al. 2010).

Wu (2017) proposed a replica placement strategy from cost-effective view in the cloud system. The author discussed about the trade-off between dataset's real replicas and pseudo-replicas (i.e., the replica is not constant in the nodes, but is created by its predecessors immediately if necessary). The proposed strategy includes two main processes. First, it introduces a cost model for deciding about number of replicas and their places. The presented cost model comprises of important factors such as storage cost, data transfer cost, and data computation cost for making acceptable cost estimation. Then, it analyzes a cost–benefit concept for data management. Finally, it provides real and pseudo-replicas technique that minimizes data management cost using generic algorithm. The evaluation results indicated that the proposed strategy compared with no replication (NREP) scheme, random (RAND) strategy, and minimum cost replica placements strategy (MCRP) (Wu 2016) is very cost-effective.

Chunlin et al. (2019) proposed a dynamic multiobjective optimized replication algorithm to improve system performance. The proposed algorithm takes into account the file unavailability, the load of data center, and cost of network transmission. The load of data center is determined based on the CPU capability, memory, disk space, and the bandwidth of network. Then, it uses the fast nondominated sorting genetic algorithm to solve the multiobjective optimized replica placement problem. It considers the binary coding scheme and determines the location where the replica is stored and the number of replicas. Moreover, it performs the replica consistency based on the reliability record table to guarantee the data availability. The proposed algorithm applies the replica migration strategy for the file access hot spots problem that is appeared by burst requests. The experimental results presented that the proposed strategy could improve the usage rate of network resource in comparison with dynamic replica adjustment strategy (DRAS) (Qu et al. 2016), ARDS algorithm.

Zhang et al. (2014) proposed a replica placement algorithm to improve user access time. The proposed scheme includes two phases. In the first phase, it determines the data center based on the *P*-center model (Rahman et al. 2008). In the second phase, it defines the specific storage server and disk array of the selected data center based on the creation and deletion cost in the hash function. Finally, the optimal replica placement solution is determined by genetic algorithm. It defines a 0–1 vector $< x_1, x_2, \ldots, x_n >$ to show the solution of genetic algorithm, and the parameter x_j indicates whether the data center *j* is selected for replica placement or not. The results of experiment indicated that the proposed algorithm could reduce the average access time.

Huang and Wu (2018) proposed a cost-effective replica placement algorithm for read-intensive and write-intensive transactional workload. The proposed algorithm takes into account data storage, updating cost, transmission time, and processing costs in cloud system and tries to determine the number of replicas. Then, it finds appropriate locations for new replicas. Moreover, it considers the user access paths to replicas and the number of servers in the cloud system for definition of objective function. The authors introduced a hybrid genetic algorithm (HGA) and presented a heuristic rule according to the data support degree to define the initial population for optimization. The data support degree is defined as the percentage of the number of requests managed by data center *i,* and then, the number of data replicas is determined based on the data support degree before creating the initial population. The simulation results indicated that the solution quality of HGA is close to the optimal solution under large-scale instances.

6.2 Data replication strategies based on ant colony algorithm

Ant colony algorithm (ACO) is a meta-heuristic approach introduced by Dorigo in 1992 to solve hard problems like the traveling salesman problem (Dorigo 1992). Ant colony algorithms were inspired by the observation of real ant colonies behavior how they forage for food. Algorithm 2 indicates the pseudo-code of ant colony algorithm.

Wang et al. (2013) developed a cost- and time-aware model for data replication that is appropriate for data-intensive service composition in cloud. For cost calculation, the proposed algorithm considers the usage-based model, the package-based, the flat fee subscription-based, and the combination-based pricing models. Then, it uses ant colony optimization to reduce the cost of data-intensive service solution based on the cost of replication and response time during replica selection process. The results of simulation proved that the presented strategy could solve replica selection problem efficiently.

Algorithm 2. Ant Colony

```
1 Initialize the pheromone for each path between tasks and resources.
2 Construction of solution for each ant.
3 Calculate the fitness value for each solution.
4 Replace the optimal solution with solution that has the better fitness than the optimal solution.
5 Update pheromone.
6 Repeat the steps 2 to 6 until termination condition is met.
7 Return the optimal solution.
```

Sun et al. (2005) introduced an ant colony-based replica selection algorithm in data grid environment. The proposed strategy considers important criteria in replica selection process: (1) disk I/O transfer that related to disk seeks times. It is obvious that for decreasing the data retrieval time, lower seek time is better. (2) Condition of network that refers to the available bandwidth for transferring replica. (3) Load of site that contains the requested replica. The results of evaluation proved that the proposed algorithm could reduce average access time, especially in data-intensive environment.

Yang et al. (2013) described a new replica selection strategy based on ant colony optimization to reduce access latency. The authors modeled task node as foraging ants and required files of task as food. Therefore, the replicas that are stored in various sites have different paths to the food. They assigned an eigenvalue to each replica, so the size of eigenvalue shows the possibility of being chosen. Moreover, the eigenvalue of replica modifies based on the various circumstances. The proposed strategy considers important factors such as replica host load and available bandwidth in selection process. The simulation results demonstrated that the proposed algorithm could improve performance in terms of effective network usage and mean execution time.

Jafari Navimipour and Alami Milani (2016) proposed a replica selection using ant colony technique to enhance the performance of cloud system. The authors considered a graph that each node indicates a data center and edges connect them. At the beginning, all ants randomly choose a cloud center since there are no pheromones. After the optimal data file is found by the first ant, then other ants will interest to use the new data center that has the target file. For choosing replica, the ants apply pheromone information that is defined based on the read historical accessibility of the replica and the size of replica. The simulation results indicated that the proposed strategy could reduce more access time compared to RTRM strategy (Bai et al. 2013).

Shojaatmand et al. (2011) used ant colony algorithm for selecting the most suitable replica in data grid environment. They defined two references for each file as logical file name (LFN) and a physical file name (PFN). LFN is independent of where the file is placed and PFN indicates a particular replica of file. The proposed strategy uses the pheromone information to show how good the replica is. Ants get the statistical information of the nodes that have the needed replica during moving in the network. Each ant places the collected information in nodes as a trail, and other ants apply this information in finding the path of better pheromone. The pheromone is defined based on the available bandwidth and the size of file. The simulation results with OptorSim demonstrated that the proposed strategy could reduce more response time compared to the no replication strategy.

6.3 Data replication strategies based on artificial bee colony algorithm

Artificial bee colony (ABC) technique was proposed by Karaboga in 2005 for optimizing numerical problems (Karaboga 2005). This strategy is inspired by the intelligent foraging behavior of honey bees. The bees in a colony are categorized into three groups: employed, unemployed foraging bees, and food sources. The first two groups (i.e., employed and unemployed foraging) search for rich food sources near to their hive. This model contains two main leading modes of behavior: (1) for positive feedback, recruitment of forager bees to rich food sources. (2) For negative feedback, abandonment of poor sources by foragers. Algorithm 3 indicates the pseudo-code of bee colony algorithm.

Algorithm 3. Artificial Bee Colony Algorithm

```
Create initial population with random solution.
Compute fitness value of population.
While (termination condition is  not met)
{
    Choose site for neighborhood search.
    Recruit bees for chosen sites (more bees for best sites).
    Choose the fittest bee from each patch.
    Assign remaining bees to search randomly.
}
```

Khojand et al. (2018) proposed a dynamic data replication based on fuzzy system for data grid environment and called predictive fuzzy replication (PFR). PFR algorithm predicts future needs based on the history usage of files and prereplicates them in the appropriate locations. PFR redefines the balanced ant colony optimization (BACO) strategy (Tharani 2016), which is developed for task scheduling in computing grids. In BACO strategy, each job is considered as an ant that tries to get resources in the system. But, PFR algorithm defines a file as an ant and a node as a resource. The authors designed the fuzzy system to define some general rules that are always true in the system. Since grid is a dynamic system, there is not accurate information at every moment. Therefore, fuzzy inference is a good choice for predicting the behavior of system. The proposed fuzzy system presents the direct relation between usage ratio and node size. In addition, it presents the opposite relation between file size and node level. The experimental results proved that PFR strategy could increase the percentage of the use of the replicas compared with predictive hierarchical fast spread (PHFS) (Mohammad Khanli et al. 2011), fast spread (FS) (Ranganathan and Foster 2001), and cascading algorithms (Ranganathan and Foster 2001).

Azimi et al. (Khalili Azimi 2019) introduced a bee colony-based replication in cloud computing. The authors considered a three-level hierarchical topology. First level is regions that are connected with low bandwidth, and the second level consists of LAN's (local area network) of each region that are connected by higher bandwidth comparing to the first level. The third level includes the sites of each LANs that are connected to each other through a high bandwidth. In the proposed bee colony-based algorithm, if new locations or new food regions show better quality or more nectar, the bee remains in new location and one unit will be added to its trial index. The quality is defined as the probability of existing files in sites. After the searching stage is finished by worker bees, then the best site is selected based on the number of requests for a file. The experiment explained that the introduced replication strategy successfully could reduce mean job time compared to LFU, LRU, and BHR (Park et al. 2003) algorithms.

Taheri et al. (2013) used bee colony optimization algorithm for simultaneous job scheduling and data replication in grid systems. The proposed scheduling algorithm arranges jobs based on the length of job, and so the longest job has higher priority than others. A specific number of positions are assigned to bees for advertising each node. If the benefit of the newly allocated job/bee is higher than older bee, then it is replaced. If the similarity of bees is lower than 80% to the dancing bees, then they will be replaced and so biasing all bees to a specific bee type is avoided. Now various types will be available on the dance floor to advertise a node for more job types. The proposed strategy arranges all files based on the size, and so the largest file has the highest priority with respect to others. For prevention of unnecessary replication, the proposed replication strategy uses the condition of "*ArrUpTimes (k)*/*MinUpTime* (D_x) < 2," where *ArrUpTime* is array of the total upload time of D_x to all its dependent jobs and *MinUpTime* (D_x) indicates the minimum uploading time of D_x stored on any node. The experimental results demonstrated that the proposed strategies could improve transfer time and resource utilization compared to the DIANA (Anjum et al. 2006), Chameleon/FLOP (Sang-Min and Jair-Hoom 2003), MinTrans (Abdi and Mohamadi 2010), and MinExe (Ranganathan and Foster 2002).

Salem et al. (2019) introduced a novel replication strategy based on ABC algorithm. In the first phase, the proposed algorithm tries to handle the least-cost path issue for determining the optimal replica placement and obtaining low cost based on the knapsack problem. The main goal is finding a solution so that the consumer can get and store replicas through the shortest path with a lower cost and provide load balancing in the system. In the second phase, ABC strategy is executed by data centers to find an optimal sequence of data replication and support the best least-cost path. The experimental results presented that the introduced strategy could reduce the data transmission in comparison with dynamic cost-aware re-replication and rebalancing strategy (DCR2S) (Gill and Singh 2016), enhanced fast spread (EFS) (Bsoul et al. 2011), and genetic algorithm (GA).

6.4 Data replication strategies based on firefly algorithm

Firefly algorithm is proposed by Yang (2013) at Cambridge University to deal with multimodal, global optimization problems. Flashing light of fireflies is the main characteristic that shows two major procedures as absorbing the mating partners and informing the potential predators. There are some physical rules in the flashing lights. The landscape of the objective function is defined by the brightness of a firefly. A mutual coupling is triggered between two fireflies after the firefly is allocated within the vicinity of another firefly. The attractiveness is proportionate to the brightness. When distance increases, the attractiveness and brightness decrease. In other words, the less bright firefly will attract to the brighter one. Algorithm 4 indicates the pseudo-code of firefly algorithm.

Algorithm 4. Firefly Algorithm

```
Generate initial population of fireflies
Evaluate light intensity $I_i$ based on the fitness function ($f(x_i)$)
For (i=1; i<number of fireflies; i++)
    For (j=1; j< number of fireflies; j++)
        If ($I_i < I_j$)
            Move firefly i towards j
        Change attractiveness based on the distance
        Evaluate new solution and update light intensity
Rank fireflies and find the current best
```

Sadeghzadeh et al. (Sadeghzadeh and Navaezadeh 2014) improved the replica placement step using firefly technique in data grid environment. The proposed strategy considers grid as a $M \times N$ matrix where M is number of sites and N indicates the number of files. The authors used the total read cost and write cost that are introduced in Tu et al. (2010). In each generation, elements superiority has been determined to find the best locations based on their personal experience and collective intelligence. The results of simulation indicated that the presented strategy could reduce the storage usage.

6.5 Data replication strategies based on particle swarm optimization algorithm

Particle swarm optimization (PSO) is a population-based approach that is proposed by Eberhart and Kennedy (1995). It is inspired by social behavior in bird flocking or fish schooling. Each particle is defined by position and velocity, which moves in a search space. In each iteration, the velocity of each particle is updated according to the local best known position and global best position. Algorithm 5 indicates the pseudo-code of PSO algorithm.

Jayasree and Saravanan (2018) presented an adaptive particle swarm division and replication of data optimization (APSDRDO) algorithm to consider the security of cloud data and automatic data updating. APSDRDO algorithm divides a replica into several fragments and stores them based on T-coloring concept. Therefore, a successful attack on a single data center should not guess the positions of other fragments inside the cloud system. The replica placement is done in two phases. In the first phase, data center is candidate according to the centrality process to improve the retrieval time. In the second phase, data center is selected by PSO algorithm and at the same time updating process is performed. The objective function is defined based on the read time and write time. The comparison results with Greedy, Division and Replication of Data in the Cloud for Optimal Performance and Security (DROPS) (Ali et al. 2018b) and Genetic Replication Algorithm (GRA) indicated that APSDRDO has better average response time.

Algorithm 5. PSO Algorithm

```
Generate a swarm with P particles
Initialize particle swarm
Evaluate fitness of particle swarm
Determine the global best and local best for each particle
Repeat
    Update velocity and position of each particle
    Compute fitness value of each particle
    Update the global best and local best for each particle
Until terminate condition is not met
```

Ebadi and Jafari Navimipour (2018) developed an energy-aware data replication algorithm for cloud. The author used PSO due to strong global search ability and tabu search (TS) due to the powerful local search capability. The fitness values of particles are obtained based on total cost (i.e., reads and writes) and energy. Then, all of the particles repeatedly move until the maximum number of iterations is met. Therefore, the proposed algorithm (HPSOTS) considers the replication problem as a 0–1 decision problem and tries to minimize total energy and cost (TEC). The experimental results demonstrated the strength of HPSOTS, especially for low storage capacities.

Munos et al. (Muñoz and Carballeira 2006) introduced a replica selection strategy based on the PSO-LRU approach for data grid environment. The authors assumed that a file location request is a bird searching food. The bird decides about location of searching according to the flock food chirp. The food chirp is decreased across distance. The proposed strategy considers some performance metrics such as hit ratio and network cost. The network cost is obtained based on the three factors as latency, bandwidth, and the size of file. If in the selected node there is not enough space, the proposed strategy deletes files based on the least recently used approach. The simulation results indicated that the proposed algorithm could reduce response time in the large distributed system.

7 Summary and discussion

This section summarizes the results of the literature review. The distribution of articles by meta-heuristic algorithms is shown in Fig. 10. We can see that 46% of the total articles used GA and 12% of the literature are related to PSO algorithm. There are many meta-heuristic techniques that

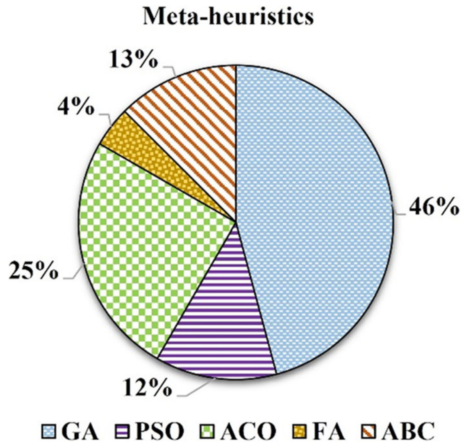


Fig. 10 Distribution of papers based on meta-heuristic algorithms

have been introduced in the last two decades to achieve better design and solve the hard problems in cloud environment such as lion optimization algorithm that has been used for task scheduling to enhance different QoS factors [e.g., resource utilization, degree of imbalance, makespan time (Ahmed Almezeini and Hafez 2017), gray wolf optimization algorithm to reduce the makespan time and energy consumption in cloud system (Natesan and Chokkalingam 2019)]. There is no single nature-inspired optimization algorithm that optimally solves all problems (i.e., no free lunch" (NFL) theorem) (Wolpert and Macready 1997). In other words, it is possible an optimization algorithm can solve a certain set of problems but are unsuitable for other types of issues (Mirjalili 2015). Consequently, the recent and effective optimization algorithms should be tested for solving the objective-based data replication problem. In addition, various modifications of the basic versions of existing meta-heuristic algorithms are presented for solving their weaknesses such as slow convergence rate and parameter adaptation. Due to the successful results of fuzzy logic in various scientific fields, researchers focus on using the positive features of fuzzy logic principles in meta-heuristics design (Kumar et al. 2019; Peraza et al. 2016). Hence, it is concluded that the enhanced versions of algorithms or hybrid meta-heuristics may offer better results compared to the other presented optimizers for complex replication problem.

To simulate cloud and grid environments, there are some prominent simulations tools. We can see in Fig. 11, many algorithms have been implemented by authors in MATLAB or Java. The most popular simulators of cloud and grid are CloudSim and OptorSim to evaluate the experimental results by extending existing classes based on the requirements of algorithm. CloudSim is a discrete-event tool that is introduced by University of Melbourne (Goyal et al. 2012). Through CloudSim large cloud data centers, hosts, virtual components can be simulated and enables users to design resource allocation, scheduling, replication, provisioning strategies, virtualization techniques, and energy management techniques (Mansouri and Javidi 2018c).

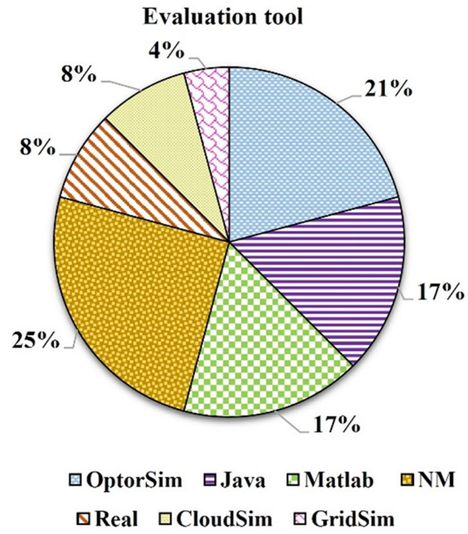


Fig. 11 Comparisons among testing environments

Also, OptorSim was proposed by University of Glasgow in Scotland to simulate the structure of a real data grid and investigate the effectiveness of replication algorithms (Bell et al. 2003). Consequently, it is expected that data replication algorithms are tested based on popular toolkits to configure a real cloud or grid scenario. Interestingly, common simulators (e.g., CloudSim and OptorSim) are also based on the most popular programming language, Java, and may be a reason for their popularity.

From Fig. 12, we can observe that the aim of the general idea of data replication is placing replicas at different locations. Nevertheless, the timing of data replication and selecting file for replication are crucial decisions and may have effect on computing time, storage space, data traffic, and so forth.

Moreover, most of works assumed only the scenario of read-only files and did not consider the consistency concept. However, there are always some read–write files in the system and some of them may be important. Generally, replica consistency management can be performed by two ways: (1) synchronously using the so-called pessimistic techniques and (2) asynchronously considering optimistic approaches. It is necessary to manage the data consistency among the various replicas distributed in different data centers. Since the cost of replication will depend on exactly, when and how those changes need to be carried out, for this aim, Terry et al. (2013) introduced consistency-based SLAs to show a broad set of acceptable consistency/latency trade-offs.

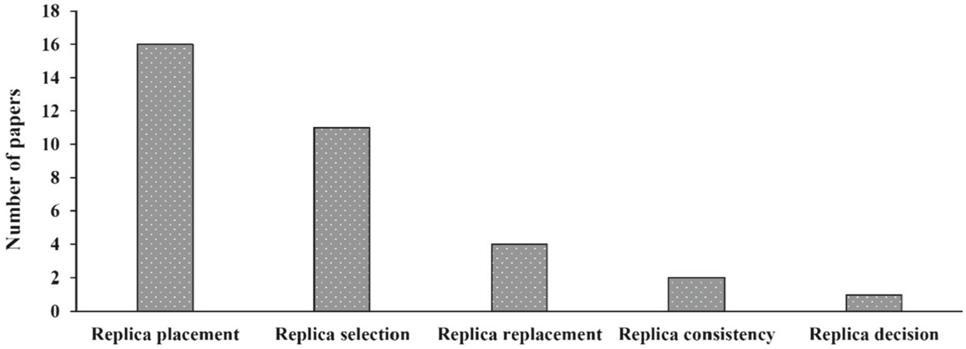


Fig. 12 Distribution of papers based on main challenges of data replication

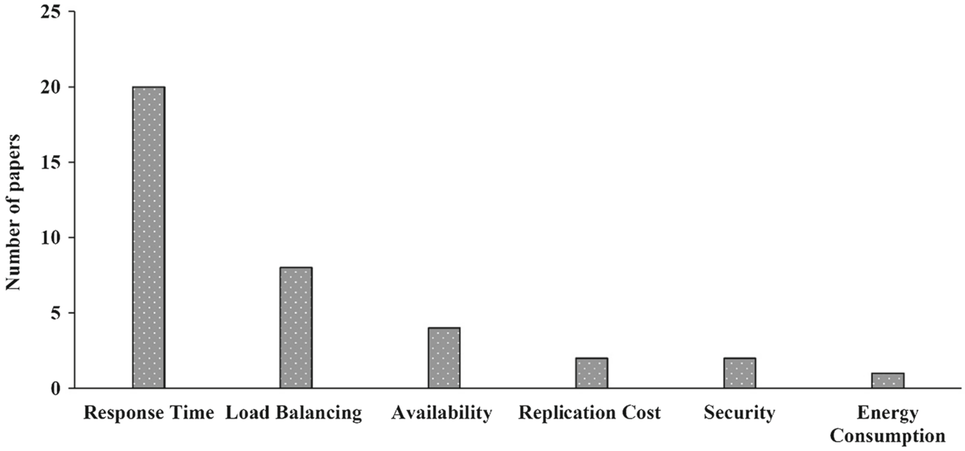


Fig. 13 QoS parameters of data replication algorithms

Figure 13 describes the number of research papers, which are considering different QoS parameters (i.e., response time, load balancing, availability, cost, security, energy, and energy). Figure 13 depicts that response time is used QoS parameter in all research papers, while only 1 research paper considered energy consumption and 2 used security parameter. It distinctly outlines the importance of security and energy factors and necessity of novel and improved data replication algorithms along with the rise in the utilization of cloud computing.

In multiobjective data replication optimization, it is expected to consider several important parameters to meet the prerequisite of users and additionally service providers. Generally, the traditional algorithms used a process based on performance objectives for data replication, but the utilization of multiple objectives like QoS, cost, and load within one strategy was not proposed. This consideration can support optimal services with minimal expense for the users.

8 Case study

In this section, we try to investigate the impact of meta-heuristic techniques on performance of data replication algorithms. We evaluate all data replication algorithms based on meta-heuristic techniques by CloudSim that is a most popular framework for cloud (Calheiros et al. 2011). We have used the same fitness function and configuration for all algorithms. We formulate multiobjective data replication problem into mathematical form and define the fitness function by Eq. (1).

$$\text{Fitness} = w_1 \times \text{RT} + w_2 \times \text{SU} + w_3 \times \text{RN} \quad (1)$$

where RT, SU, and RN represent response time, storage usage, and number of replicas, respectively. w_1, w_2, and w_3 indicate three parameters corresponding to the importance of fitness factors. In this paper, an equal importance considers for all parameters so all weights are set to one. Table 6 indicates the commonly used parametric settings of all strategies.

8.1 Based on cloud structure

In this section, we try to evaluate different algorithms based on various numbers of tasks and virtual machines. Table 7 indicates the values of configuration for CloudSim (Mansouri et al. 2019).

Figure 14 indicates the fitness values for different meta-heuristic-based data replication algorithms (i.e., GA, ACO, FA, BA, PSO, WOA, GWO, and ABC). In Fig. 14a, it is obvious that when the number of tasks increases, the fitness

Table 6 Values for parameters of the meta-heuristics algorithms

Algorithms	Parameters	Values
ABC	Employed bees	30
	Onlooker bees	29
	Random scouts	1
ACO	Evaporation rate (ρ)	0.1
	Quantity of deposit pheromone by the best ant (Q)	0.2
	Pheromone factor (α)	1
	Heuristics factor (β)	1
BA	Frequency [min, max]	[0, 1]
	Initial loudness	1
	Initial pulse rate	0.5
FA	Alfa	0.25
	Beta	0.8
	Gamma	1
GA	Crossover ratio	0.9
	Mutation ratio	0.1
	Selection mechanism	Roulette wheel
PSO	Acceleration constants	[2.1, 2.1]
	Inertia w	[0.9, 0.6]
GWO	α	Min = 0 and max = 2
WOA	a	2 to 0
	r	[0, 1]

Table 7 Detailed characteristics of CloudSim

Parameter	Value
Number of tasks	[4000–6500]
Number of VMs	[20–70]
Number of files	50
Size of files	1500
Bandwidth	800
Iteration	100
Population size	60

value will be increased for all replication methods. In addition, GWO and WOAs show better behavior compared to others in high load condition. For example, WOA in average improves the fitness by 28% and 7% in terms of different number of tasks and VMs compared to FA and 36% and 12% compared to PSO. It is because WOA has a good capability of exploration due to the position updating mechanism of whales. Throughout the initial step of algorithm, this mechanism leads to the whales are moved

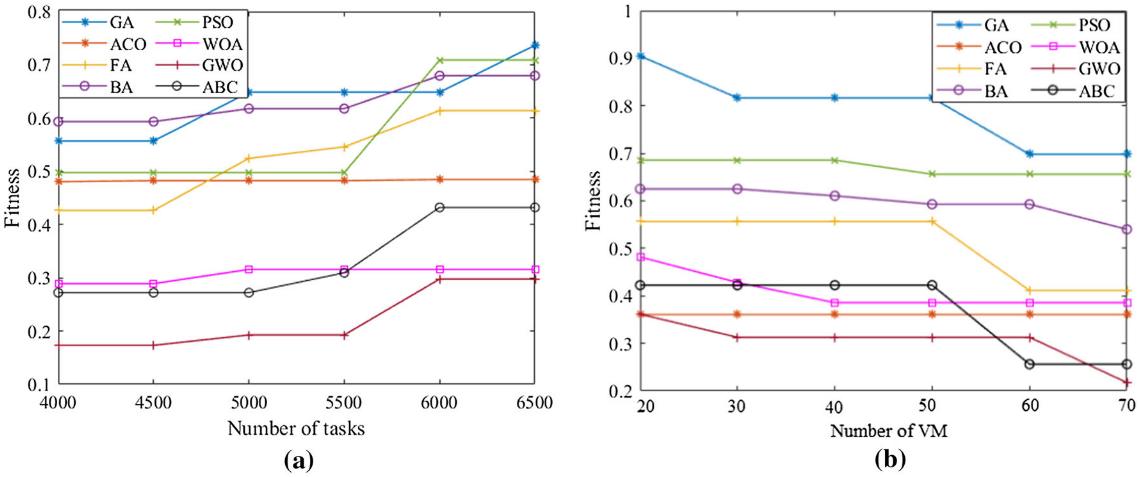


Fig. 14 Fitness values based on **a** number of tasks and **b** number of VMs

randomly around each other. In the next steps, the positions of whales are rapidly updated and then whales move in the best direction as the spiral-shaped route. In WOA, these two phases are performed independently, in half iteration each local optima are avoided, and at the same time the convergence rate is improved through the iterations.

While most of the other optimization techniques such as PSO and FA do not use operators to consecrate a specific iteration for exploration or the exploitation since they consider only one updating method, hence the probability of trapping into local solution is increased. With 20 VMs, we can observe that GWO is the best and GA is the worst since GWO can provide a suitable balance between exploration and exploitation.

For complex problems, the standard randomization method of WOA may lead to increase computational time since randomization has a key role in exploration and exploitation process, while the randomization method has little effect on ABC compared to WOA. In ABC method, the probability is only used to update the employee bees and onlooker bees. The employed bees are responsible for diversification and so perform a local search to each nectar source. The onlooker bees are considered for condensation and updating the better food sources. ABC algorithm can escape from local solutions by considering these groups of bees and so enhance the search capability. The result of this advantage of ABC can be seen in Fig. 14 where ABC improves the fitness compared to WOA, ACO, and GA by 5%, 14%, and 22% in terms of different number of VMs, respectively.

Figure 15 shows the performance of meta-heuristics in terms of fitness value by boxplot format that explains the empirical distribution of data. The boxplot graph in Fig. 15a is generated for 4000 tasks and for different number of virtual machines, while Fig. 15b is generated for 50 VMs and different number of tasks. In addition, the outcomes are obtained after simulating the experiment 30 times. The quality of the algorithm can be determined by the location of its box plot. Therefore, in our case (i.e., minimization optimization) a lower location of the box means the better quality of solutions. Figure 15 shows that the interquartile range and medians of ABC and GWO are comparatively low which means that these two methods perform better to achieve lower fitness. It can be observed from Fig. 15 that the median in the PSO and BA boxplot is in the middle of the rectangle and the whiskers are about the same length. There exists the number of outliers in FA, which is a point of concern to utilize this algorithm as optimization algorithm in replication problem. The ABC plot indicates less variation and spread compared to the other plots. The median of this model is approximately in the middle. Smaller boxes of ABC and GWO show smaller variance in the fitness value, and so they have a stable performance.

8.2 Based on heuristic structure

In this section, we compare replication algorithms in terms of population size and number of iterations. Table 8 shows the CloudSim configuration.

Another parameter that can be discussed for the evaluation of meta-heuristic algorithms is the size of population. In Fig. 16, GWO and ABC have shown the best results among the evaluated strategies. GWO quickly convergences to the global solution by 100 iterations. But other strategies find the promising region with the higher number of iterations. In addition, the ABC algorithm indicates its superior efficiency in Fig. 16, since the searching processes of ABC avoid from trapping in local optima.

GWO can achieve to the global solution (0.2) with a very smooth convergence rate, and ABC is the second best algorithm (0.25). We can observe that BA and FA indicate

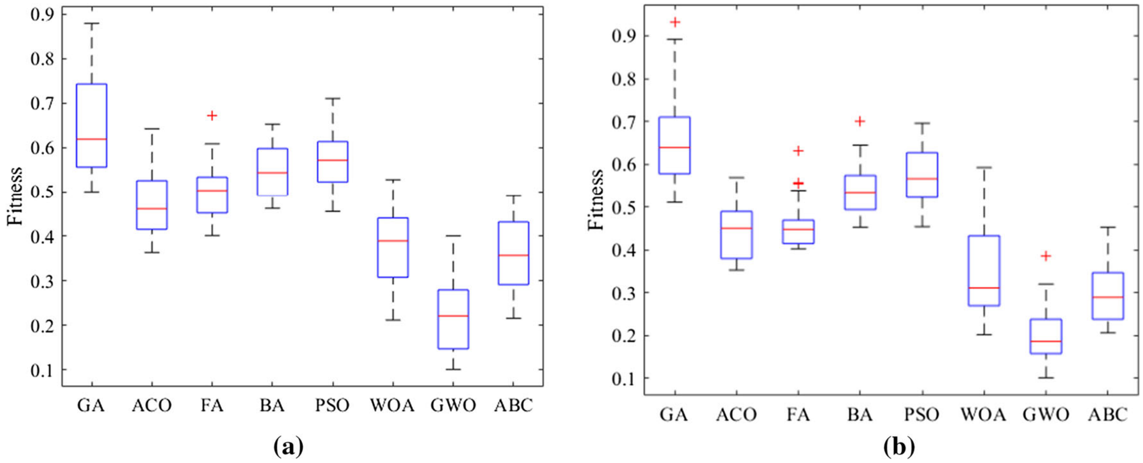


Fig. 15 Boxplot of fitness based on different **a** number of VMs and **b** number of tasks

Table 8 CloudSim configuration

Parameter	Value
Number of tasks	4000
Number of VMs	50
Number of files	50
Size of files	1500
Bandwidth	800
Iteration	[1–100]
Population	[30–60]

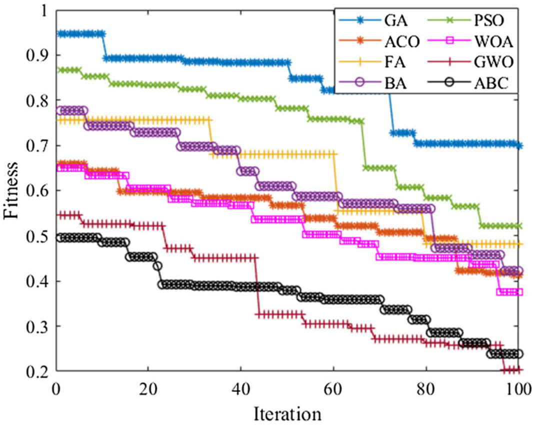


Fig. 16 Fitness value versus number of iterations

the lower convergence rates compared to WOA and ABC with comparably low efficiencies. In other words, BA and FA can reach to global solution with a higher number of iterations. This is, perhaps, because of the poor exploration process of BA and the constant parameters of FA during the searching process and hence may trap in local optima.

BA has better convergence rate than PSO and GA. The main advantage of BA compared to PSO is that BA solves problem with echolocation and frequency tuning. As a result, it presents some capabilities similar to the key feature of particle swarm optimization. In addition, BA can automatically zoom into a promising area by switching from explorative moves to local intensive exploitation, whereas PSO suffers from local optima and low convergence speed to global solution, especially for high-dimensional space.

Figure 17 indicates the fitness value based on different size of populations. It is clear from results that the lowest fitness value (about 2.23) is obtained with size of population 35 by GWO. The main reason is that GWO algorithm applies several candidate solutions to avoid locally optimal solutions. It also suddenly jumps toward the desired region of search space by sharing the information of multiple candidate solutions.

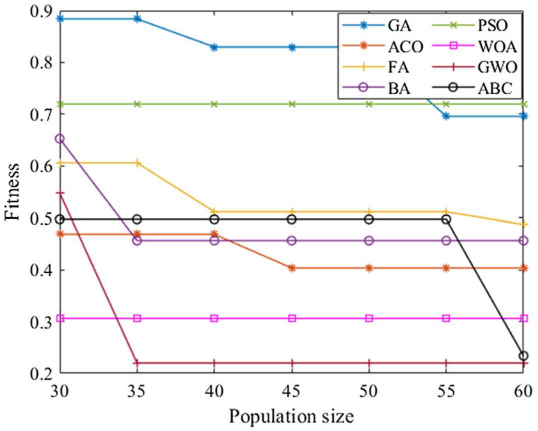


Fig. 17 Fitness value versus population size

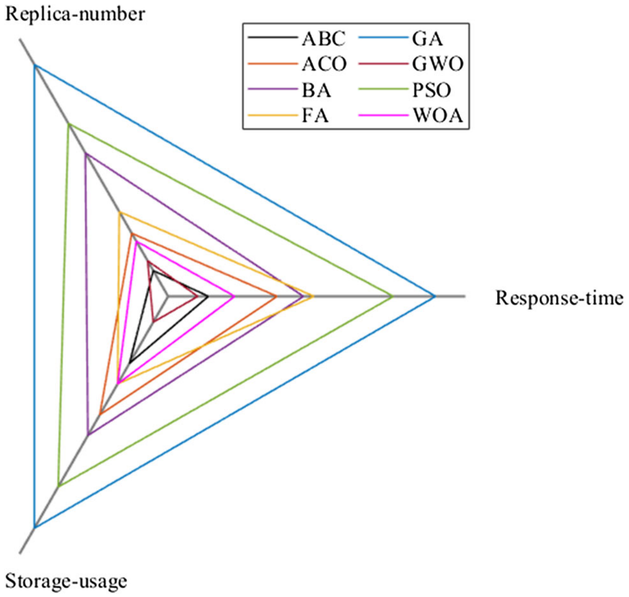


Fig. 18 Response time, storage usage, and number of replicas for different algorithms

Moreover, BA has better performance in terms of fitness (in average improves fitness by 23%) compared to PSO. It also can interpret from Fig. 17 that the increasing the number of particles cannot be effect on PSO to achieve better fitness because its capability of producing new solution is weaker than other new techniques such as GWO and WOA.

The comparison of response time, storage usage, and number of replicas is shown in Fig. 18, revealing that GWO can replicate files in the best manner. In summary, the results prove that GWO shows high performance in solving data replication problems. The main reason is the operators of GWO algorithm to evade local optima and converge faster toward the optimum.

We know that an optimization strategy is proper for solving a certain set of problems but may be inappropriate for some other problems. Therefore, this domain of research is open and allows the researchers to enhance the existing algorithms or introduce new meta-heuristic algorithms for obtaining consistent and accurate results.

9 Open research challenges

We list some open research topics that need attention.

9.1 Modeling the provider profit

Taking the technical side (i.e., delay, portability, etc.) and business side (release clauses, notice time constraints, price difference) together makes the problem of data replication complex. The reduction of operating expenses and increasing the profit of the provided services are critical due to the substantial infrastructure investments being made by cloud providers. One solution for this purpose is maximizing the efficiency of resource utilization. Nevertheless, increasing supplier profits does not necessarily coincide with the improvement of user service quality. The objective of data replication algorithm must be analyzing this multidimensional trade-off. However, fewer of the introduced algorithms consider the economic impact of data replication on the cloud provider. Therefore, studying the combined effectiveness of the optimal number of replicas and strategically storing them in such a way satisfies the response time objective for job exaction while simultaneously enables the provider to return a profit is still an open issue.

There are different types of costs that contribute toward the total expenditure of provider, for example, the computational investment cost that supports hardware to function properly. Network usage cost is another cost that is appeared due to data transmission and remote data access. Storing new replicas will require storage space and increase the expenditure of storage cost. On the other hand, it is difficult to determine how a change in one unit of expenditure leads to a change in another. In addition, the relationship among various kinds of expenditures can change from one provider to another. Consequently, a data replication algorithm should try to estimate the expenditures that affect the profitability of the service provider during its decisions. More work is required toward designing data replication algorithm under specific budget constraints.

9.2 Considering replica consistency

When a data file is changed in the system, the problem of maintaining consistency among existing replicas appears. Therefore, proposing a dynamic replica consistency model for different levels of consistency is critical. In addition, the consistency problem for metadata, i.e., additional information about data like directories and catalogs, exists. This kind of metadata and catalogs is used by replica management service to find the characteristics of replicas for each data. Such directories can also be copied, and their consistency is important to the correct operation of the system. Most of the works done in replication assumed that data are read-only and did not focus on consistency. It is necessary that the consistency algorithms dynamically adapt replica consistency number at runtime to provide a balance between consistency and the quality of service.

9.3 Modeling the energy consumption

Nowadays, one of the critical issues in cloud computing is high energy consumption. When number of replicas is significantly increased, then energy consumption is increased. Many researchers believe that rather than considering a performance approach, it might be more important to focus on energy as a critical parameter during data replication. Nevertheless, energy consumption modeling is another main question that is ignored by most optimization functions. If number of replicas is decreased, then performance of system may degrade. Hence modeling of energy consumption characteristics of data center and balancing a variety of trade-offs with various proposing replication management techniques will give better results. Moreover, it is necessary to consider the shutdown and power-on methods with the objective of optimizing the energy consumption of the data centers.

9.4 Deeper look at reliability and security

Data reliability and security have been widely concerned for current dynamic cloud storage, since public clouds are vulnerable to different kinds of the Denial of Service (DoS) attacks like Distributed Denial of Service (DDoS) and Economic Denial of Service (EDoS) (Masdari et al. 2016). Therefore, considering a data protection model in which a user's sensitive data is replicated on various locations is essential. When there are several replicas in various locations, an attacker must delete or modify all of them to make data unavailable to the user and so data replication can improve the data survivability. On the other hand, the data security should be reduced by creating replicas of data file since it makes data easier to be stolen. Proposing a function of time to model data survivability and security for cloud environment under the protection of data replications are also fields that are continuously drawing the attention of researchers. The traditional data replication algorithms can obtain high data reliability by multireplicas, but leading to large storage consumption. Therefore, these algorithms reduce storage consumption by deleting redundant replicas, but data reliability cannot be guaranteed. Consequently, a dynamic approach for creating minimal replicas that can satisfy the data reliability and availability requirements based on file's storage expectation should be considered.

9.5 Investigate the scalability problem

It is obvious that databases scale up in many dimensions like data structures, number of users, size, and complexity. Hence, all of the aforementioned aspects must be considered as a whole, so that the provider can allocate resources in a manner that can satisfy the requirements of data-intensive tasks. Consequently, another concern is the scalability of strategies as the system grows geographically.

9.6 Exploiting new meta-heuristic algorithms

Also observed from this survey that most of the strategies applied very basic meta-heuristic algorithms like standard PSO and genetic, while recently various modifications are also presented in the basic versions of existing meta-heuristics to solve the drawbacks of the basic ones. In addition, there are novel meta-heuristic techniques such as grasshopper (Saremi et al. 2017) algorithm that provide superior results in comparison with conventional and recent algorithms in the literature. Hence, these search optimization algorithms with high robustness and effectiveness may also be applied to solve data replication problem and provide solutions that are more accurate. Consequently, the recent and effective optimization approaches are necessary to investigate for solving the multiple objective-based data replication problem because the traditional strategy is only considered in most of the works.

9.7 Implementing with suitable simulators

Simulators play a key role to model the system and evaluating algorithms since experimentation in a real environment is very difficult and expensive. Consequently, there has been a significant increase in the development of cloud simulation frameworks with various features. For example, GreenCloud simulation (Kliazovich et al. 2012) tool focuses on energy awareness in every network devices and secCloudSim (Rehman et al. 2014) simulator presents the basic security features of authentication and authorization. Therefore, the researchers must choose the most appropriate simulator based on a set of requirements.

We can see that most of the algorithms included in this survey used personal code to test the strategies. As a next step, these replication algorithms must be implemented in valid simulators based on the real-world scenarios. In addition, most of works compared the simulation results with very simple approaches and so it is interesting to compare the proposed ones with the recent algorithms that are already far better than the basic algorithms.

9.8 Considering the real patterns of user application

Most of works test their algorithms by any orbital setting, but these evaluations do not seem to be enough. More study is necessary to design the user application models. For instance, the four features of big data as volume, velocity, variety, and veracity (Mansouri et al. 2019) can show the final form of the user application. Therefore, the replication algorithm developer will have to put into consideration more complex scenarios where the degree of interdependency will reach a new level.

9.9 Lose the discussion parameters

Most of the algorithms reviewed in this paper do not investigate the impact of various considered parameters on performance results. For instance, history length and network topology have a great impact on the obtained experiment results. Therefore, many experiments are still necessary for thoroughly evaluating the performances of algorithms.

Furthermore, since the execution of a replication algorithm based on meta-heuristic technique closely depends on the performance of the used meta-heuristic technique, the meta-heuristic technique overhead on the algorithm performances should be assessed.

Another problem is the ambiguity in the explanation of some replication algorithms and the lack of necessary and somehow critical details. For instance, the pseudo-code of algorithms and the values of the used parameters are not reported and so their implementations by others are very hard.

9.10 Optimizing the number of replicas

One of the main problems in data replication algorithm is achieving the performance factors, while a particular degree of replication (i.e., a certain number of replicas) is maintained. Therefore, an optimal number of replicas should be determined with economic and performance consideration, since the value of threshold for number of replicas plays a key role in achieving optimality for both users and service providers.

9.11 Using learning techniques

In real systems available today it is hard to gather complete information and discover knowledge about the system in advance. Data mining techniques can efficiently present new meaningful knowledge about tasks, files, and resources to enhance system management. With supervised, unsupervised, and semi-supervised learning techniques, we can find file access patterns, file correlations, and user access behavior. Then, this information can be applied to predict the future behavior system and enhance data replication. In addition, the clustering techniques should be used to find a balanced solution based on QoS requirements like security, load balancing, response time, and cost.

9.12 Widen your horizons: game theory

Another original future direction that we present is to apply game theory for data replication problem. Game theory is a formal study of decision-making and a powerful mathematical analysis for multiple levels of optimality (Yang et al. 2020). It is suitable when two or more agents with various preferences must make mutual decisions. This interdependence causes each agent to consider the other player's possible decisions in own strategy. The agents seek to maximize their individual utility, and a data replication should be proposed in a manner that benefits both resource providers and users. For example, if a solution causes a lower response delay and cost for the users, then the broker profit is increased. Moreover, predicting the player behavior based on prior knowledge would be beneficial. To the best of our knowledge, there is no data replication algorithm based on game theory; thus, we encourage researchers to apply game theory during replication. Game theory-based strategies have low complexity and high computational speeds (Bielik and Ahmad 2012).

10 Conclusions

In this paper, we provide a review of data replication algorithms for cloud computing as well as data grid systems. Only few works classified data replication by taking meta-heuristic approaches as one of the classification criteria. Therefore, we have studied the recent dynamic replication strategies only according to meta-heuristic techniques and discussed their contributions.

According to the findings of a literature review from 2006 to 2019, it can be seen that there is no such meta-heuristic replication algorithm, which can fulfill all the required factors, but they can perform better when some particular factors among response time, resource utilization, latency, etc. have been considered at a time. From the above study, it can be observed that there is still a lot of approach to be proposed in the field of data replication for distributed environments.

Security is one of the main factors that are neglected by most of replication algorithms, and so to thwart the effects of security attacks, future study on data replication problem should notice different security-related parameters during replication decision. For example, data replication strategies can be combined with trust management approaches and trust level of each user should be applied besides the other conventional replication factors in cloud. Furthermore, some important objectives like load balancing on data center should be considered more to achieve green cloud computing.

Meta-heuristic algorithms have experienced a noteworthy shift toward the hybridization with other techniques for combinatorial optimization problems. The hybridization of various algorithms tries to exploit the complementary character of several optimization techniques and benefits from synergy. Hence, the recent hybrid meta-heuristics would be beneficial for solving the multiple objective-based data replication problem because the traditional algorithms are only investigated in most of the works.

Finally, we are convinced that research on game theory-based replication algorithm is still in its early days and researchers interested in this topic can obtain useful contributions. We hope that this review gives some guidance to this interesting line of research.

Compliance with ethical standards

Conflict of interest N. Mansouri and M.M. Javidi declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Abdi S, Mohamadi S (2010) Two level job scheduling and data replication in data grid. Int J Grid Comput Appl (IJGCA) 1:23–37

Ahmed Almezeini N, Hafez A (2017) Task scheduling in cloud computing using lion optimization algorithm. Int J Adv Comput Sci Appl 8(11):77–83

Al Jadaan O, Abdulal W, Abdul Hameed M, Jabas A (2010) Enhancing data selection using genetic algorithm. In: International conference on computational intelligence and communication networks

Alami Milani B, Navimipour N (2016) A comprehensive review of the data replication techniques in the cloud environments: major trends and future directions. J Netw Comput Appl 64:229–238

Alghamdi M, Tang B, Chen Y (2017) Profit-based file replication in data intensive cloud data centers. In: IEEE international conference on communications

Ali M, Kashif B, Khan U, Bhardwaj V, Keqin L, Albert Z (2018a) DROPS: division and replication of data in cloud for optimal performance and security. IEEE Trans Cloud Comput 6:303–315

Ali M, Bilal K, Khan SU, Veeravalli B, Li K, Zomaya AY (2018b) DROPS: division and replication of data in cloud for optimal performance and security. IEEE Trans Cloud Comput 6(2):3030–3315

Aljoumah E, Al-Mousawi F, Ahmad I, Al-Shammri M, Al-Jady Z (2015) SLA in cloud computing architectures: a comprehensive study. Int J Grid Distrib Comput 8(5):7–32

Almomani O, Madi M (2014) A GA-based replica placement mechanism for data grid. Int J Adv Comput Sci Appl 5(10):1–6

Amjad T, Sher M, Daud A (2012) A survey of dynamic replication strategies for improving data availability in data grids. Future Gener Comput Syst 28:337–349

Anjum A, McClatchey R, Ali A, Willers I (2006) Bulk scheduling with the DIANA scheduler. IEEE Trans Nucl Sci 53:18–29

Aznoli F, Jafari Navimipour N (2017) Cloud services recommendation: reviewing the recent advances and suggesting the future research directions. J Netw Comput Appl 77:73–86

Bai X, Jin H, Liao X, Shi X, Shao Z (2013) RTRM: a response time-based replica management strategy for cloud storage system. In: Park JJ et al (eds) Grid and pervasive computing. Springer, Berlin, pp 124–133

Basturk B, Karaboga D (2006) An artificial bee colony (ABC) algorithm for numeric function optimization. IEEE Swarm Intell Symp 8:687–697

Bell WH, Cameron DG, Capozza L, Millar AP, Stockinger K, Zini F (2003) Optorsim: a grid simulator for studying dynamic data replication strategies. Int J High Perform Comput Appl 17(4):403–416

Bielik N, Ahmad I (2012) Cooperative versus non-cooperative game theoretical techniques for energy aware task scheduling. In: International green computing conference

Bilal K, Khan SU, Zhang L, Li H, Hayat K, Madani SA, Min-Allah N, Wang L, Chen D, Iqbal M, Xu CZ, Zomaya AY (2013) Quantitative comparisons of the state of the art data center architectures. Concurr Comput Pract Exp 25(12):1771–1783

Boru D, Kliazovich D, Granelli F, Bouvry P, Zomaya AY (2015) Energy-efficient data replication in cloud computing datacenters. Cluster Comput 18(1):385–402

Bsoul M, Al-Khasawneh A, Abdallah E, Kilani Y (2011) Enhanced fast spread replication strategy for data grid. J Netw Comput Appl 34:575–580

Calheiros RN, Ranjan R, Beloglazov A, De Rose CAF, Buyya R (2011) CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithm. Softw Pract Exp 41(1):23–50

Chunlin L, Ping WY, Hengliang T, Youlong L (2019) Dynamic multi-objective optimized replica placement and migration strategies for SaaS applications in edge cloud. Future Gener Comput Syst 100:921–937

Cui L, Zhang J, Yue L, Shi Y, Li H, Yuan D (2018) A genetic algorithm based data replica placement strategy for scientific applications in clouds. IEEE Trans Serv Comput 11(4):727–739

Dinesh Reddy V, Gangadharan GR, Subrahmanya G, Rao VRK (2019) Energy-aware virtual machine allocation and selection in cloud data centers. Soft Comput 23(6):1917–1932

Dokeroglu T, Sevinc E, Kucukyilmaz T, Cosar A (2019) A survey on new generation metaheuristic algorithms. Comput Ind Eng 137:106040

Dorigo M (1992) Optimization, learning and natural algorithms. Ph.D. thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy

Dorigo M, Birattari M, Stutzle T (2006) Ant colony optimization. IEEE Comput Intell Mag 4:28–39

Ebadi Y, Jafari Navimipour N (2018) An energy-aware method for data replication in the cloud environments using a Tabu search and particle swarm optimization algorithm. Concurr Comput Pract Exp 31:e4757

Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: Proceedings of the sixth international symposium on micro machine and human science (MHS'95), pp 39–43

Ebrahimzade H, Khayati GR, Schaffie M (2018) A novel predictive model for estimation of cobalt leaching from waste Li-ion batteries: application of genetic programming for design. J Environ Chem Eng 6(4):3999–4007

Ebrahimzade H, Khayati GR, Schaffie M (2020) PSO–ANN-based prediction of cobalt leaching rate from waste lithium–ion batteries. J Mater Cycles Waste Manag 22(1):228–239

El-Henawy I, Abdelmegeed NA (2018) Meta-heuristics algorithms: a survey. Int J Comput Appl 179(22):45–54

Farzampour A, Khatibinia M, Mansouri I (2019) Shape optimization of butterfly-shaped shear links using grey wolf algorithm. Ingegneria Sismica 36(1):27–41

Foster I, Zhao Y, Raicu I, Lu S (2008) Cloud computing and grid computing 360-degree compared. In: Grid computing environments workshop, pp 1–10

Gill NK, Singh S (2016) A dynamic, cost-aware, optimized data replication strategy for heterogeneous cloud data centers. Future Gener Comput Syst 65:10–32

Goldberg DE, Holland JH (1988) Genetic algorithms and machine learning. Mach Learn 3(2):95–99

Goyal T, Singh A, Agrawal A (2012) Cloudsim: simulator for cloud computing infrastructure and modeling. Procedia Eng 38:3566–3572

Grace K, Rajkuma M, Sumeetha S, Selvanayaki P (2014) GA based replica selection in data grid. In: International conference on advances in engineering and technology

Hamrouni T, Slimani S, Ben Charrada F (2016) A survey of dynamic replication and replica selection strategies based on data mining techniques in data grids. Eng Appl Artif Intell 48:140–158

Hashemi SM, Khatibi Bardsiri A (2012) Cloud computing vs. grid computing. ARPN J Syst Softw 2(5):188–194

Henry Holland J (1992) Adaptation in natural and artificial systems, 2nd edn. MIT Press, Cambridge

Huang X, Wu F (2018) A cost-effective data replica placement strategy based on hybrid genetic algorithm for cloud services. In: International conference on research and practical issues of enterprise information systems, pp 43–56

Hussain K, Najib Mohd Salleh M, Cheng S, Shi Y (2019) Metaheuristic research: a comprehensive survey. Artif Intell Rev 52(4):2191–2233

Jafari Navimipour N, Alami Milani B (2016) Replica selection in the cloud environments using an ant colony algorithm. In: Third international conference on digital information processing, data mining, and wireless communications, pp 105–110

Jayasree P, Saravanan V (2018) Apsdrdo: adaptive particle swarm division and replication of data optimization for security in cloud computing. IOSR J Eng

Junfeng T, Weiping L (2016) Pheromone-based genetic algorithm adaptive selection algorithm in cloud storage. Int J Grid Distrib Comput 9(6):269–278

Karaboga D (2005) An idea based on honey bee swarm for numerical optimization. Technical report-TR06, Engineering Faculty, Computer Engineering Department, Erciyes University

Khalili Azimi S (2019) A bee colony (beehive) based approach for data replication in cloud environments. In: Kouhsari SM (ed) Fundamental research in electrical engineering. Springer, Singapore, pp 1039–1052

Khojand M, Fatan Serj M, Ashrafi S, Namaki V (2018) Predicting dynamic replication based on fuzzy system in data grid. arXiv:1804.02963

Kingsy Grace R, Manimegalai R (2014) Dynamic replica placement and selection strategies in data grids—a comprehensive survey. J Parallel Distrib Comput 74:2099–2108

Kliazovich D, Bouvry P, Khan SU (2012) GreenCloud: a packet-level simulator of energy-aware cloud computing data centers. J Supercomput 62:1263–1283

Kumar M, Sharma SC, Goel A, Singh SP (2019) Comprehensive survey for scheduling techniques in cloud computing. J Netw Comput Appl 143:1–33

Li R, Hu Y, Lee P (2017) Enabling efficient and reliable transition from replication to erasure coding for clustered file systems. IEEE Trans Parallel Distrib Syst 28(9):2500–2513

Limam S, Mokadem R, Belalem G (2019) Data replication strategy with satisfaction of availability, performance and tenant budget requirements. Cluster Comput 22(4):1199–1210

Liu L, Yang Y, Wang H, Tan Z, Li C (2017) A group based genetic algorithm data replica placement strategy for scientific workflow. In: 16th international conference on computer and information science, pp 459–464

Liu J, Shen H, Narman HS, Lin Z, Li Z (2018) Popularity-aware multi-failure resilient and cost-effective replication for high data durability in cloud storage. Trans Parallel Distrib Syst 30:2355–2369

Long SQ, Zhao YL, Chen W (2014) MORM: a multi-objective optimized replication management strategy for cloud storage cluster. J Syst Architect 60(2):234–244

Ma K, Yang B (2017) Stream-based live data replication approach of in-memory cache. Concurr Comput Pract Exp 29(11):1–9

Mafarja M, Mirjalili S (2018) Whale optimization approaches for wrapper feature selection. Appl Soft Comput 62:441–453

Mansouri N (2014) Network and data location aware approach for simultaneous job scheduling and data replication in large-scale data grid environments. Front Comput Sci 8(3):391–408

Mansouri N (2016) Adaptive data replication strategy in cloud computing for performance improvement. Front Comput Sci 10(5):925–935

Mansouri Y, Buyya R (2018) Dynamic replication and migration of data objects with hot-spot and cold-spot statuses across storage data centers. J Parallel Distrib Comput 126:121–133

Mansouri N, Dastghaibyfard GH (2013) Enhanced dynamic hierarchical replication and weighted scheduling strategy in data grid. J Parallel Distrib Comput 73:534–543

Mansouri N, Javidi MM (2018a) An efficient data replication strategy in large-scale data grid environments based on availability and popularity. AUT J Model Simul 50(1):39–50

Mansouri N, Javidi MM (2018b) A new prefetching-aware data replication to decrease access latency in cloud environment. J Syst Softw 144:197–215

Mansouri N, Javidi MM (2018c) A hybrid data replication strategy with fuzzy-based deletion for heterogeneous cloud data centers. J Supercomput 74(10):5349–5372

Mansouri N, Javidi MM (2019) Cost-based job scheduling strategy in cloud computing environments. Distrib Parallel Databases. https://doi.org/10.1007/s10619-019-07273-y

Mansouri N, Dastghaibyfard GH, Horri A (2011) A novel job scheduling algorithm for improving data grid's performance. In: International conference on P2P, parallel, grid, cloud and internet computing

Mansouri N, Kuchaki Rafsanjani M, Javidi MM (2017) DPRS: a dynamic popularity aware replication strategy with parallel download scheme in cloud environments. Simul Model Pract Theory 77:177–196

Mansouri N, Javidi MM, Mohammad Hasani Zade B (2019) Using data mining techniques to improve replica management in cloud environment. Soft Comput. https://doi.org/10.1007/s00500-019-04357-w

Mansouri N, Mohammad Hasani Zade B, Javidi MM (2019b) Hybrid task scheduling strategy for cloud computing by modified particle swarm optimization and fuzzy theory. Comput Ind Eng 130:597–633

Masdari M, Salehi F, Jalali M, Bidaki M (2016) A survey of PSO-based scheduling algorithms in cloud computing. J Netw Syst Manag 25(1):122–158

Michael MA, Linton A, Michael F, Sebastien G (2010) Autonomic clouds on the grid. J Grid Comput 8:1–18

Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl Based Syst 89:228–249

Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61

Mirzai NM, Zahrai SM, Bozorgi F (2017) Proposing optimum parameters of TMDs using GSA and PSO algorithms for drift reduction and uniformity. Struct Eng Mech 63(2):147–160

Mohammad Khanli L, Isazadeh A, Shishavan TN (2011) PHFS: a dynamic replication method, to decrease access latency in the multi-tier data grid. Future Gen Comput Syst 27(3):233–244

Mokadem R, Hameurlain A (2020) Data replication strategy with tenant performance and provider economic profit guarantees in cloud data centers. J Syst Softw 159:110447

Moura J, Hutchison D (2016) Review and analysis of networking challenges in cloud computing. J Netw Comput Appl 60:113–129

Muñoz VM, Carballeira FG (2006) PSO-LRU algorithm for data grid replication service. In: International conference on high performance computing for computational science, pp 656–669

Nadh Singh BR, Raja Srinivasa Reddy B (2017) A review on big data mining in cloud computing. In: Saini H, Sayal R, Rawat S (eds) Innovations in computer science and engineering. Springer, Singapore, pp 131–142

Nanda SJ, Panda G (2014) A survey on nature inspired metaheuristic algorithms for partitional clustering. Swarm Evol Comput 16:1–18

Natesan G, Chokkalingam A (2019) Optimal task scheduling in the cloud environment using a mean grey wolf optimization algorithm. Int J Technol 10(1):126–136

Park AM, Kim JH, Go YB, Yoon WS (2003) Dynamic grid replication strategy based on internet hierarchy. In: International workshop on grid and cooperative computing, vol 1001, pp 1324–1331

Peraza C, Valdez F, Garcia M, Melin P, Castillo O (2016) A new fuzzy harmony search algorithm using fuzzy logic for dynamic parameter adaptation. Algorithms 9(4):69

Pitchai R, Babu S, Supraja P, Anjanayya S (2019) Prediction of availability and integrity of cloud data using soft computing technique. Soft Comput 23:8555–8562

Qu K, Meng L, Yang Y (2016) A dynamic replica strategy based on Markov model for Hadoop distributed file system, HDFS. In: International conference on cloud computing and intelligence systems, IEEE Computer Society Press, New York, pp 337–342

Rahman RM, Barker K, Alhajj R (2008) Replica placement strategies in data grid. J Grid Comput 6(1):103–123

Ranganathan K, Foster I (2001) Identifying dynamic replication strategies for a high performance data grid. In: International workshop on grid computing, pp 75–86

Ranganathan K, Foster I (2002) Decoupling computation and data scheduling in distributed data-intensive applications. In: Proceedings of 11th IEEE international symposium on high performance distributed computing (HPDC'02)

Rehman UU, Ali A, Anwar Z (2014) secCloudSim: secure cloud simulator. In: 12th international conference on frontiers of information technology, pp 208–213

Sadeghzadeh M, Navaezadeh S (2014) Improving replica in data grid by using firefly algorithm. In: International conference on challenges in IT, engineering and technology (ICCIET'2014), pp 17–18

Salem R, Salam MA, Abdelkader H, Awad A, Arafa A (2019) An artificial bee colony algorithm for data replication optimization in cloud environments. IEEE Access 7:1–12

Sang-Min P, Jair-Hoom K (2003) Chameleon: a resource scheduler in a data grid environment. In: Proceedings of third IEEE international symposium on cluster computing and the grid (CCGRID'03), pp 258–265

Saremi S, Mirjalili S, Lewis A (2017) Grasshopper optimization algorithm: theory and application. Adv Eng Softw 105:30–47

Séguéla M, Mokadem R, Pierson JM (2019) Comparing energy-aware vs. cost-aware data replication strategy. In: Tenth international green and sustainable computing conference (IGSC). IEEE, Alexandria, VA, USA

Shijie J, Yi P, Weisheng L, Liyin S (2010) Study on analyzing questionnaire survey by Monte Carlo simulation. In: International conference on E-business and E-government

Shojaatmand A, Saghiri N, Hashemi S, Abbasi Dezfoli M (2011) Improving replica selection in data grid using a dynamic ant algorithm. Int J Inf Stud 3(4):139

Shojaiemehr B, Rahmani AM, Nasih Qader N (2018) Cloud computing service negotiation: a systematic review. Comput Stand Interfaces 55:196–206

Shvachko K, Hairong K, Radia S, Chansler (2010) The Hadoop distributed file system. In: Proceedings of the 26th symposium on mass storage systems and technologies, pp 1–10

Singh Kushwah V, Kumar Goyal S, Sharma A (2018) Meta-heuristic techniques study for fault tolerance in cloud computing environment: a survey work. In: Ray K, Sharma T, Rawat S, Saini R, Bandyopadhyay A (eds) Soft computing: theories and applications. Springer, Singapore, pp 1–11

Sun M, Sun J, Lu E, Yu C (2005) Ant algorithm for file replica selection in data grid. In: First international conference on semantics, knowledge and grid

Sun DW, Chang GR, Gao S, Jin LZ, Wei Wang X (2012) Modeling a dynamic data replication strategy to increase system availability in cloud computing environments. J Comput Sci Technol 27(2):256–272

Taheri J, Choon Lee Y, Zomaya AY, Jay Siegel H (2013) A bee colony based optimization approach for simultaneous job scheduling and data replication in grid environments. Comput Oper Res 40(6):1564–1578

Terry DB, Prabhakaran V, Kotla R, Balakrishnan M, Aguilera MK, Abu-Libdeh H (2013) Consistency-based service level agreements for cloud storage. In: Proceedings of the twenty-fourth ACM symposium on operating systems principles

Tharani R (2016) Balanced ant colony optimization algorithm for job scheduling in grid computing. Int J Eng Res Technol 4(11):1–6

Tos U, Mokadem R, Hameurlain A, Ayav T, Bora S (2015) Dynamic replication strategies in data grid systems: a survey. J Supercomput 71(11):4116–4140

Tos U, Mokadem R, Hameurlain A, Ayav T, Bora S (2018) Ensuring performance and provider profit through data replication in cloud systems. Cluster Comput 21:1479–1492

Tsai CW, Rodrigues J (2014) Metaheuristic scheduling for cloud: a survey. IEEE Syst J 8(1):279–297

Tsai CW, Tsai PW, Pan JS, Chao HC (2015) Metaheuristics for the deployment problem of WSN: a review. Microprocess Microsyst 39(8):1305–1317

Tu M, Li P, Yen IL, Thuraisingham BM, Khan L (2010) Secure data objects replication in data grid. IEEE Trans Depend Secure Comput 7(1):50–64

Tziritas N, Kolodziej J, Zomaya AY, Madani SA, Min-Allah N, Wang L, Xu CZ, Marwan Malluhi Q, Pecero JE, Balaji P, Vishnu A, Ranjan R, Zeadally S, Li H (2015) Performance analysis of data intensive cloud systems based on data management and replication: a survey. Distrib Parallel Databases 34(2):179–215

Wang L, Luo J, Shen J, Dong F (2013) Cost and time aware ant colony algorithm for data replica in alpha magnetic spectrometer experiment. In: IEEE international congress on big data, pp 247–254

Wei Q, Veeravalli B, Gong B, Zeng L, Feng D (2010) CDRM: a cost-effective dynamic replication management scheme for cloud storage cluster. In: IEEE international conference on cluster computing, pp 188–196

Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1(1):67–82

Wu X (2016) Data sets replicas placements strategy from cost-effective view in the cloud. Sci Program 11:1–13

Wu X (2017) Combination replicas placements strategy for data sets from cost-effective view in the cloud. Int J Comput Intell Syst 10:521–539

Xu Q, Xu Z, Wang T (2015) A data-placement strategy based on genetic algorithm in cloud computing. Int J Intell Sci 5:145–157

Yang X-S (2009) Firefly algorithms for multimodal optimization. In: International symposium on stochastic algorithms, pp 169–178

Yang X-S (2010) A new metaheuristic bat-inspired algorithm. In: Nature inspired cooperative strategies for optimization (NICSO 2010), pp 65–74

Yang XS (2013) Firefly algorithm: recent advances and applications. Int J Swarm Intell 1(1):36–50

Yang L, Lin J, Zheng Y (2013) A replica selection strategy on ant-algorithm in data-intensive applications. Int J Online Eng 9:38–41

Yang J, Jiang B, Lv Z, Raymond Choo KK (2020) A task scheduling algorithm considering game theory designed for energy management in cloud computing. Future Gen Comput Syst 105:985–992

Yuan D, Yang Y, Liu X, Chen JJ (2010) A data placement strategy in scientific cloud workflows. Future Gener Comput Syst 26(8):1200–1214

Zhang B, Wang X, Huang M (2014) A data replica placement scheme for cloud storage under healthcare IoT environment. Appl Mech Mater 556–562:5511–5517

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.