УНИВЕРСИТЕТ ИТМО
НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
июль–август 2021 Том 21 № 4 http://ntv.ifmo.ru/
SCIENTIFIC AND TECHNICAL JOURNAL OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS
July–August 2021 Vol. 21 No 4 http://ntv.ifmo.ru/en/
ISSN 2226-1494 (print) ISSN 2500-0373 (online)
НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

КОМПЬЮТЕРНЫЕ СИСТЕМЫ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ
COMPUTER SCIENCE

doi: 10.17586/2226-1494-2021-21-4-463-472

Nature-inspired metaheuristic scheduling algorithms in cloud: a systematic review

Sandeep Kumar Bothra[1]✉, Sunita Singhal[2]✉

[1,2] Manipal University Jaipur, Jaipur, Rajasthan, 303007, India

[1] bothrajain@gmail.com✉, http://orcid.org/0000-0003-0555-569X
[2] sunita.singhal@jaipur.manipal.edu✉, http://orcid.org/0000-0003-2462-8102

Abstract
Complex huge-scale scientific applications are simplified by workflow to execute in the cloud environment. The cloud is an emerging concept that effectively executes workflows, but it has a range of issues that must be addressed for it to progress. Workflow scheduling using a nature-inspired metaheuristic algorithm is a recent central theme in the cloud computing paradigm. It is an NP-complete problem that fascinates researchers to explore the optimum solution using swarm intelligence. This is a wide area where researchers work for a long time to find an optimum solution but due to the lack of actual research direction, their objectives become faint. Our systematic and extensive analysis of scheduling approaches involves recently high-cited metaheuristic algorithms like Genetic Algorithms (GA), Whale Search Algorithm (WSA), Ant Colony Optimization (ACO), Bat Algorithm, Artificial Bee Colony (ABC), Cuckoo Algorithm, Firefly Algorithm and Particle Swarm Optimization (PSO). Based on various parameters, we do not only classify them but also furnish a comprehensive striking comparison among them with the hope that our efforts will assist recent researchers to select an appropriate technique for further undiscovered issues. We also draw the attention of present researchers towards some open issues to dig out unexplored areas like energy consumption, reliability and security for considering them as future research work.

Keywords
genetic algorithm, literature review, nature inspired algorithm, metaheuristic scheduling algorithm, swarm intelligence

For citation: Bothra S.K., Singhal S. Nature-inspired metaheuristic scheduling algorithms in cloud: a systematic review. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2021, vol. 21, no. 4, pp. 463–472. doi: 10.17586/2226-1494-2021-21-4-463-472

УДК 004.75

Биоинспирированные метаэвристические алгоритмы построения расписаний в облаке: систематический обзор

Сандип Кумар Ботра[1]✉, Сунита Сингхал[2]✉

[1,2] Университет Манипала в Джайпуре, Джайпур, Раджастан, 303007, Индия

[1] bothrajain@gmail.com✉, http://orcid.org/0000-0003-0555-569X
[2] sunita.singhal@jaipur.manipal.edu✉, http://orcid.org/0000-0003-2462-8102

Аннотация
Применение сложных крупномасштабных научных приложений упрощается в случае их обработки в облачной среде. Дальнейшее развитие облачных технологий связано с решением ряда новых проблем. Центральной темой парадигмы облачных вычислений является планирование рабочих процессов с использованием биоинспирированных метаэвристических алгоритмов. NP-полная задача (NP-completeness) привлекает исследователей к поиску оптимального решения с использованием роевого интеллекта. В работе представлены систематизированный анализ и оценка метаэвристических алгоритмов, таких как генетический (Genetic Algorithms, GA), китовый (Whale Search Algorithm, WSA), муравьиный (Ant Colony Optimization, ACO), летучих мышей (Bat Algorithm, BA), пчелиный (Artificial Bee Colony, ABC), кукушкин поиск (Cuckoo Algorithm, CA), светлячковый (Firefly Algorithm, FA), оптимизация роем частиц (Particle Swarm Optimization, PSO). Представлены параметры алгоритмов, дана их классификация, приведено подробное сравнение. Уделено внимание нерешенным проблемам, таким как потребление энергии, надежность и безопасность. Представленные результаты позволят исследователям выбрать подходящие решения возможных новых проблем в облачных вычислениях.

Ключевые слова
генетический алгоритм, обзор, биоинспирированный алгоритм, метаэвристический алгоритм, роевой интеллект

Ссылка для цитирования: Ботра Сандип Кумар, Сингхал Сунита. Биоинспирированные метаэвристические алгоритмы построения расписаний в облаке: систематический обзор // Научно-технический вестник информационных технологий, механики и оптики. 2021. Т. 21, № 4. С. 463–472 (на англ. яз.). doi: 10.17586/2226-1494-2021-21-4-463-472

Introduction

In the last few years, the distributed computing paradigm has become a buzzword due to its robust features like reliability, elasticity, scalability, and sharing ability. Due to the pay-as-you-go (PAYG) and dynamic scalable nature Cloud Computing is an emerging technology of the distributed computing paradigm [1]. Scheduling is a process used to allocate resources among a set of tasks in a distributed environment in order to achieve Quality of Service (QoS) within a time frame, otherwise end-users will be hesitant to pay the service provider, despite the service provider's promises to users via Service Level Agreement (SLA) [2, 3]. Optimum resource scheduling is one of the central themes in the cloud, which is NP-complete. We have not had such an algorithm till now that generates an optimal solution within the polynomial time for the NP-complete problem [4]. Due to the local optimum nature of the heuristic approach, researchers are moving towards meta-heuristic techniques. The global optimum result can be achieved by the nature-inspired algorithm which is meta-heuristic in flavor. Nature-inspired algorithms may be biotic and abiotic phenomena. The bio-inspired algorithm is biotic, whereas the algorithm based on physical and chemical properties is abiotic. Bio-inspired algorithms are mostly based on the behavior of plants and animals, like flower pollination algorithm, Strawberry Plant Algorithm, Dolphin echolocation algorithm, etc. but not all. Some bio-inspired algorithms are not dependent upon the behavior of animals, like queen-bee evolution etc. Mostly bio-inspired algorithms are swarm-intelligence based like ant colony optimization etc. Physical and chemical Properties-based algorithms are black hole algorithm etc. An overview of nature-inspired metaheuristic algorithms is illustrated by (Fig. 1).

Scheduling strategies are classified as optimal or sub-optimal [5]. To achieve an optimal solution to the NP-complete problem is very expensive, so it is better to find an approximate solution, i.e. sub-optimal. This is the reason why researchers focus on resolving such problems through metaheuristic techniques. Heuristic techniques are problem-specific and thus they cover small domain areas. Because of their problem independence, metaheuristic techniques attract researchers. Nowadays researchers attempt to solve such problems using a hybrid technique that combines heuristic and metaheuristic techniques.

Following a review of the literature, an extract of a comparative analysis on optimization techniques is shown in (Table 1).

The Metaheuristic Algorithm is defined by literature [6] as "an iterative master process that guides and modifies the operations of subordinate heuristics to efficiently produce high-quality solutions. It may manipulate a complete (or incomplete) single solution or a collection of solutions per iteration. The subordinate heuristics may be high (or low) level procedures, or a simple local search, or just a construction method". The authors [7] define a metaheuristic algorithm as "an iterative generation process that guides a subordinate heuristic by intelligently combining different concepts for exploring and exploiting the search space, learning strategies are used to structure information in order to find efficiently near-optimal-solutions".

Literature Review

We have studied various scheduling approaches based on metaheuristic algorithms like Artificial Bee Colony (ABC), Whale Search Algorithm (WSA), Bat Algorithm, Cuckoo Algorithm, Firefly Algorithm, Genetic Algorithm

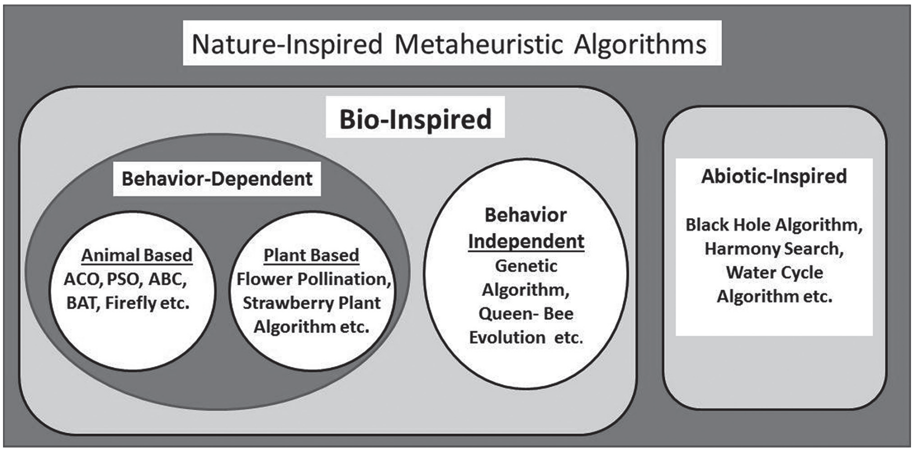

Fig. 1. Representation of Nature-Inspired Metaheuristic Algorithms

Table 1. Comparative Study of Optimization Techniques

Heuristic Algorithm	Metaheuristic Algorithm	Hybrid Algorithm
Problem dependent	Problem independent	Mix mode
Cover specific area of problem	Cover large area due to independent nature	Cover huge area of problem
Rule-based solution	Framework based solution	Mix mode
Processing time — low	Processing time — high	Processing time — medium
Function nature — white-box	Function nature — black-box	Function nature — mix mod
Domain area — small	Domain area — large	Domain area — very large

(GA), Particles Search Optimization (PSO), and Ant Colony Optimization (ACO). The summary of all these is represented in (Fig. 2).

The authors improved the particle swarm optimization to schedule the workflow. They began by using a nonlinear reducing method of inertia weight to manage the global and local performance of particles; followed by a perfect scheduling plan to achieve the shortest possible time and cost, but they ignored the dynamic feature of the cloud computing environment [8]. A multi-objective algorithm to schedule workflow is presented by authors based on the PSO approach. To achieve their goals, they include two parameters, makespan and resource utilization, as well as a rigorous encoding scheme, in their novel algorithm.

Although their experimental result illustrates that their approach is more robust than the baseline approaches, they ignore the balancing of VMs [9]. The authors of [10] included a simulated annealing algorithm with PSO to escape sinking into local optima and enhance the convergence speed of the algorithm. Their main goal was to reduce the execution time of tasks as well as efficiently utilize the cloud's resources, but they did not focus on dynamic scheduling of workflow and security concepts.

In the paper [11], the authors tried to reduce the execution time and cost of the workflow by applying the two ant colonies approach and focused on executing maximum tasks parallel in ACO. To achieve the global optimum objective, they designed a new technique to update the pheromone. A complementary heuristic strategy (CHS) and an elite study strategy (ESS) are applied to achieve multiple objectives of the algorithm. Allocation of underutilized virtual machines by Pareto distribution is applied by authors [12] to minimize execution cost and execution time in ACO. They also adopt the approach of minimum migration of virtual machines to boost the performance of their approach in the assessment of execution time and cost of workflow, but their practical approach is based on a very small size of the workflow, so the performance of the algorithm is not reliable.

The authors of [13] used the ACO technique to minimize the makespan by grouping the ordered tasks, but they did not consider cost, security, or load balance.

The paper [14] introduced a hybrid approach to schedule workflow by applying Artificial Bee Colony (ABC) with PSO. Their approach showed better results due to exploring the wider area of a solution space. The study [15], proposes an Artificial Bee Colony (ABC) based algorithm, in which the authors emphasize on the quality of service policies and crucial security concepts. To minimize the execution cost, execution time, migration of task, and load-balance of VMs, a hive table is maintained in a data center. Only the ABC approach is not enough to handle all these parameters, so they had to develop a hybrid technique.

The authors make an attempt to schedule the workflow using the firefly algorithm (FA), taking into account reliability, makespan, and resource utilization while maintaining a balanced load among various virtual machines. To select the proper virtual machine, they applied a rule-based strategy. They did not focus on booting time and termination delay of VMs, which impacted their algorithm's objective [16]. This dimness is removed in [17], where authors proposed a cost-effective approach using the firefly algorithm (FA) to schedule the scientific workflow under deadline constraint while considering performance variation of CPU as well as termination delay. To design the humpback whale optimization algorithm, intelligent techniques should be applied to enhance the performance.

By applying the Cuckoo Optimization Algorithm (COA) with a harmony search approach, the authors improved the scheduling performance in a cloud environment [18] where

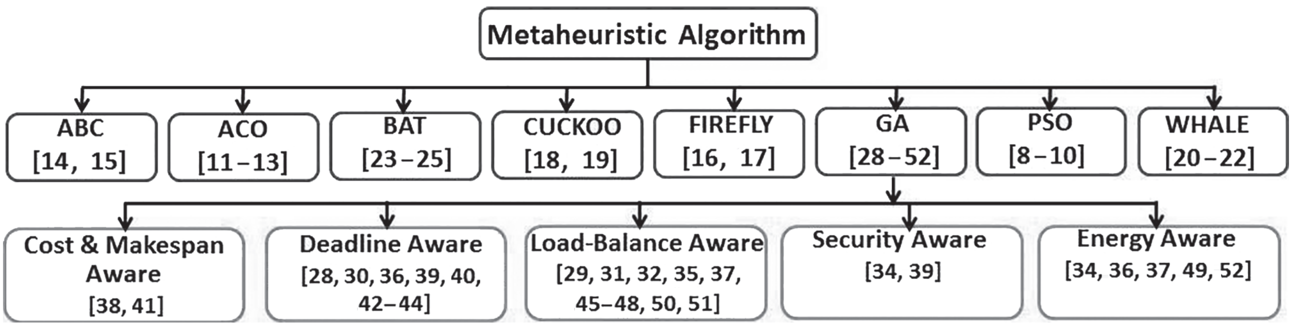

Fig. 2. Prominent Meraheuristic Algorithm

they included cost, energy consumption, penalty factors and utilization of memory but they did not concern about load balance among the processors, which is improved in the work [19].

Vocalization of humpback whale optimization algorithm [20] is proposed to minimize the execution cost and time. This approach minimizes energy consumption to protect the environment. The authors presented a multi-objective deadline constraint-based workflow scheduling algorithm based on whale social behavior, in which they attempted to minimize makespan by considering load balance among virtual machines, but they failed to consider the dynamic nature of cloud computing, which plays an important role in the scheduling process [21]. This is solved in [22], where authors considered the dynamic behavior of cloud in their grouping whale's optimization algorithm. The first population is arranged in ascending order, then it is divided into several groups and a member is selected randomly from each group to encircle the prey section to minimize the time of response as well as execution and enhance the throughput in a cloud computing environment.

An algorithm based on the BAT optimization strategy to schedule the workflow was proposed to optimize time and reliability in the cloud [23]. The authors applied the greedy approach to minimize the cost and execution time by improving the reliability under budget constraint, but there is no awareness about the energy consumption etc. To remove this weakness, the authors [24] gave more emphasis on energy consumption in their approach, although they included execution time and throughput but they did not consider communication time, which is an important factor in minimize the execution time and enhancing the throughput. Load balance among various VMs was also not considered by them. The work [25] proposes to use the Bat algorithm to balance the load on the various VMs, where authors tried to improve the resource allocation for VMs.

The papers [26, 27] provide meticulous information regarding metaheuristic algorithms. After reviewing various genetic algorithms [28–52], we present our deep investigation in brief (Tables 2, 3 and 4) and a comparative analysis on some metaheuristic algorithms are depicted in (Table 5).

We have analyzed various articles [53–57] and can conclude that in a cloud environment there are various issues required to be resolved.

Table 2. Study of Various Genetic Algorithms in Reverse Chronological Order

Reference	Publication Year	Objective	Resource Type	Nature of Input (Independent Task/Workflow)	Workflow Type	Experimental Environment
[28]	2020	Multi-Objective	Heterogeneous	Workflow	Scientific	CloudSim, JAVA
[29]	2020	Single-Objective	Heterogeneous	Independent	—	CloudSim
[30]	2020	Multi-Objective	Homogeneous	Workflow	Scientific	WorkflowSim
[31]	2020	Multi-Objective	Heterogeneous	Workflow	Scientific	CloudSim
[32]	2019	Multi-Objective	Heterogeneous	Independent	—	CloudSim
[33]	2019	Multi-Objective	Heterogeneous	Workflow	Scientific	jMetal Tool
[34]	2019	Multi-Objective	Heterogeneous	Workflow	Scientific	WorkflowSim
[35]	2019	Single-Objective	Homogeneous	Workflow	Scientific	C++
[36]	2018	Multi-Objective	Heterogeneous	Workflow & Independent	Simple	JAVA
[37]	2018	Single-Objective	Heterogeneous	Independent	—	MATLAB
[38]	2018	Single-Objective	Heterogeneous	Independent	—	CloudSim
[39]	2018	Single-Objective	Heterogeneous	Workflow	Scientific	CloudSim
[40]	2017	Single-Objective	Heterogeneous	Workflow	Scientific	WorkflowSim
[41]	2017	Multi-Objective	Heterogeneous	Workflow	Scientific	MATLAB
[42]	2016	Single-Objective	Heterogeneous	Workflow	Scientific	CloudSim
[43]	2016	Single-Objective	Heterogeneous	Workflow	Scientific	WorkflowSim
[44]	2015	Single-Objective	Heterogeneous	Workflow	Simple	No Mention
[45]	2014	Single-Objective	Heterogeneous	Workflow	Simple	C# Language
[46]	2014	Single-Objective	Heterogeneous	Workflow	Simple	In Real Cloud
[47]	2014	Single-Objective	Heterogeneous	Independent	—	MATLAB
[48]	2013	Single-Objective	Homogeneous	Independent	—	CloudAnalyst
[49]	2012	Multi-Objective	Homogeneous	Independent	—	CloudSim
[50]	2012	Single-Objective	Heterogeneous	Workflow	Scientific	No Mention
[51]	2011	Single-Objective	Homogeneous	Workflow	Simple	CloudSim
[52]	2011	Multi-Objective	Homogenous	Independent	—	No Mention

Table 3. Comparison of Various Scheduling Approaches based on GA

Reference	Scheduling Approach	Initialization	Selection	Crossover	Mutation
[28]	GA with HEFT	First by HEFT and remaining using efficient routine	Roulette Wheel method	Single-point, Double-points, Triple-points	Single-point and Double-point with swap
[29]	GA with Greedy Selection Technique	Randomly using Binary code	Roulette Wheel method	Double-points	Double-point with new method
[30]	GA with HEFT	Initialization using HEFT	Tournament Selection	Two-point crossover	Single-point simple swap
[31]	GA (For each parameter 1st get best solution, after that super best selected)	Randomly	Roulette Wheel method	Clustered crossover operator	Single-point simple swap
[32]	GA with EDA	Use EDA to initialize the probability then set to 1/m to ensure the randomness of the initial population.	Roulette Wheel method	One-point crossover	Double-point simple swap
[34]	Multi-populated GA	Heuristic in the generation	No Mention	Random	Random
[35]	GA with PSO	First generate randomly then apply PSO	1st particle by Gbest and 2nd by random	Single- point crossover	Single-point simple swap
[36]	GA with Gap Search Algorithm	Randomly	Tournament Based	Single- point crossover	Task Ti is mutated by transferring it from VMm to VMn
[37]	Parallel GA with Priorities Strategy	Randomly	Roulette Wheel method	Single- point crossover	Single-point simple swap
[39]	GA with Multi- Population and PSO	Best Sec Operator Based Initialization	Tournament Based	Three crossover (cross-Uniform, cross- Average, cross - BLX)	Two mutation (mut-Strong, mutLimit) operators
[40]	GA with PEFT and PGA	First Chromosome by PEFT and rest Randomly	Tournament Based	Single- point crossover	Single-point simple swap
[41]	NSGA-III	Randomly and new method	Niche-Preservation Operation is applied	Single point -Simulated Binary Crossover (SBX)	Gaussian Mutation
[42]	GA with some part of JIT-C	First randomly then JIT approach	Tournament Based	Two-point crossover	Single-point simple swap
[44]	GA	No Mention	Roulette wheel Method	Two-point crossover	Single-point simple swap
[45]	GA with HEFT	By 3 heuristic rank policies	No Mention	One-point crossover with new technique	Single-point with their new technique
[46	GA with Best Fit and Round Robin Approach	Based on Best Fit and Round Robin Approach	No Mention	Randomly Gene Selection	First randomly select gene then replace its resource by less loaded resource having better failure rate
[47]	GA with job spanning time and load balancing Greedy Approaches	By Applying Greedy approach	Rotating Selection	One Point	Local Search
[48]	GA	Randomly	Randomly	Single-point	Toggled from 1 to 0 or 0 to 1
[49]	Energy consumption with ETU_GA and ETDF_GA	Randomly	Elitist Generational Strategy and Roulette wheel method	Single-point	Single-point simple swap
[51]	Genetic Algorithm with Markov Decision Process	Randomly	Roulette Wheel method	Single-point	Single-point simple swap
[52]	GA with Feitelson's Parallel Workload Archive	Greedy method and Random method	Tournament Strategy	Two-point crossover	Single-point simple swap

Table 4. Matrix Considered to Schedule Workflow using GA

Reference	Cost	Makespan	Deadline	Load Balance	Security	Energy
[28]	✓	✓	✓	✓		
[29]	✓	✓		✓		
[30]	✓	✓	✓			
[31]	✓	✓		✓		
[32]		✓		✓		
[33]	✓	✓				
[34]	✓	✓			✓	✓
[35]		✓		✓		
[36]	✓	✓	✓			✓
[37]				✓		✓
[38]	✓	✓				
[39]	✓		✓		✓	
[40]	✓		✓			
[41]	✓	✓				
[42]	✓		✓			
[43]	✓		✓			
[44]	✓	✓	✓			
[45]	✓	✓		✓		
[46]		✓		✓		
[47]		✓		✓		
[48]				✓		
[49]		✓				✓
[50]				✓		
[51]	✓			✓		
[52]						✓

An extract from a comparative analysis on some metaheuristic algorithms is illustrated in (Table 5).

There are several available survey articles on metaheuristic algorithms, however in this study, we comprehensively covered the most recent metaheuristic techniques and focused on their pros and cons. In fact, the current scenario necessitates the security of sensitive data/tasks and awareness of energy consumption. As per our survey, the major security components of the cloud include authentication services, integrity services, and confidentiality services. As there has been a little bit of research in these areas we have attempted to draw the

Table 5. Strength and Limitation of GA, PSO and ACO

Algorithm	Strength	Limitation
GA	Other techniques can be combined easily.	Encoding scheme is complex.
	Search space can be explored in various directions simultaneously.	Convergence rate is low.
	Manipulation of various parameters can be done at the same time.	Crossover and Mutation rates depend on stability.
	An efficient global optimum solution can be achieved for various problems.	
	Able to resolve complex optimization problem of various types.	
PSO	Low level of dependency during initial point.	Convergence rate is very low.
	There are few parameters to adjust.	Trapping into local optima.
	Performs good global search.	Capacity of local search is weak.
ACO	Graph based complex problem can be solve easily.	Theoretical analysis is very complex.
	Able to solve problem related to the dynamic nature.	Initialization of parameters is based on trial and errors.

attention of current researchers to it. Our observations, which are based on this survey, focus on a variety of technical issues and will guide present researchers in their decision for selection of an appropriate metaheuristic technique and lighting the path of research.

Observations and Discussion

Our observation based on the above described surveys is as follows:

Using local search approaches to build the initial population can increase the quality of resolutions obtained by metaheuristic algorithms which are based on the population. The elite solutions, which are derived from previous generations' greatest answers, can also be utilized to populate future generations' beginning populations. If these elites are strengthened before becoming a part of the following generation, they can produce better results than the initial elites.

Combining a metaheuristic algorithm with another metaheuristic algorithm which is based on population or a local search-based metaheuristic method, can improve solution quality or convergence speed.

The transition operators employed in metaheuristic algorithms have been modified by researchers. It is beneficial to improve the consistency of the solution by changing the transfer operator.

Virtual Machine placement optimization, Virtual Machine consolidation, and Dynamic Voltage and Frequency Scaling strategies are commonly used for energy conservation. The most significant disadvantage of this method is that frequency and voltage can only be altered to a limited extent.

Service providers should agree to a dual Service Level Agreement with the consumer, with the second SLA being optional and selected only when the cloud customer requires the "Green mode". The term "green mode" refers to a mode in which the primary purpose is to save energy at the expense of production.

The majority of energy-aware scheduling research has used metaheuristic strategies with the goal of lowering energy consumption. Computation resources produce a lot of heat, which makes execution more error-prone and, as a result, can reduce machine efficiency and shorten computer life spans.

In order to solve large-scale combinatorial and multimodal problems, exact optimization algorithms are ineffective. An exhaustive search for the algorithms is impractical for dealing with these problems because the search space grows exponentially with the size of the problem. As a result, a large number of researchers have used meta-heuristic algorithms to solve the issues. Metaheuristic algorithms have various advantages:

They are not designed to solve a particular problem and can be used to solve multimodal complex problems.

They are easy to use in parallel processing and are adaptable to changing situations and environments.

They can include mechanisms to prevent them from being stuck in local optima.

Because of their discovery and extraction capabilities, these algorithms are able to identify promising regions in a reasonable amount of time.

Although the listed algorithms have shown satisfactory results in a variety of fields, they do not guarantee that an ideal solution can ever be found, and they do have some inevitable drawbacks like consuming long execution time, trapping in local optima, low convergence speed, several parameter tuning, complicated encoding scheme, and only decent output in real or binary search spaces. As a result, it appears that improving the performance of previous meta-heuristics, or even introducing new ones, is fruitful for current researchers.

Conclusion

We analyzed the most famous bio-inspired metaheuristic algorithms like ABC, ACO, BAT, Cuckoo, Firefly, GA, PSO, and Whale Optimization algorithms. We illustrated the classification of metaheuristic algorithms based on various factors and presented the comparison among the latest approaches with the hope that our efforts will give directions to current researchers to select the appropriate technique which meets their objective.

We discovered that most researchers are concerned with minimizing execution cost, execution time, response time, makespan, and increasing throughput, even though current researchers are becoming more concerned with protecting the environment by reducing energy consumption and carbon-dioxide emissions without affecting the Service Level Agreement (SLA). We also draw attention to unresolved issues as trapping in local optima, low convergence speed, several parameters tuning, complicated encoding scheme etc. and recent challenges in areas like reliability, security, and privacy for future research work.

References

1. Ibrahim S., He B., Jin H. Towards pay-as-you-consume cloud computing. *Proc. of the IEEE International Conference on Services Computing (SCC)*, 2011, pp. 370–377. https://doi.org/10.1109/scc.2011.38
2. Kumar P., Kumar R. Issues and challenges of load balancing techniques in cloud computing: A survey. *ACM Computing Surveys*, 2019, vol. 51, no. 6, pp. 120. https://doi.org/10.1145/3281010
3. Aljoumah E., Al-Mousawi F., Ahmad I., Al-Shammri M., Al-Jady Z. SLA in cloud computing architectures: A comprehensive study. *International Journal of Grid and Distributed Computing*, 2015, vol. 8, no. 5, pp. 7–32. https://doi.org/10.14257/ijgdc.2015.8.5.02
4. Weinman J. Cloud computing is NP-complete. *Proc. Tech. Symp. ITU Telecom World*, 2011, pp. 75–81.

Литература

1. Ibrahim S., He B., Jin H. Towards pay-as-you-consume cloud computing // Proc. of the IEEE International Conference on Services Computing (SCC). 2011. P. 370–377. https://doi.org/10.1109/scc.2011.38
2. Kumar P., Kumar R. Issues and challenges of load balancing techniques in cloud computing: A survey // ACM Computing Surveys. 2019. V. 51. N 6. P. 120. https://doi.org/10.1145/3281010
3. Aljoumah E., Al-Mousawi F., Ahmad I., Al-Shammri M., Al-Jady Z. SLA in cloud computing architectures: A comprehensive study // International Journal of Grid and Distributed Computing. 2015. V. 8. N 5. P. 7–32. https://doi.org/10.14257/ijgdc.2015.8.5.02
4. Weinman J. Cloud computing is NP-complete // Proc. Tech. Symp. ITU Telecom World. 2011. P. 75–81.

5. Fister Jr, I., Yang X.S., Fister I., Brest J., Fister D. A brief review of nature-inspired algorithms for optimization. *Elektrotehniski Vestnik*, 2013, vol. 80, no. 3, pp. 116–122.
6. Osman I.H., Kelly J.P. Meta-heuristics: an overview. *Meta-Heuristics, Springer*, 1996, pp. 1–21. https://doi.org/10.1007/978-1-4613-1361-8_1
7. Laporte G., Osman I.H. Routing problems: A bibliography. *Annals of Operations Research*, 1995, vol. 61, no. 1, pp. 227–262. https://doi.org/10.1007/bf02098290
8. Xu R., Wang Y., Cheng Y., Zhu Y., Xie Y., Sani A.S., Yuan D. Improved particle swarm optimization based workflow scheduling in cloud-fog environment. *Lecture Notes in Business Information Processing*, 2018, vol. 342, pp. 337–347. https://doi.org/10.1007/978-3-030-11641-5_27
9. Shubham, Gupta R., Gajera V., Jana P.K. An effective multi-objective workflow scheduling in cloud computing: A PSO based approach. *Proc. 9th International Conference on Contemporary Computing (IC3 2016)*, 2016, pp. 7880196. https://doi.org/10.1109/ic3.2016.7880196
10. Zhan S., Huo H. Improved PSO-based task scheduling algorithm in cloud computing. *Journal of Information & Computational Science*, 2012, vol. 9, no. 13, pp. 3821–3829.
11. Chen Z., Zhan Z., Lin Y., Gong Y., Gu T., Zhao F., Yuan H., Chen X., Li Q., Zhang J. Multiobjective cloud workflow scheduling: A multiple populations ant colony system approach. *IEEE Transactions on Cybernetics*, 2019, vol. 49, no. 8, pp. 2912–2926. https://doi.org/10.1109/tcyb.2018.2832640
12. Lal A., Krishna C.R. Critical path-based ant colony optimization for scientific workflow scheduling in cloud computing under deadline constraint. *Advances in Intelligent Systems and Computing*, 2018, vol. 696, pp. 447–461. https://doi.org/10.1007/978-981-10-7386-1_39
13. Sharma S., Jain R. EACO: an enhanced ant colony optimization algorithm for task scheduling in cloud computing. *International Journal of Security and Its Applications*, 2019, vol. 13, no. 4, pp. 91–100. https://doi.org/10.33832/ijsia.2019.13.4.09
14. Meshkati J., Safi-Esfahani F. Energy-aware resource utilization based on particle swarm optimization and artificial bee colony algorithms in cloud computing. *Journal of Supercomputing*, 2019, vol.75, no. 5, pp. 2455–2496. https://doi.org/10.1007/s11227-018-2626-9
15. Thanka M.R., Maheswari P.U., Edwin E.B. An improved efficient: Artificial Bee Colony algorithm for security and QoS aware scheduling in cloud computing environment. *Cluster Computing*, 2019, vol. 22, no. 5, pp. 10905–10913. https://doi.org/10.1007/s10586-017-1223-7
16. Yang X.S. Multiobjective firefly algorithm for continuous optimization. *Engineering with Computers*, 2013, vol. 29, no. 2, pp. 175–184. https://doi.org/10.1007/s00366-012-0254-1
17. Chakravarthi K.K., Shyamala L., Vaidehi V. Cost-effective workflow scheduling approach on cloud under deadline constraint using firefly algorithm. *Applied Intelligence*, 2021, vol. 51, no. 3, pp. 1629–1644. https://doi.org/10.1007/s10489-020-01875-1
18. Pradeep K., Jacob T.P. A hybrid approach for task scheduling using the cuckoo and harmony search in cloud computing environment. *Wireless Personal Communications*, 2018, vol. 101, no. 4, pp. 2287–2311. https://doi.org/10.1007/s11277-018-5816-0
19. Ghasemi S., Hanani A. A cuckoo-based workflow scheduling algorithm to reduce cost and increase load balance in the cloud environment. *International Journal on Informatics Visualization*, 2019, vol. 3, no. 1, pp. 79–85. https://doi.org/10.30630/joiv.3.1.220
20. Masadeh R., Sharieh A., Mahafzah B. Humpback whale optimization algorithm based on vocal behavior for task scheduling in cloud computing. *International Journal of Advanced Science and Technology*, 2019, vol. 13, no. 3, pp. 121–140.
21. Thennarasu S.R., Selvam M., Srihari K. A new whale optimizer for workflow scheduling in cloud computing environment. *Journal of Ambient Intelligence and Humanized Computing*, 2021, vol. 12, no. 3, pp. 3807–3814. https://doi.org/10.1007/s12652-020-01678-9
22. Hemasian-Etefagh F., Safi-Esfahani F. Dynamic scheduling applying new population grouping of whales meta-heuristic in cloud computing. *Journal of Supercomputing*, 2019, vol. 75, no. 10, pp. 6386–6450. https://doi.org/10.1007/s11227-019-02832-7
23. Kaur N., Singh S. A budget-constrained time and reliability optimization bat algorithm for scheduling workflow applications in clouds. *Procedia Computer Science*, 2016, vol. 98, pp. 199–204. https://doi.org/10.1016/j.procs.2016.09.032
24. Gu Y., Budati C. Energy-aware workflow scheduling and optimization in clouds using bat algorithm. *Future Generation Computer Systems*, 2020, vol. 113, pp. 106–112. https://doi.org/10.1016/j.future.2020.06.031

5. Fister Jr, I., Yang X.S., Fister I., Brest J., Fister D. A brief review of nature-inspired algorithms for optimization // Elektrotehniski Vestnik. 2013. V. 80. N 3. P. 116–122.
6. Osman I.H., Kelly J.P. Meta-heuristics: an overview // Meta-Heuristics. Springer, 1996. pp. 1–21. https://doi.org/10.1007/978-1-4613-1361-8_1
7. Laporte G., Osman I.H. Routing problems: A bibliography // Annals of Operations Research. 1995. V. 61. N 1. P. 227–262. https://doi.org/10.1007/bf02098290
8. Xu R., Wang Y., Cheng Y., Zhu Y., Xie Y., Sani A.S., Yuan D. Improved particle swarm optimization based workflow scheduling in cloud-fog environment // Lecture Notes in Business Information Processing. 2018. V. 342. P. 337–347. https://doi.org/10.1007/978-3-030-11641-5_27
9. Shubham, Gupta R., Gajera V., Jana P.K. An effective multi-objective workflow scheduling in cloud computing: A PSO based approach // Proc. 9th International Conference on Contemporary Computing (IC3 2016). 2016. P. 7880196. https://doi.org/10.1109/ic3.2016.7880196
10. Zhan S., Huo H. Improved PSO-based task scheduling algorithm in cloud computing // Journal of Information & Computational Science. 2012. V. 9. N 13. P. 3821–3829.
11. Chen Z., Zhan Z., Lin Y., Gong Y., Gu T., Zhao F., Yuan H., Chen X., Li Q., Zhang J. Multiobjective cloud workflow scheduling: A multiple populations ant colony system approach // IEEE Transactions on Cybernetics. 2019. V. 49. N 8. P. 2912–2926. https://doi.org/10.1109/tcyb.2018.2832640
12. Lal A., Krishna C.R. Critical path-based ant colony optimization for scientific workflow scheduling in cloud computing under deadline constraint // Advances in Intelligent Systems and Computing. 2018. V. 696. P. 447–461. https://doi.org/10.1007/978-981-10-7386-1_39
13. Sharma S., Jain R. EACO: an enhanced ant colony optimization algorithm for task scheduling in cloud computing // International Journal of Security and Its Applications. 2019. V. 13. N 4. P. 91–100. https://doi.org/10.33832/ijsia.2019.13.4.09
14. Meshkati J., Safi-Esfahani F. Energy-aware resource utilization based on particle swarm optimization and artificial bee colony algorithms in cloud computing // Journal of Supercomputing. 2019. V. 75. N 5. P. 2455–2496. https://doi.org/10.1007/s11227-018-2626-9
15. Thanka M.R., Maheswari P.U., Edwin E.B. An improved efficient: Artificial Bee Colony algorithm for security and QoS aware scheduling in cloud computing environment // Cluster Computing. 2019. V 22. N 5. P. 10905–10913. https://doi.org/10.1007/s10586-017-1223-7
16. Yang X.S. Multiobjective firefly algorithm for continuous optimization // Engineering with Computers. 2013. V. 29. N 2. P. 175–184. https://doi.org/10.1007/s00366-012-0254-1
17. Chakravarthi K.K., Shyamala L., Vaidehi V. Cost-effective workflow scheduling approach on cloud under deadline constraint using firefly algorithm // Applied Intelligence. 2021. V. 51. N 3. P. 1629–1644. https://doi.org/10.1007/s10489-020-01875-1
18. Pradeep K., Jacob T.P. A hybrid approach for task scheduling using the cuckoo and harmony search in cloud computing environment // Wireless Personal Communications. 2018. V. 101. N 4. P. 2287–2311. https://doi.org/10.1007/s11277-018-5816-0
19. Ghasemi S., Hanani A. A cuckoo-based workflow scheduling algorithm to reduce cost and increase load balance in the cloud environment // International Journal on Informatics Visualization. 2019. V. 3. N 1. P. 79–85. https://doi.org/10.30630/joiv.3.1.220
20. Masadeh R., Sharieh A., Mahafzah B. Humpback whale optimization algorithm based on vocal behavior for task scheduling in cloud computing // International Journal of Advanced Science and Technology. 2019. V. 13. N 3. P. 121–140.
21. Thennarasu S.R., Selvam M., Srihari K. A new whale optimizer for workflow scheduling in cloud computing environment // Journal of Ambient Intelligence and Humanized Computing. 2021. V. 12. N 3. P. 3807–3814. https://doi.org/10.1007/s12652-020-01678-9
22. Hemasian-Etefagh F., Safi-Esfahani F. Dynamic scheduling applying new population grouping of whales meta-heuristic in cloud computing // Journal of Supercomputing. 2019. V. 75. N 10. P. 6386–6450. https://doi.org/10.1007/s11227-019-02832-7
23. Kaur N., Singh S. A budget-constrained time and reliability optimization bat algorithm for scheduling workflow applications in clouds // Procedia Computer Science. 2016. V. 98. P. 199–204. https://doi.org/10.1016/j.procs.2016.09.032
24. Gu Y., Budati C. Energy-aware workflow scheduling and optimization in clouds using bat algorithm // Future Generation Computer Systems. 2020. V. 113. P. 106–112. https://doi.org/10.1016/j.future.2020.06.031

25. Ullah A., Nawi N.M., Khan M.H. BAT algorithm used for load balancing purpose in cloud computing: an overview. *International Journal of High Performance Computing and Networking*, 2020, vol. 16, no. 1, pp. 43–54. https://doi.org/10.1504/ijhpcn.2020.110258
26. Abdel-Basset M., Abdel-Fatah L., Sangaiah A.K. Metaheuristic algorithms: A comprehensive review. *Computational Intelligence for Multimedia Big Data on The Cloud With Engineering Applications*, 2018, pp. 185–231. https://doi.org/10.1016/b978-0-12-813314-9.00010-4
27. Yang X.S. *Nature-Inspired Metaheuristic Algorithms*. Luniver press, 2010, 148 p.
28. Iranmanesh A., Naji H.R. DCHG-TS: a deadline-constrained and cost-effective hybrid genetic algorithm for scientific workflow scheduling in cloud computing. *Cluster Computing*, 2021, vol. 24, no. 2, pp. 667–681. https://doi.org/10.1007/s10586-020-03145-8
29. Zhou Z., Li F., Zhu H., Xie H., Abawajy J.H., Chowdhury M.U. An improved genetic algorithm using greedy strategy toward task scheduling optimization in cloud environments. *Neural Computing and Applications*, 2020, vol. 32, no. 6, pp. 1531–1541. https://doi.org/10.1007/s00521-019-04119-7
30. Aziza H., Krichen S. A hybrid genetic algorithm for scientific workflow scheduling in cloud environment. *Neural Computing and Applications*, 2020, vol. 32, no. 18, pp. 15263–15278. https://doi.org/10.1007/s00521-020-04878-8
31. Sardaraz M., Tahir M. A parallel multi-objective genetic algorithm for scheduling scientific workflows in cloud computing. *International Journal of Distributed Sensor Networks*, 2020, vol. 16, no. 8. https://doi.org/10.1177/1550147720949142
32. Pang S., Li W., He H., Shan Z., Wang X. An EDA-GA hybrid algorithm for multi-objective task scheduling in cloud computing. *IEEE Access*, 2019, vol. 7, pp. 146379–146389. https://doi.org/10.1109/access.2019.2946216
33. Wangsom P., Lavangnananda K., Bouvry P. Multi-objective scientific-workflow scheduling with data movement awareness in cloud. *IEEE Access*, 2019, vol. 7, pp. 177063–177081. https://doi.org/10.1109/access.2019.2957998
34. Hammed S.S., Arunkumar B. Efficient workflow scheduling in cloud computing for security maintenance of sensitive data. *International Journal of Communication Systems*, 2019, pp. e4240. https://doi.org/10.1002/dac.4240
35. Gupta I., Gupta S., Choudhary A., Jana P.K. A hybrid meta-heuristic approach for load balanced workflow scheduling in IaaS cloud. *Lecture Notes in Computer Science*, 2019, vol. 11319, pp. 73–89. https://doi.org/10.1007/978-3-030-05366-6_6
36. Rehman A., Hussain S.S., ur Rehman Z., Zia S., Shamshirband S. Multi-objective approach of energy efficient workflow scheduling in cloud environments. *Concurrency and Computation: Practice and Experience*, 2019, vol. 31, no. 8, pp. e4949. https://doi.org/10.1002/cpe.4949
37. Ashouraei M., Khezr S.N., Benlamri R., Navimipour N.J. A new SLA-aware load balancing method in the cloud using an improved parallel task scheduling algorithm. *Proc. 6th International Conference on Future Internet of Things and Cloud (FiCloud)*, 2018, pp. 71–76. https://doi.org/10.1109/ficloud.2018.00018
38. Jana B., Poray J. A hybrid task scheduling approach based on genetic algorithm and particle swarm optimization technique in cloud environment. *Advances in Intelligent Systems and Computing*, 2018, vol. 695, pp. 607–614. https://doi.org/10.1007/978-981-10-7566-7_61
39. Shishido H.Y., Estrella J.C., Toledo C.F.M., Arantes M.S. Genetic-based algorithms applied to a workflow scheduling algorithm with security and deadline constraints in clouds. *Computers & Electrical Engineering*, 2018, vol. 69, pp. 378–394. https://doi.org/10.1016/j.compeleceng.2017.12.004
40. Kaur G., Kalra M. Deadline constrained scheduling of scientific workflows on cloud using hybrid genetic algorithm. Proc. 7th International Conference on Cloud Computing, Data Science and Engineering, Confluence, 2017, pp. 276–280. https://doi.org/10.1109/confluence.2017.7943162
41. Wangsom P., Lavangnananda K., Bouvry P. The application of nondominated sorting genetic algorithm (NSGA-III) for scientific-workflow scheduling on cloud. *Proc. 8th Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA 2017)*, 2017, pp. 269–287.
42. Meena J., Kumar M., Vardhan M. Cost effective genetic algorithm for workflow scheduling in cloud under deadline constraint. *IEEE Access*, 2016, vol. 4, pp. 5065–5082. https://doi.org/10.1109/access.2016.2593903

25. Ullah A., Nawi N.M., Khan M.H. BAT algorithm used for load balancing purpose in cloud computing: an overview // International Journal of High Performance Computing and Networking. 2020. V. 16. N 1. P. 43–54. https://doi.org/10.1504/ijhpcn.2020.110258
26. Abdel-Basset M., Abdel-Fatah L., Sangaiah A.K. Metaheuristic algorithms: A comprehensive review // Computational Intelligence for Multimedia Big Data on The Cloud with Engineering Applications. 2018. P. 185–231. https://doi.org/10.1016/b978-0-12-813314-9.00010-4
27. Yang X.S. Nature-Inspired Metaheuristic Algorithms. Luniver press, 2010. 148 p.
28. Iranmanesh A., Naji H.R. DCHG-TS: a deadline-constrained and cost-effective hybrid genetic algorithm for scientific workflow scheduling in cloud computing // Cluster Computing. 2021. V. 24. N 2. P. 667–681. https://doi.org/10.1007/s10586-020-03145-8
29. Zhou Z., Li F., Zhu H., Xie H., Abawajy J.H., Chowdhury M.U. An improved genetic algorithm using greedy strategy toward task scheduling optimization in cloud environments // Neural Computing and Applications. 2020. V. 32. N 6. P. 1531–1541. https://doi.org/10.1007/s00521-019-04119-7
30. Aziza H., Krichen S. A hybrid genetic algorithm for scientific workflow scheduling in cloud environment // Neural Computing and Applications. 2020. V. 32. N 18. P. 15263–15278. https://doi.org/10.1007/s00521-020-04878-8
31. Sardaraz M., Tahir M. A parallel multi-objective genetic algorithm for scheduling scientific workflows in cloud computing // International Journal of Distributed Sensor Networks. 2020. V. 16. N 8. https://doi.org/10.1177/1550147720949142
32. Pang S., Li W., He H., Shan Z., Wang X. An EDA-GA hybrid algorithm for multi-objective task scheduling in cloud computing // IEEE Access. 2019. V. 7. P. 146379–146389. https://doi.org/10.1109/access.2019.2946216
33. Wangsom P., Lavangnananda K., Bouvry P. Multi-objective scientific-workflow scheduling with data movement awareness in cloud // IEEE Access. 2019. V. 7. P. 177063–177081. https://doi.org/10.1109/access.2019.2957998
34. Hammed S.S., Arunkumar B. Efficient workflow scheduling in cloud computing for security maintenance of sensitive data // International Journal of Communication Systems. 2019. P. e4240. https://doi.org/10.1002/dac.4240
35. Gupta I., Gupta S., Choudhary A., Jana P.K. A hybrid meta-heuristic approach for load balanced workflow scheduling in IaaS cloud // Lecture Notes in Computer Science. 2019. V. 11319. P. 73–89. https://doi.org/10.1007/978-3-030-05366-6_6
36. Rehman A., Hussain S.S., ur Rehman Z., Zia S., Shamshirband S. Multi-objective approach of energy efficient workflow scheduling in cloud environments // Concurrency and Computation: Practice and Experience. 2019. V. 31. N 8. P. e4949. https://doi.org/10.1002/cpe.4949
37. Ashouraei M., Khezr S.N., Benlamri R., Navimipour N.J. A new SLA-aware load balancing method in the cloud using an improved parallel task scheduling algorithm // Proc. 6th International Conference on Future Internet of Things and Cloud (FiCloud). 2018. P. 71–76. https://doi.org/10.1109/ficloud.2018.00018
38. Jana B., Poray J. A hybrid task scheduling approach based on genetic algorithm and particle swarm optimization technique in cloud environment // Advances in Intelligent Systems and Computing. 2018. V. 695. P. 607–614. https://doi.org/10.1007/978-981-10-7566-7_61
39. Shishido H.Y., Estrella J.C., Toledo C.F.M., Arantes M.S. Genetic-based algorithms applied to a workflow scheduling algorithm with security and deadline constraints in clouds // Computers & Electrical Engineering. 2018. V. 69. P. 378–394. https://doi.org/10.1016/j.compeleceng.2017.12.004
40. Kaur G., Kalra M. Deadline constrained scheduling of scientific workflows on cloud using hybrid genetic algorithm // Proc. 7th International Conference on Cloud Computing, Data Science and Engineering, Confluence. 2017. P. 276–280. https://doi.org/10.1109/confluence.2017.7943162
41. Wangsom P., Lavangnananda K., Bouvry P. The application of nondominated sorting genetic algorithm (NSGA-III) for scientific-workflow scheduling on cloud // Proc. 8th Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA 2017). 2017. P. 269–287.
42. Meena J., Kumar M., Vardhan M. Cost effective genetic algorithm for workflow scheduling in cloud under deadline constraint // IEEE Access. 2016. V. 4. P. 5065–5082. https://doi.org/10.1109/access.2016.2593903

43. Liu L., Zhang M., Buyya R., Fan Q. Deadline-constrained coevolutionary genetic algorithm for scientific workflow scheduling in cloud computing. *Concurrency and Computation: Practice and Experience*, 2017, vol. 29, no. 5, pp. e3942. https://doi.org/10.1002/cpe.3942
44. Chen Z.G., Du K.J., Zhan Z.H., Zhang J. Deadline constrained cloud computing resources scheduling for cost optimization based on dynamic objective genetic algorithm. *Proc. 2015 IEEE Congress on Evolutionary Computation (CEC)*, 2015, pp. 708–714. https://doi.org/10.1109/cec.2015.7256960
45. Xu Y., Li K., Hu J., Li K. A genetic algorithm for task scheduling on heterogeneous computing systems using multiple priority queues. *Information Sciences*, 2014, vol. 270, pp. 255–287. https://doi.org/10.1016/j.ins.2014.02.122
46. Delavar A.G., Aryan Y. HSGA: a hybrid heuristic algorithm for workflow scheduling in cloud systems. *Cluster Computing*, 2014, vol. 17, no. 1, pp. 129–137. https://doi.org/10.1007/s10586-013-0275-6
47. Wang T., Liu Z., Chen Y., Xu Y., Dai X. Load balancing task scheduling based on genetic algorithm in cloud computing. *Proc. 12th International Conference on Dependable, Autonomic and Secure Computing (DASC)*, 2014, pp. 146–152. https://doi.org/10.1109/DASC.2014.35
48. Dasgupta K., Mandal B., Dutta P., Mandal J.K., Dam S. A genetic algorithm (GA) based load balancing strategy for cloud computing. *Procedia Technology*, 2013, vol. 10. pp. 340–347. https://doi.org/10.1016/j.protcy.2013.12.369
49. Ying C.-T., Yu J. Energy-aware genetic algorithms for task scheduling in cloud computing. *Proc. 7th ChinaGrid Annual Conference (ChinaGrid 2012)*, 2012, pp. 43–48. https://doi.org/10.1109/chinagrid.2012.15
50. Zhao E.-D., Qi Y.-Q., Xiang X.-X., Chen Y. A data placement strategy based on genetic algorithm for scientific workflows. *Proc. 8th International Conference on Computational Intelligence and Security (CIS 2012)*, 2012, pp. 146–149. https://doi.org/10.1109/cis.2012.40
51. Barrett E., Howley E., Duggan J. A learning architecture for scheduling workflow applications in the cloud. *Proc. 9th European Conference on Web Services (ECOWS 2011)*, 2011, pp. 83–90. https://doi.org/10.1109/ecows.2011.27
52. Kessaci Y., Melab N., Talbi E.-G. A pareto-based GA for scheduling HPC applications on distributed cloud infrastructures. *Proc. of the International Conference on High Performance Computing and Simulation (HPCS 2011)*, 2011, pp. 456–462. https://doi.org/10.1109/hpcsim.2011.5999860
53. Singhal S., Patel J. Load balancing scheduling algorithm for concurrent workflow. *Computing and Informatics*, 2018, vol. 37, no. 2, pp. 311–326. https://doi.org/10.4149/cai_2018_2_311
54. Bansal S., Hota C. Efficient Algorithm on heterogeneous computing system. *Proc. of the International Conference on Recent Trends in Information Systems (ReTIS 2011)*, 2011, pp. 57–61. https://doi.org/10.1109/retis.2011.6146840
55. Khoruzhnikov S.E., Shevel A.Y. Management system for scalable geographically distributed data center. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2019, vol. 19, no. 5, pp. 931–938. (in Russian). https://doi.org/10.17586/2226-1494-2019-19-5-931-938
56. Becker M., Gatchin Yu., Karmanovsky N., Terentiev A., Fyodorov D. Information security in cloud computing: problems and prospects. *Scientific and Technical Bulletin of St. Petersburg State University of Information Technologies, Mechanics and Optics*, 2011, no. 1(71), pp. 97–102. (in Russian)
57. Khan D.V., Razgulyaev K.A., Tyagunov D.M. Web portals for management of cloud services within data centres. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2019, vol. 19, no. 6, pp. 1169–1171. (in Russian). https://doi.org/10.17586/2226-1494-2019-19-6-1169-1171

43. Liu L., Zhang M., Buyya R., Fan Q. Deadline-constrained coevolutionary genetic algorithm for scientific workflow scheduling in cloud computing // Concurrency and Computation: Practice and Experience. 2017. V. 29. N 5. P. e3942. https://doi.org/10.1002/cpe.3942
44. Chen Z.G., Du K.J., Zhan Z.H., Zhang J. Deadline constrained cloud computing resources scheduling for cost optimization based on dynamic objective genetic algorithm // Proc. 2015 IEEE Congress on Evolutionary Computation (CEC). 2015. P. 708–714. https://doi.org/10.1109/cec.2015.7256960
45. Xu Y., Li K., Hu J., Li K. A genetic algorithm for task scheduling on heterogeneous computing systems using multiple priority queues // Information Sciences. 2014. V. 270. P. 255–287. https://doi.org/10.1016/j.ins.2014.02.122
46. Delavar A.G., Aryan Y. HSGA: a hybrid heuristic algorithm for workflow scheduling in cloud systems // Cluster Computing. 2014. V. 17. N 1. P. 129–137. https://doi.org/10.1007/s10586-013-0275-6
47. Wang T., Liu Z., Chen Y., Xu Y., Dai X. Load balancing task scheduling based on genetic algorithm in cloud computing // Proc. 12th International Conference on Dependable, Autonomic and Secure Computing (DASC). 2014. P. 146–152. https://doi.org/10.1109/DASC.2014.35
48. Dasgupta K., Mandal B., Dutta P., Mandal J.K., Dam S. A genetic algorithm (GA) based load balancing strategy for cloud computing // Procedia Technology. 2013. V. 10. P. 340–347. https://doi.org/10.1016/j.protcy.2013.12.369
49. Ying C.-T., Yu J. Energy-aware genetic algorithms for task scheduling in cloud computing // Proc. 7th ChinaGrid Annual Conference (ChinaGrid 2012). 2012. P. 43–48. https://doi.org/10.1109/chinagrid.2012.15
50. Zhao E.-D., Qi Y.-Q., Xiang X.-X., Chen Y. A data placement strategy based on genetic algorithm for scientific workflows // Proc. 8th International Conference on Computational Intelligence and Security (CIS 2012). 2012. P. 146–149. https://doi.org/10.1109/cis.2012.40
51. Barrett E., Howley E., Duggan J. A learning architecture for scheduling workflow applications in the cloud // Proc. 9th European Conference on Web Services (ECOWS 2011). 2011. P. 83–90. https://doi.org/10.1109/ecows.2011.27
52. Kessaci Y., Melab N., Talbi E.-G. A pareto-based GA for scheduling HPC applications on distributed cloud infrastructures // Proc. of the International Conference on High Performance Computing and Simulation (HPCS 2011). 2011. P. 456–462. https://doi.org/10.1109/hpcsim.2011.5999860
53. Singhal S., Patel J. Load balancing scheduling algorithm for concurrent workflow // Computing and Informatics. 2018. V. 37. N 2. P. 311–326. https://doi.org/10.4149/cai_2018_2_311
54. Bansal S., Hota C. Efficient Algorithm on heterogeneous computing system // Proc. of the International Conference on Recent Trends in Information Systems (ReTIS 2011). 2011. P. 57–61. https://doi.org/10.1109/retis.2011.6146840
55. Хоружников С.Э., Шевель А.Е. Система управления масштабируемым географически распределенным центром обработки данных // Научно-технический вестник информационных технологий, механики и оптики. 2019. Т. 19. № 5. С. 931–938. https://doi.org/10.17586/2226-1494-2019-19-5-931-938
56. Беккер М.Я., Гатчин Ю.А., Кармановский Н.С., Терентьев А.О., Федоров Д.Ю. Информационная безопасность при облачных вычислениях: проблемы и перспективы // Научно-технический вестник Санкт-петербургского государственного университета информационных технологий, механики и оптики. 2011. № 1(71). С. 97–102.
57. Хан Д.В., Разгуляев К.А., Тягунов Д.М. Разработка порталов для управления облачными сервисами в составе центра обработки и хранения данных // Научно-технический вестник информационных технологий, механики и оптики. 2019. Т. 19. № 6. С. 1169–1171. https://doi.org/10.17586/2226-1494-2019-19-6-1169-1171

Authors

Sandeep Kumar Bothra — M.Tech., Researcher, Manipal University Jaipur, Jaipur, Rajasthan, 303007, India, http://orcid.org/0000-0003-0555-569X, bothrajain@gmail.com

Sunita Singhal — PhD, Associate Professor, Associate Professor, Manipal University Jaipur, Jaipur, Rajasthan, 303007, India, SC 57194063002, http://orcid.org/0000-0003-2462-8102, sunita.singhal@jaipur.manipal.edu

Авторы

Сандип Кумар Ботра — магистр технологий, исследователь, Университет Манипала в Джайпуре, Джайпур, Раджастан, 303007, Индия, http://orcid.org/0000-0003-0555-569X, bothrajain@gmail.com

Сунита Сингхал — PhD, доцент, доцент, Университет Манипала в Джайпуре, Джайпур, Раджастан, 303007, Индия, SC 57194063002, http://orcid.org/0000-0003-2462-8102, sunita.singhal@jaipur.manipal.edu

Received 25.06.2021
Approved after reviewing 05.07.2021
Accepted 30.07.2021

Статья поступила в редакцию 25.06.2021
Одобрена после рецензирования 05.07.2021
Принята к печати 30.07.2021