МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

**"САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО"**

Институт компьютерных наук и технологий

Направление **02.03.01** : Математика и компьютерные науки

Literature Review:

«Modern Scheduling Algorithms in Cloud-Fog and High-Performance Computing Environments»

Student: Darya Yashnova 5130201/20002

Teacher: Ph.D, Motorin D. E.

Saint-Petersburg, 2024

1 Analysis

Research in the field of optimizing task scheduling in cloud computing has significantly increased in response to the growing requirements for cloud applications and the need for efficient use of network resources. Many approaches have been proposed, each with its own advantages and limitations. This review examines seven key articles that contribute to this area of research.

1.1 Heuristic approaches

Heuristic algorithms are often used to solve the task scheduling optimization problem.

The WOA(Whale Optimization Algorithm), a metaheuristic algorithm inspired by the hunting behavior of humpback whales, is often used to optimize task scheduling algorithms [2], [5], [6].

The PSO(Particle Swarm Optimization) heuristic is also used — an algorithm based on swarm behavior that searches for the optimal solution by moving in a multi-dimensional solution space with a swarm of particles [2], [3], [5], [7]. These algorithms are often tested on the Cybershake, Montage, Epigenomics, Sipht, and Inspiral streams [2], [7]. Fitness functions are often used to evaluate the effectiveness of optimization algorithms[2], [5], [6], [7].

1.2 DRL and expert policies

DRL (Deep Reinforcement Learning) is an algorithm that automatically selects the optimal policy. Its main idea is to obtain rewards through interaction between the agent and the environment in order to maximize returns and achieve specific optimization goals [4], [5]. Several DRL algorithms based on expert knowledge have been developed [4], [5]. DRL can optimize task scheduling in conjunction with WOA [5] and through a network of experts [4].

1.3 Optimization Parameters

In addition to performance metrics such as runtime, cloud computing systems also face the challenge of minimizing power consumption. Several studies emphasize the importance of energy-efficient task optimization [2], [3], [5], [6]. Some researches discuss methods and algorithms designed to reduce energy consumption while maintaining a high level of performance. Various algorithms evaluate metrics such as execution time [7], [1], [4], [6] or average waiting time [1], [4], cost [2], [3], [7]. Optimization of tasks in cloud computing often involves balancing competing goals, so often several performance parameters are evaluated at once.

1.4 Conclusion

Optimizing task scheduling in cloud computing is a multi-faceted task that requires a comprehensive approach. Heuristic algorithms, such as WOA and PSO, demonstrate high efficiency in solving time planning problems. DRL algorithms based on expert knowledge offer innovative approaches to automatically selecting the optimal security policy. Energy efficiency is also an important aspect that requires special attention. Research in this area is ongoing, and further developments can lead to significant improvements in the performance and efficiency of cloud systems.

References

[1] Yellamma Pachipala, Kavya Sri Sureddy, A.B.S. Sriya Kaitepalli, Nagalakshmi Pagadala, Sai Satwik Nalabothu, Mihir Iniganti (2024)
Optimizing Task Scheduling in Cloud Computing: An Enhanced Shortest Job First Algorithm, doi: https://doi.org/10.1016/j.procs.2024.03.250, (ICIDCA 2024).

[2] Sumit Bansal, Himanshu Aggarwal (2023)
A Hybrid Particle Whale Optimization Algorithm with application to workflow scheduling in cloud–fog environment,
doi: https://doi.org/10.1016/j.dajour.2023.100361, Decision Analytics Journal.

[3] Briag Dupont, Nesryne Mejri, Georges Da Costa (2020)
Energy-aware scheduling of malleable HPC applications using a Particle Swarm optimised greedy algorithm, doi: https://doi.org/10.1016/j.suscom.2020.100447, Sustainable Computing: Informatics and Systems.

[4] Jingbo Li, Xingjun Zhang, Jia Wei, Zeyu Ji, Zheng Wei (2022)
GARLSched: Generative adversarial deep reinforcement learning task scheduling optimization for large-scale high performance computing systems,
doi: https://doi.org/10.1016/j.future.2022.04.032, Future Generation Computer Systems.

[5] Ting Shu, Zhijie Pan, Zuohua Ding, Zhangqing Zu (2024)
Resource scheduling optimization for industrial operating system using deep reinforcement learning and WOA algorithm,
doi: https://doi.org/10.1016/j.eswa.2024.124765, Expert Systems With Applications.

[6] Sudheer Mangalampalli, Ganesh Reddy Karri, G Naga Satish (2023)
Efficient Workflow Scheduling algorithm in cloud computing using Whale Optimization,
doi: https://doi.org/10.1016/j.procs.2023.01.170, Workflow Scheduling algorithm in cloud computing.

[7] Haithem Hafsi, Hamza Gharsellaoui, Sadok Bouamama (2022)
Genetically-modified Multi-objective Particle Swarm Optimization approach for high-performance computing workflow scheduling ,
doi: https://doi.org/10.1016/j.asoc.2022.108791,Applied Soft Computing.

Appendix

Table 1

Research Paper	Heuristic methods	Prioritization	Math methods	Comparative studies	Optimization Parameters
Optimizing Task Scheduling in Cloud Computing: An Enhanced Shortest Job First Algorithm		Dynamic prioritization	SJF	+	Task completion time, Resource utilization
A Hybrid Particle Whale Optimization Algorithm with application to workflow scheduling in cloud–fog environment	PWOA		Fitness Function	+	TET, TEC
Energy-aware scheduling of malleable HPC applications using a Particle Swarm optimised greedy algorithm	PSO	FIFO	PSO, Greedy algotithm, Friedman's test	+	Task Reorganization parameters, Server Shutdown Options
GARLSched: Generative adversarial deep reinforcement learning task scheduling optimization for large-scale high performance computing systems	Expert policies: F1, F2, F3, F4, WPT, UNICEF		DRL, GAN, MDP	+	AVGwt, AVGbsld
Resource scheduling optimization for industrial operating system using deep reinforcement learning and WOA algorithm	WOA	Service metrics	DRL, QoS evaluation Normalization and linear reduction of parameters	+	Convergence speed, solution quality, performance stability

Research Paper	Heuristic methods	Prioritization	Math methods	Comparative studies	Optimization Parameters
Efficient Workflow Scheduling algorithm in cloud computing using Whale Optimization	WOA	Task's size, execution time	Fitness function	+	Migrationtime, energy consumption, makespan
Genetically-modified Multi-objective Particle Swarm Optimization approach for high-performance computing workflow scheduling	PSO, genetic algorithms	Time and cost to complete the task	Goal functions: Min RunTime(S), Min Cost(S), Crossover operations	+	Inverted Generational Distance (IGD), Hypervolume (HV)

Research results

Research Paper	Testing environment	Testing flows	Compared algorithms	Winning amount
Optimizing Task Scheduling in Cloud Computing: An Enhanced Shortest Job First Algorithm	CloudSim simulation framework	«Light», «Moderate» and «Heavy» workloads	SJF, RoundRobin, FCFS, **MSJF**	The results showed that the modified SJF has a lower average waiting time on 20-40%, which indicates its effectiveness in the operational allocation of resources.
A Hybrid Particle Whale Optimization Algorithm with application to workflow scheduling in cloud–fog environment	–	Cybershake, Montage, Epigenomics, Sipht and Inspiral	WOA, PSO, **PWOA**	The average TEC reduced from 8.64% to 22.68%compared to the PSO algorithm. Compared to the standard WOA algorithm, the reductions are from 30.43% to 71.31%.
Energy-aware scheduling of malleable HPC applications using a Particle Swarm optimised greedy algorithm	Simulation environment was developed using python	–	FIFO, FIFO-Rcfg, FIFO-Poff, FIFO-Rcfg-Poff, Rand-Param1, Rand-Param2, Rand-Param3, **Swarm1, Swarm2, Swarm3**	Results of the resilience test as a percentage for each scenario: - Swarm2: 1st place (100%) - FIFO: 2nd place (90%) - Swarm3, FIFO-Rcfg, Swarm1, FIFO-Poff: 3-rd place (80%) - FIFO-Rcfg-Poff, Rand-Param1, Rand-Param3: 4-rd place(70%) - Rand - Param2: 5th place (60%)
GARLSched: Generative adversarial deep reinforcement learning task scheduling optimization for large-scale high performance computing systems	–	Lublin-256, HPC2N, SDSC-BLUE, SDSC-SP2	DRL,MLP, **GARLSCHED**	For the AVGwt and AVGbsld metrics, in terms of mean value, the GARLSched achieved the highest performance in all workloads, from 41% to 66% indicating that the algorithm achieves better performance under the high load of medium-scale and large-scale computing resources.

Research results

Research Paper	Testing environment	Testing flows	Compared algorithms	Features of the experiment results
Resource scheduling optimization for industrial operating system using deep reinforcement learning and WOA algorithm	CEC2017 test suite	–	GA, PSO, GWO, BAT, MFO, BOA, WOA, EWOA, RLGWO, RLWOA , **DWOA**	Comparative experiments were conducted with 11 known algorithms based on CEC2017 control functions and RIS problems. Results show significant advantages in convergence speed, solution quality, and performance stability. Wilcoxon rank test and Friedman test confirm DWOA superiority.
Efficient Workflow Scheduling algorithm in cloud computing using Whale Optimization	Workflowsim	–	PSO, ACO, CS, GA, **PWhale**	Modeling has shown that the proposed algorithm minimizes execution time, migration, and power consumption compared to existing approaches. However, there is a disadvantage – it is impossible to predict the number and type of upcoming tasks
Genetically-modified Multi-objective Particle Swarm Optimization approach for high-performance computing workflow scheduling	CloudSim toolkit	Cybershake, Montage, Epigenomics, Sipht and Inspiral	SMPSO,OMOPSO, NSGAII, **GMPSO**	Well-known scientific processes were modeled. Although the main emphasis was placed on high-performance workflows, it should be noted that the gain for small workflow sizes was negligible compared to the results obtained for large workflows. However, GMPSO algorithm can be adapted to work processes of any size.