12-7-2022

Investigating different optimization criteria for a hybrid job scheduling approach based on heuristics and metaheuristics

Abhijith Remesh
VLBA lab, OVGU, abhijith.remesh@ovgu.de

Abdulrahman Nahhas
VLBA lab, OVGU, abdulrahman.nahhas@ovgu.de

Andrey Kharitonov
VLBA lab - OVGU, andrey.kharitonov@ovgu.de

Klaus Turowski
VLBA lab - OVGU, klaus.turowski@ovgu.de

Investigating different optimization criteria for a hybrid job scheduling approach based on heuristics and metaheuristics

Full research paper

Abhijith Remesh
Very Large Business Applications Lab
Otto von Guericke University
Magdeburg, Germany
Email: abhijith.remesh@ovgu.de

Abdulrahman Nahhas
Very Large Business Applications Lab
Otto von Guericke University
Magdeburg, Germany
Email: abdulrahman.nahhas@ovgu.de

Andrey Kharitonov
Very Large Business Applications Lab
Otto von Guericke University
Magdeburg, Germany
Email: andrey.kharitonov@ovgu.de

Klaus Turowski
Very Large Business Applications Lab
Otto von Guericke University
Magdeburg, Germany
Email: klaus.turowski@ovgu.de

Abstract

The Information Technology industry has revolutionized through the advent of cloud computing as the cloud offers dynamic computing utilities to global users. The performance of cloud computing services depends on the process of job scheduling. There has been a great research focus on the different amalgamation of heuristics with meta-heuristics (hybrid scheduling approaches) in the cloud computing scheduling context with the aim of optimizing several performance metrics. This paper discusses a hybrid job scheduling approach that intends to optimize the performance metrics namely makespan, average flow time, average waiting time, and throughput. The main focus of this paper is to evaluate this hybrid job scheduling approach based on different optimization criteria which includes single-objective and multi-objectives functions based on the aforementioned performance metrics on different large-scale problem instances. This helps us to investigate and identify the best optimization criteria for the hybrid job scheduling approach.

Keywords Heuristics, Meta-heuristics, task scheduling, cloud computing, optimization criteria

1 Introduction

Cloud computing technology is derived from the predecessor technologies such as grid and distributed computing. Cloud computing enabled the distribution of different kinds of computing resources such as hardware, software, platforms, storage and network to the world-wide consumers over the internet as utility services (Aladwani 2020). NIST defines cloud computing as follows, "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" (Mell and Grance 2011). This cloud computing model consists of five essential characteristics, three service models and four deployment models (Mell and Grance 2011). The key features of cloud computing include on-demand self-service, wide range of network connectivity, massive pool of computing resources with rapid elasticity, scalability and pay-per-use business model (Jain and Upadhyay 2017). The Cloud computing paradigm has three kinds of service models and four types of deployment models. The service models are namely Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). The four deployment models include private cloud, community cloud, public cloud, and hybrid cloud (Nagadevi et al. 2013). The performance of cloud computing systems depends on resource allocation and appropriate scheduling (Venu 2020). Any unit of work which is to be finished within a certain period of time is known as a task or a job. Task scheduling in general is considered as a NP complete problem and a good task scheduling approach must map tasks to VMs in optimal and efficient way. The process of task scheduling can be performed with respect to the optimization on specific performance metrics (Mathew et al. 2014). In cloud computing environment, the scheduling process happens at both the host level and VM level. The scheduling at the host level where the VMs are assigned among the hosts is known as VM allocation whereas the scheduling at the VM level where the tasks are mapped among the VMs is known as task scheduling (Soltani et al. 2017). Furthermore, the allocation of tasks to resources or VMs are performed with the aim of optimizing certain performance metrics such as makespan, execution time, flow time and so on. This paper focuses on the field of task scheduling among the VMs.

In the scientific community, several scheduling approaches has been proposed over the last two decades. Certain research articles discuss the generic baseline scheduling heuristics and their improved versions, certain research papers discuss the metaheuristic-based scheduling heuristics, and some discuss the different amalgamations of heuristics and metaheuristics scheduling methods. Different scheduling heuristics exhibit better performance with respect to different metrics. Performance metrics can be either conflicting or complementing in nature. Thus, there are chances that combing several scheduling heuristics can give satisfactory performance over different metrics. Hence, several research papers proposed hybrid scheduling strategies that combines base-line scheduling heuristics with metaheuristics optimization methods. This paper translates the hybrid design discussed in the research paper (Nahhas et al. 2021) into the task scheduling context as a hybrid adaptive scheduling approach. This approach incorporates several generic baseline scheduling heuristics within a metaheuristic optimization method, genetic algorithm. This paper intends to investigate this approach with respect to the scheduling-oriented metrics such as makespan, average flow time, average waiting time and throughput. Every metaheuristic optimization method works based on an objective function. The optimization potential and performance results depend on the objective function of the metaheuristic technique. This paper evaluates the proposed hybrid scheduling approach on stated metrics by investigating the genetic algorithm-based hybrid scheduling approach on different objective functions and intends to identify the best objective function with better optimization capability which gives better performance over all metrics. Moreover, most of the scheduling strategies discussed in the scientific community are evaluated using small-scale user synthetic workloads. This paper investigates and evaluates the proposed hybrid scheduling approach by using real-world parallel workload logs that resembles large-scale problem instances. The process of scheduling can become a bottleneck as the workload increases and the process of scheduling may also get impacted based on the nature and intensity of the workload. Two real-world parallel workload logs have been utilized in this paper which are taken from parallel workload archives (Feitelson 2005). This paper aims to identify the suitable objective function for the proposed hybrid adaptive scheduling approach by investigating the approach on different objective functions, on different workload traces and observing the values of resultant metrics. Hence, this paper aims to address and investigate the following research questions:

1. Which formulation of objective function in the hybrid scheduling approach provides the better optimization potential and better results over all metrics?

The structure of the paper is as follows; the first section provides the general introduction, motivation and focus of the paper by addressing the relevant research questions. The second section describes some of the related work which discusses the research articles addressing certain scheduling heuristics, meta-heuristic scheduling techniques, hybrid scheduling approaches and performance comparison among them with respect to particular scheduling-oriented metrics, The third paper describes the overview of the hybrid scheduling approach on a high level with focus on the performance metrics considered and different objective functions formulated. The fourth section represents the experimental setup and related parameters considered for the simulation, discusses the experimental results and provide analysis of the results. The fifth section describes the conclusion and prospects of the research work.

2 Related work

Since the last two decades, the scientific community produced numerous contributions in the context of task scheduling in the cloud computing environment. The related work has been conducted and studied in two phases. The first phase discusses the research articles in which certain generic baseline scheduling heuristics are evaluated and compared among one another on the basis of certain performance metrics. The second phase discusses the research articles in which certain metaheuristic scheduling approaches and hybrid scheduling approaches are proposed. The literature research helped us to identify certain scheduling-oriented performance metrics and real-world parallel workload logs which resembles large-scale problem instances.

2.1 Heuristic techniques

Heuristics are problem-solving or decision-making techniques for finding an approximate solution within a given time period. Heuristic methods do not guarantee a correct and optimal solutions. However, solutions obtained through heuristic methods are satisfactory and reasonable (Todd 2001). This section mainly describes the comparative performance analysis of different scheduling heuristics with respect to certain performance metrics in the context of cloud computing.

Sharma et al. (2017) conducted a relative performance analysis of Min-Min and Max-Min with respect to makespan and it was concluded from the experimental results that the Max-Min outperformed the Min-Min when the large-sized tasks are greater in number and vice-versa. Jemina P. and Lawrence (2014) performed a comparative study between Max-Min and Min-Min heuristic on the basis of metric makespan and observed that Max-Min heuristic achieves better makespan in comparison to the Min-Min heuristic. Madni et al. (2017) compared the performance of six heuristics namely First Come First Serve, Minimum Completion Time, Minimum Execution Time, Max-Min, Min-Min and Suffrage in terms of cost, degree of imbalance, makespan and throughput. The authors conducted the evaluation in both homogeneous and heterogeneous environments using large-scale parallel workload traces namely HPC2N and NASA Ames iPSC/860. The authors concluded that, on an overall perspective, Min-Min heuristic performed better than other heuristics, while Max-Min and Suffrage heuristic also gave good results, and the performance of heuristics depends on the nature of workload traces. Sindhu and Mukherjee (2011) proposed two scheduling heuristics namely Longest Cloudlet Fastest Processing Element (LCFP) and Shortest Cloudlet Fastest Processing Element (SCFP). In LCFP, the larger tasks are mapped to resources with high computational capacity whereas in SCFP, the shorter tasks are assigned to resources with high computational capacity. Experiments were conducted with respect to makespan by varying the number of cloudlets, length of cloudlets and number of VMs. Experimental results indicated that these heuristics exhibited similar performances at a smaller number of tasks and LCFP exhibited better performance relatively at larger number of tasks. Aladwani (2020) performed a comparative analysis of prominent scheduling heuristics namely First Come First Serve (FCFS), Shortest Job First (SJF), Max-Min in terms of performance metrics namely total waiting time and total finish time (makespan). The authors concludes that among the heuristics, SJF performs the best in terms of total waiting time and makespan. Streit (2002) presented a simple job scheduler and an advanced dynP job scheduler with a dynamic switching policy to switch the scheduling policy during the run time by utilizing the following scheduling policies namely FCFS, SJF and Longest Job First (LJF). The authors concluded that the advanced dynamic job scheduler outperformed the simple one on the basis of performance metrics such as utilization and average response time. Bandaranayake et al. (2020) presented a new scheduling strategy called Total Resource Execution Time Aware Algorithm (TRETA) with the objective of minimizing makespan. The authors suggests that the total execution time of computing resource is a crucial factor for arriving at an optimal schedule. The authors compare the proposed strategy against other baseline heuristics namely Min-Min, Max-Min, FCFS and Minimum Completion Time in terms of the metrics makespan, degree of imbalance and throughput using the large-scale real-world NASA workload logs. The authors concluded that the proposed approach shows significant level of improvement relatively based on the aforementioned metrics.

2.2 Metaheuristic and Hybrid techniques

A metaheuristic is a problem-independent, high-level approach which is used to solve a wide range of problems (Ezugwu et al., 2021). They guide the whole search process, thus facilitating the systematic and efficient identification of global optimum solutions. It is basically an iterative approach that guides a subordinate heuristic, thus exploring and exploiting search space with the aim of finding optimal solutions (Christiansen and Fagerholt, 2009). This section discusses the research articles which proposes metaheuristic-based scheduling approaches, hybrid scheduling approaches and their performance comparison among them on the basis of certain performance metrics. This section helps us to identify the state-of-the-art hybrid approaches which leveraged metaheuristics with generic scheduling heuristics for optimal task scheduling in the cloud environment.

Kaur and Verma (2012) proposed a modified genetic algorithm (GA) which combines the following heuristics namely SCFP and LCFP. In contrast to the standard GA where the initial population is generated randomly, the modified GA generates the initial population with LCFP and SCFP heuristics. Experiment results depicted that the modified GA exhibited better performance under heavy loads in terms of average makespan and execution cost when compared with the standard GA. Verma and Kumar (2012) proposed an improved version of GA which combines the standard GA with Min-Min and Max-Min heuristic. In this improved version, the initial population is generated with Min-Min and Max-Min heuristics hoping to produce better solutions which in turn provides better future generations on subsequent cross over and mutation. The experimental results depicted that the makespan of improved genetic algorithm is less than that of the standard GA. Abdi et al (2014) proposed a modified Particle Swarm Optimization (PSO) in which standard PSO algorithm is combined with SCFP heuristic. The initial population is generated randomly in standard PSO whereas SCFP heuristic is used to generate the initial population in the hybrid approach. It was observed that the proposed approach minimized the makespan when compared with standard GA and PSO. Singh and Kalra (2015) proposed a modified genetic algorithm in which Enhanced Max-Min heuristic has been used for generating the initial population of the genetic algorithm. It was observed that this proposed modified GA outperforms the other modified GA where the initial population was generated with LCFP and SCFP heuristics as discussed in (Kaur & Verma, 2012) and the other modified GA where the initial population has been generated with Min-Min and Max-Min heuristic methods as discussed in (Verma & Kumar, 2012). Nahhas et al. (2021) presented a hybrid approach that uses both heuristics and metaheuristics approach with the aim of optimizing the VM allocation problem considering resource utilization. These authors suggest that a hybrid load management strategy using heuristics and genetic algorithm-based optimization approaches consumes less power in data centers when compared to a specific or generic approach. The authors considered various VM allocation policies for the hybrid approach which were encoded as integers in the chromosomes of the genetic algorithm-based optimization model such that each chromosome represents a specific combination of VM allocation policies. The paper concludes that the proposed hybrid approach results in a significant reduction in energy consumption, appreciable improvement in VM migrations, and a slight increment in SLA violations as opposed to individual VM allocation policies.

However, this paper adopts and translates the hybrid design discussed in (Nahhas et al 2021) as an adaptive hybrid task scheduling approach. This hybrid approach utilizes certain baseline scheduling heuristics identified from the literature research. The research paper (Bandaranayake et al 2020) is identified as the reference paper for comparison. Thus, the cloud infrastructure setup, performance metrics, heuristics and large-scale workload trace discussed in this paper is used for primary evaluation of the approach. The main focus of the paper is to identify the suitable objective function for the genetic algorithm-based hybrid approach by investigating the approach on different objective functions. This is done by evaluating the approach at each particular different objective function on the same workload traces and observing the values of the same metrics.

3 Hybrid Task Scheduling approach

This section presents and describes the proposed hybrid task scheduling approach to optimize the task scheduling problem in the cloud computing environment. The hybrid design discussed in the research paper (Nahhas et al 2021) where it is applied on the VM placement problem is translated into the context of task scheduling as a hybrid task scheduling approach.

3.1 Genetic algorithm-based optimization model

In the research paper (Nahhas et al 2020), the proposed approach utilizes a genetic algorithm-based optimization model in which VM allocation policies are fed as inputs to the optimization model. The hybrid task scheduling approach proposed in this paper also utilizes the same genetic algorithm-based optimization model where on the contrary, scheduling heuristics are fed as inputs to the optimization model. Thus, this approach uses the genetic algorithm-based optimization model to arrive at a particular sequence of scheduling heuristics after n generations and the scheduling heuristics are executed based on this sequence. The optimization is done through genetic algorithm (GA) with the objective of identifying better combination of scheduling heuristics as generations pass by for efficient task scheduling. The basic unit of genetic algorithm is termed as a chromosome or a solution candidate. In our genetic algorithm-based optimization model, a chromosome or a solution candidate is regarded as a combination of scheduling heuristics and follows integer encoding such that each integer represents a particular scheduling heuristic. Eight baseline generic scheduling heuristics identified from the literature research namely FCFS, Random, SCFP, LCFP, Min-Min, Max-Min, SJF and LJF are selected for the genetic algorithm-based optimization model. These scheduling heuristics represented as integers ranging from 0 to 7, are used for chromosome encoding. The length of the chromosome is defined as 24 in our approach which implies that a chromosome is combination of 24 scheduling heuristics in which each scheduling heuristic is an integer between 0 and 7. Each scheduling heuristic is switched after one by one based on a switching interval. The switching interval is based on a time slice which is predefined and set as 25s. In this approach, an initial population of such chromosomes or solution candidates are generated randomly and the population size is also predefined and set as 10. These solution candidates undergo evolutions over many generations to arrive at better solution candidates. The generation size is also predefined and set as 50. In this paper, this genetic algorithm-based hybrid task scheduling approach is investigated by considering different objective functions and is evaluated on different large-scale workload traces on the basis of same performance metrics. This is done to identify the objective function for the approach which provides the better optimization over all the metrics. This will be discussed in the following sections.

3.2 Performance metrics & Objective functions

The hybrid task scheduling approach is investigated and evaluated based on the following metrics namely makespan, throughput, average flow time and average waiting time. These metrics are selected as a result of literature research. The definition of these metrics is described as follows.

Makespan (in seconds) is defined as the time taken to complete the last task in the set of tasks (finish time of the last task) (Bandaranayake et al 2020). It is the total time taken to finish the execution of a set of jobs.

$$Makespan \ = Finish\ Time_i \ \ i \ = last\ task$$

The throughput defines the number of tasks executed per unit time interval (Bandaranayake et al 2020).

$$Throughput \ = \frac{Number\ of\ Tasks}{Makespan}$$

Total waiting time (in seconds) is the sum of waiting times of a set of tasks (Aladwani 2020).

$$Total\ waiting\ time = \sum_{n=1}^{total\ tasks} Waiting\ Time$$

$$Average\ waiting\ time = \frac{Total\ waiting\ time}{Number\ of\ tasks}$$

Flow time (in seconds) is defined as the sum of completion time of a set of tasks (Singh and Kalra 2015; Sindhu and Mukherjee 2011).

$$flow\ time = \sum_{n=1}^{total\ tasks} Completion\ Time$$

$$Average\ flow\ time = \frac{flow\ time}{Number\ of\ tasks}$$

The metrics namely makespan, throughput, average flow time, and average waiting time are used for the evaluation of the hybrid task scheduling approach. The objective of the approach is to reduce makespan, average flow time, average waiting time and increase throughput. Different objective functions namely single-objective and multi-objective functions are formulated based on these metrics. Hence, the genetic algorithm-based optimization model of the approach is investigated by considering the single-objective and multi-objective functions to identify the suitable objective function for the approach. The considered single-objective and multi-objective functions are as follows. Single objective functions include objective function based on makespan alone, objective function based on flow time alone, and objective function based on total waiting time alone. Multi-objective functions include objective function based on makespan and flow time, objective function based on total waiting time, and flow time, objective function based on makespan, flow time, and total waiting time.

4 Experimentation, Results and Analysis

This section discusses the experimental setup and provides a comparative analysis of the simulation results from the experiments. The proposed hybrid task scheduling approach is modeled and simulated using the CloudSimPlus framework. The individual baseline scheduling heuristics and the different hybrid GA approaches on different objective functions are subject to run on the experimental setup to investigate their comparative performance. The simulation results from these experiments enable us to answer the addressed research questions.

4.1 Experimental Setup

The cloud infrastructure settings used for the experiments are referenced from the research paper (Bandaranayake et al 2020). Our cloud infrastructure as shown in table 1, consists of a data center with 20 hosts and 20 VMs are placed across these 20 hosts such that each host will provision one VM. The host and VM processing capacity are to be the same. Two large-scale workload traces are used for evaluation namely NASA iPSC and KTH SP2 which are taken from parallel workload archives (Feitelson 2005). Experiments are conducted on servers with Intel Core i5 3570 @ 3.40GHz CPU, 16 GB RAM, and storage capacity of 512 GB HDD and 128 GB SSD.

Cloud Infrastructures	Value
Datacenter	1
Hosts	20
VmScheduler	Space Shared
VMs	20
CloudletScheduler	Space Shared
VM PES	128
VM RAM	1024
VM Bandwidth	1000
VM Storage	100000
VM MIPS	1000 – 400
Cloudlets	NASA & KTH workload

Table 1. Cloud Infrastructure

4.2 Comparative analysis of the results

This section presents the comparative analysis of the results obtained from the evaluation of the different hybrid GA scheduling approaches with different optimization criteria as shown in table 2 on large-scale problem instances namely NASA workload and KTH workload trace.

Hybrid GA approaches	Objective functions
Hybrid-GA-m	Makespan
Hybrid-GA-w	Total waiting time
Hybrid-GA-f	Flow time
Hybrid-GA-m-f	Makespan & flow time
Hybrid-GA-w-f	Total waiting time & flow time
Hybrid-GA-m-w-f	Makespan, Total waiting time & Flow time

Table 2. Hybrid GA approaches with different optimization criteria

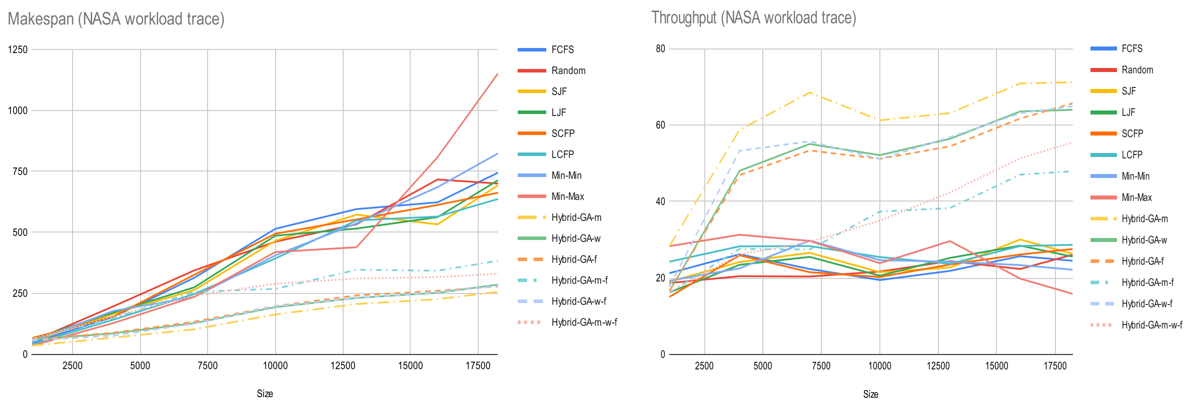


Figure 1: Comparing makespan and throughput among different hybrid GA approaches and baseline heuristics on NASA workload trace.

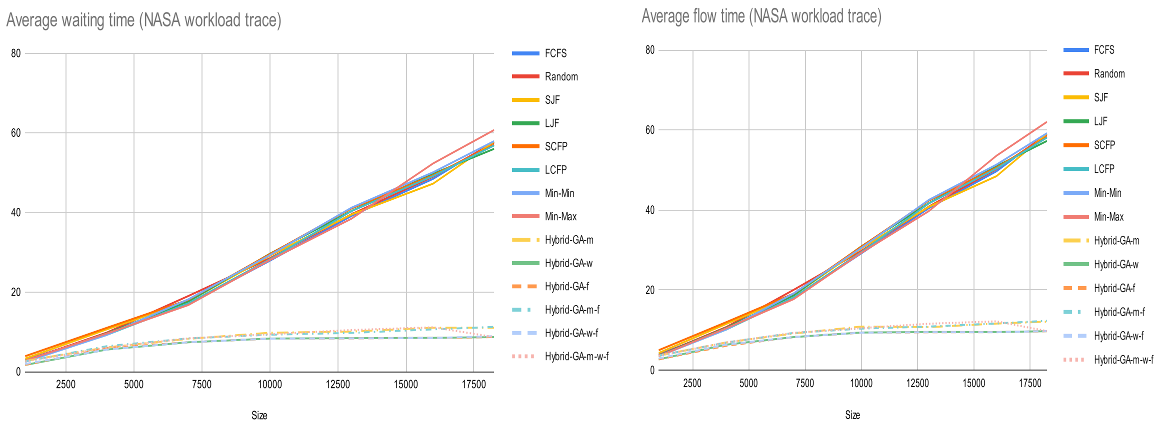


Figure 2: Comparing average waiting time and average flow time among different hybrid GA and baseline heuristics approaches on NASA workload trace.

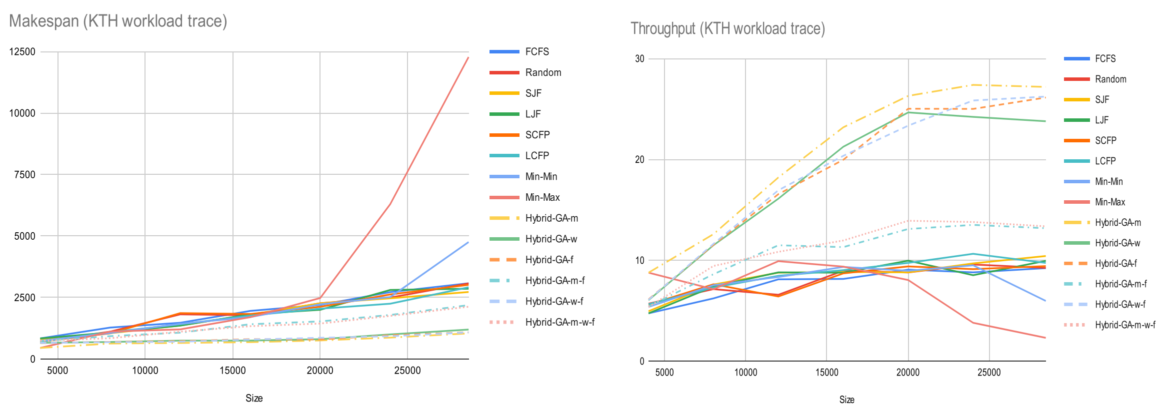
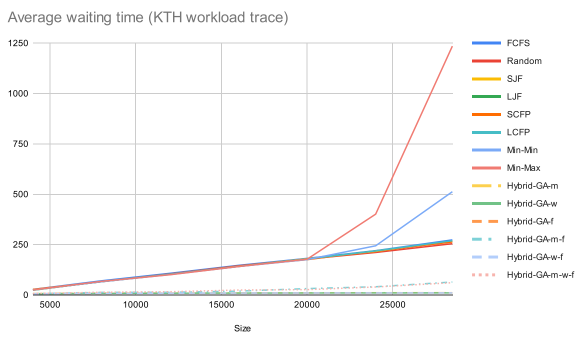


Figure 3: Comparing makespan and throughput among different hybrid GA approaches and baseline heuristics on KTH workload trace.



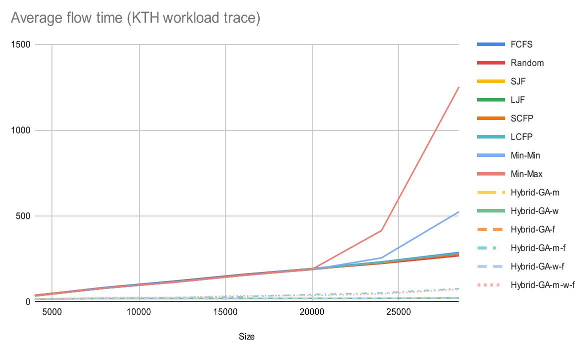


Figure 4: Comparing average waiting time and average flow time among different hybrid GA approaches and baseline heuristics on KTH workload trace.

Figures 1, 2, 3, and 4 represent the comparison of makespan, throughput, average waiting time, and average flow time among hybrid GA approaches and baseline scheduling heuristics on two large-scale real-world workload traces. Figures 1, 2, 3, and 4 clearly depicts that all the hybrid GA approaches perform much better when compared to baseline individual scheduling heuristics in terms of makespan, throughput, average waiting time, and average flow time on both large-scale workloads. It is also clear that all the hybrid GA approaches consistently outperforms the baseline heuristics at different workload sizes of both workloads. The identification of the best hybrid GA approach among the different hybrid GA approaches requires closer examination of the details. Hence, the different hybrid GA approaches have been investigated at full workload sizes of both workload which are illustrated in tables 3 and 4.

HybridGA approaches	Makespan	Throughput	Avg waiting time	Avg flow time
Hybrid-GA-m	256.28	71.18	11.04	12.12
Hybrid-GA-w	285.68	63.96	8.68	9.74
Hybrid-GA-f	278.09	65.65	8.68	9.74
Hybrid-GA-m-f	383.31	47.89	11.24	12.32
Hybrid-GA-w-f	281.05	64.91	8.91	9.76
Hybrid-GA-m-w-f	329.99	55.43	8.70	9.76

Table 3. Comparing metrics among hybrid GA approaches at full NASA workload trace

HybridGA approaches	Makespan	Throughput	Avg waiting time	Avg flow time
Hybrid-GA-m	1047.26	27.19	11.45	22.97
Hybrid-GA-w	1202.08	23.79	10.28	21.97
Hybrid-GA-f	1090.24	26.14	10.51	21.69
Hybrid-GA-m-f	2193.83	13.18	64.3	76.69
Hybrid-GA-w-f	1085.98	26.23	10.42	21.83
Hybrid-GA-m-w-f	2130.94	13.37	60.87	73.10

Table 4. Comparing metrics among hybrid GA approaches at full KTH workload trace

Table 3 and 4 represents the metrics (in seconds) obtained for different hybrid GA approaches at full sizes of NASA and KTH workload trace. From these tables, it is able to compare and contrast the different hybrid GA approaches based on single objective functions and multi-objective functions in terms of the metrics makespan, throughput, average flow time and average waiting time. While comparing among the hybrid GA approaches based on single objective functions on both workloads, the following relative findings can be made. Hybrid-GA approach based on makespan produced best values for makespan and throughput while producing worse values for average waiting time and flow time.

Hybrid GA approach based on total waiting time produced the best value for average waiting time while producing satisfactory values for other metrics. Hybrid GA approach based on flow time produced best value for average flow time while producing better values for other metrics. Hence, among the hybrid GA approaches based on single objective functions, hybrid-GA-approach based on flow time produced better values for all the metrics. While comparing the hybrid GA approaches based on multi-objective functions on both workloads, the following comparative findings can be made. Hybrid GA approach based on makespan and flow time produced worse values for all metrics. Hybrid GA approach based on makespan, flow time and total waiting time produced satisfactory results for all metrics. Hybrid GA approach based on flow time and total waiting time produced better results for all the metrics. Therefore, it can be stated that hybrid GA approach based on flow time produce the best optimization potential over all metrics followed by hybrid GA approach based on flow time and total waiting time which answers the addressed research question.

5 Conclusion and Future prospects

This paper presents an adaptive hybrid task scheduling approach that uses genetic algorithm as the optimization model. The optimization potential of the genetic algorithm depends on the optimization criteria or the objective function used. This paper intends to investigate and identify the best optimization criteria or objective function formulation for the approach with the aim of optimizing the metrics namely makespan, throughput, average flow time, and average waiting time. The hybrid GA approach with different objective functions (different combinations of metrics) has been formulated namely hybrid-GA-m, hybrid-GA-w, hybrid-GA-f, and hybrid-GA-m-f. hybrid-GA-f-w, hybrid-GA-m-w-f These different hybrid GA approaches have been evaluated and investigated on two large-scale workload traces namely NASA and KTH workload traces in terms of makespan, throughput, average waiting time and average flow time. The experimental simulation results suggest that the hybrid GA approach based on flow time produced the best optimization on all metrics followed by the hybrid GA approach based on flow time and total waiting time. The experimental results hold the same behavior on both the workloads which further consolidates the inference. Moreover, the experimental results also depict that all the hybrid GA approaches outperform the baseline scheduling heuristics in terms of the stated metrics consistently at different workload sizes of both workloads which justifies the adaptive performance of the proposed approach. The future prospects include the identification and inclusion of other task scheduling-oriented metrics and the formulation of objective functions based on these metrics. Another prospect includes the investigation of other scheduling heuristics to be used in the hybrid approach, investigation of alternative mechanisms to generate the initial population for the genetic algorithm-based optimization model other than the random mechanism. This approach has been currently evaluated using independent tasks with no order of precedence. Hence, this approach can be evaluated using dependent tasks which can be another future prospect of this research work.

6 References

Abdi, S., Motamedi, S., and Sharifian, S. (2014). “Task scheduling using modified PSO algorithm in cloud computing environment,” *International Conference on Machine Learning, Electrical and Mechanical Engineering*, pp. 37-41

Aladwani, T.2020. “Types of Task Scheduling Algorithms in Cloud Computing Environment”.

Bandaranayake, K., Jayasena., K. and Kumaram, B. 2020. “An efficient task scheduling algorithm using total resource execution time aware algorithm in cloud computing,” *International Conference on Smart Cloud (SmartCloud)*, pp. 29-34.

Christiansen, M., and Fagerholt, K. (2009). “Maritime inventory routing problems,” Encyclopedia of optimization, pp. 1947-1955.

Ezugwu, A. E., Shukla, A. K., Nath, R., Akinyelu, A. A., Agushaka, J. O., Chiroma, H., Muhuri, P. K. (2021). “Metaheuristics: A comprehensive overview and classification along with bibliometric analysis,” *Artificial Intelligence Review*, pp. 1–80.

Feitelson, D., 2005. Parallel Workloads Archive, December 8 (https://www.cs.huji.ac.il/labs/parallel/workload, accessed April 15, 2021)

Jain, A., and Upadhyay, A. 2017. “Cloud scheduling using meta heuristic algorithms,” *International Journal of Computer Sciences and Engineering* (1:5), pp. 132-139.

Jemina, P. R., and Lawrence, D. L. A. 2014. "Performance evaluation of Min-Min and Max-Min algorithms for job scheduling in federated cloud," *International Journal of Computer Applications*, pp. 47-54.

Kaur., S., and Verma., A. (2012). "An efficient approach to genetic algorithm for task scheduling in cloud computing environment," *International Journal of Information Technology and Computer Science*

Madni, H., Shafie, A. L., Abdullahi, M., Abdulhamid, S. and Usman, M. 2017. "Performance comparison of heuristic algorithms for task scheduling in IaaS cloud computing environment".

Mathew, T., Sekaran, C., and Jose, J.2014. "Study and analysis of various task scheduling algorithms in the cloud computing environment", pp.658-664.

Mell, P. M., and Grance T. 2011. "The NIST definition of cloud computing".

Nagadevi, S., Satyapriya K., and Malathy D.2013. "A survey on economic cloud schedulers for optimized task scheduling".

Nahhas, A., Cheyyanda, J. T., and Turowski, K.(2021). "An adaptive scheduling framework for the dynamic virtual machines placement to reduce energy consumption in cloud data centers,"*Proceedings of the 54th Hawaii International Conference on System Sciences*, pp. 878

Sharma, N., Tyagi, S. and Altri, S. (2017). "A Comparative analysis of Min-Min and Max-Min algorithms based on the makespan parameter," *International Journal of Advanced Research in Computer Science*, pp. 3

Sindhu, S., and Mukherjee, S. 2011. "Efficient task scheduling algorithms for cloud computing environment," *International Conference on High Performance Architecture and Grid Computing*, pp. 79-83

Singh. S., and Kalra, M. 2015. "Scheduling of independent tasks in cloud computing using modified genetic algorithm," *Proceedings - 2014 6th International Conference on Computational Intelligence and Communication Networks*, pp. 565-569

Soltani, N., Neysiani, B. N., and Barekatain, B.2017. "Heuristic algorithms for task scheduling in cloud computing: A survey," *International Journal of Computer Network and Information Security*,pp. 16-22

Streit, A. 2002. "A self-tuning job scheduler family with dynamic policy switching," *Workshop on Job Scheduling Strategies for Parallel Processing*, pp. 1-23

Todd, P. M., 2001. "Heuristics for Decision and Choice," *International Encyclopedia of the Social & Behavioral Sciences*, pp. 2-4.

Venu, G.2020. "Task scheduling in cloud computing: A survey*," International Journal for Research in Applied Science and Engineering Technology,* pp.2258-2266.

Verma, A., and Kumar, P. (2012). "Independent task scheduling in cloud computing by improved genetic algorithm," *International Journal of Advanced Research in Computer Science and Software Engineering*, pp. 111-114.