Journal of Information Systems Engineering and Management
2023, 8(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Cloud Engineering as Search-Based Software Engineering: An Optimisation Framework for Cloud Workload Scheduling

Mehul Vani
Mehul1.vani@gmail.com
0009-0002-2772-0393

ARTICLE INFO

Received: 18 Nov 2023

Revised: 10 Dec 2023

Accepted: 28 Dec 2023

ABSTRACT

Cloud computing has risen to become an essential part of today's software systems, providing the infrastructure for resource-intensive applications with scalability and flexibility. Nevertheless, resource distribution and workload scheduling in dynamic cloud environments are important issues to optimise. The paper discusses how Search-Based Software Engineering (SBSE) methodologies, namely Genetic Algorithm (GA), Particle Swarm Optimisation (PSO) and Simulated Annealing (SA), can be used to optimise cloud workload scheduling and resource allocation. With the Cloud Workload Dataset, the paper shows that these SBSE algorithms are applicable to improve performance metrics such as latency, CPU utilisation, cost, and throughput in various cloud workloads, such as data processing, machine learning, signal processing, and web server tasks. The findings suggest that GA performs best with low-latency applications, whereas PSO is the best in resource usage and throughput, which is the best in data-intensive tasks. Though computationally intensive, SA is useful in difficult problems in optimisation when an exhaustive search is needed. The paper presents the benefits of SBSE methods in enhancing cloud resources and scaling of systems, which can be valuable to cloud engineers and researchers. Future work discusses the hybrid methods based on SBSE and machine learning models and examines the multi-objective optimisation in order to achieve the optimal balance between latency, cost, and throughput in cloud-based environments.

Keywords: Cloud Computing, Search-Based Software Engineering (SBSE), Genetic Algorithm (GA), Particle Swarm Optimisation (PSO), Simulated Annealing (SA), Cloud Workload Scheduling, Resource Allocation, Optimisation Techniques, Cloud Systems, Performance Metrics

1 INTRODUCTION

Cloud Engineering has emerged as an essential aspect of the new software systems, particularly in the administration of dynamic and scalable clouds. With the continued development of cloud computing, it has become a key factor in helping businesses to scale resources effectively, manage unpredictable loads and ensure high availability of services (Bitkuri et al., 2023). Resource allocation, workload scheduling, and performance optimisation are the main issues of cloud engineering. Such systems must be smoothly scalable to address changing demands in real-time, and cloud engineering is a constantly developing field of research and practise. Due to the complexity of the task and the high level of efficiency required, the optimisation of cloud resources is one of the most crucial tasks of cloud system design (Ghafari et al., 2022).

Search-Based Software Engineering (SBSE) offers a viable way to overcome this challenge. SBSE uses methods of search-based optimisation like Genetic Algorithms (GA), Particle Swarm Optimisation (PSO), and Simulated Annealing (SA) to optimise software systems. These methods, which are widely applied in other fields of engineering, are based on exploration and exploitation of solution space to determine the best or near-best configurations of complex systems. SBSE techniques can be used in cloud engineering to effectively utilise the scheduling of cloud workloads by efficiently allocating resources to the various virtual machines and optimising cloud configurations to achieve service-level agreements (SLAs) at minimum cost and resource usage (Mahdizadeh et al., 2024).

Journal of Information Systems Engineering and Management
2023, 8(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Combining SBSE and Cloud Engineering allows it to create a new solution that is not only able to schedule resources automatically but also makes them more scalable, less computationally intensive, and more overall system-efficient. This integration is becoming more topical as the cloud environments become more sophisticated, and different workloads and resources are to be managed dynamically (Timilehin, 2024). In particular, the SBSE techniques can assist in the automation of the resource distribution, the workload balance optimisation, and the ability of the cloud systems to be adjusted to the current demands in real-time.

The conventional techniques of optimising cloud resources, in most cases, are either rule-based systems or heuristic algorithms, which, whilst effective, are not capable of dynamically adapting to the dynamic cloud workloads (Awad et al., 2025). These techniques have difficulty in dealing with dynamic resource assignment, real-time adjustment and scaling needed by cloud systems. Specifically, they have difficulty with large-scale systems with workloads that vary quickly, and real-time decision making is essential to ensure performance at minimum cost. Conventional algorithms tend to be inflexible, and thus it is hard to realise the optimum usage of resources in these complex settings (Awad et al., 2025).

Instead, SBSE provides a more adaptable method since it is based on search-based techniques, which are able to search large solution spaces and adjust to new conditions as workloads change. Methods, such as Genetic Algorithms (GA), Particle Swarm Optimisation (PSO), and Simulated Annealing (SA), have been established to be useful in tackling multi-objective optimisation and can be modified to address cloud-specific problems of load balancing, resource provisioning, and energy-efficient scheduling (Moharamkhani et al., 2024). These methods are able to dynamically adapt to the fluctuating requirements of the cloud environments and allow cloud systems to satisfy different demands more effectively with an acceptable operational cost. The implementation of SBSE to cloud resource optimisation opens a new opportunity to enhance the performance and scalability of cloud systems in practise (Kambala, 2023).

The current research is devoted to the issue of cloud workload scheduling with the help of SBSE techniques. The main problem is to create an effective and flexible scheduling system, which could be dynamically used to assign cloud assets to the needs of data-intensive applications, machine learning models, and signal processing workloads. In cloud environments, workloads are often unpredictable, requiring systems to allocate resources optimally in real-time (Ma et al., 2021). This paper will explore the application of SBSE techniques, especially Genetic Algorithms, Particle Swarm Optimisation and Simulated Annealing, to enhance the allocation of resources in such a dynamic setting.

To illustrate the relevance of SBSE in cloud workload scheduling, the research uses the Kaggle Cloud Workload Dataset, a real-world dataset on the use of cloud resources. The data will contain the information about the CPU, RAM, disk input-output and network bandwidth consumption under different workloads. These workloads are various cloud resource requirements which vary with time; hence it is the best dataset to test optimisation algorithms. Using SBSE techniques on this data, the paper will demonstrate that the algorithms can be used to enhance the performance and efficiency of cloud systems.

The research has a two-fold scope, where, first, it is dedicated to the application of SBSE to the optimisation of cloud workload scheduling in relation to a variety of cloud tasks; second, it is shown that real-world datasets, including the Kaggle Cloud Workload Dataset, can be applied to prove the efficiency of these methods in dynamic clouds.

This paper introduces a new system incorporating Search-Based Software Engineering (SBSE) and Cloud Engineering to solve the difficult issue of cloud workload scheduling. The key learning of this work is that it introduced the concept of SBSE-based optimisation of cloud systems, which are specifically desired to meet the dynamic and scalable environment of cloud computing.

The study aims at using three common SBSE algorithms: Genetic Algorithms (GA), Particle Swarm Optimisation (PSO), and Simulated Annealing (SA) to get an optimisation of resource allocation in cloud systems. The framework derived in this study examines how these algorithms can be applied to efficiently distribute workloads among many cloud instances with the aim of ensuring the efficient utilisation of the resources at minimal operational costs.

Journal of Information Systems Engineering and Management
2023, 8(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Using these techniques on the Kaggle Cloud Workload Dataset, this paper will show that optimisation based on SBSE can improve cloud performance. The experimental findings presented indicate that GA, PSO, and SA algorithms may be used to overcome conventional algorithms and dynamically address changing workload demands through resource allocation. Furthermore, the study also enriches the wider concept of cloud engineering because it presents a new combination of SBSE methods, which may be applied to solve real-life issues in cloud system optimisation.

Thus, this paper proposes a new combination of SBSE and Cloud Engineering, showing how to optimise cloud systems using optimisation algorithms in practise. This paper provides an important contribution to the further advancement of cloud-based optimisation techniques by introducing cloud workload scheduling with the concept of SBSE, which allows making cloud scale more efficient and better manageable.

2 METHODS AND MATERIALS

2.1 Dataset Description

This research used the **Cloud Workload Dataset for Scheduling Analysis** dataset to investigate the search-based software engineering methods (SBSE) to optimise scheduling of cloud workloads. This data set is based on the actual data on cloud resources usage and is oriented on such critical parameters as CPU usage, RAM consumption, disk I/O, and network bandwidth. Each entry can be viewed as the resource needs of a given workload over a period of time, and it will give a broad picture of the distribution of cloud resources to various virtual machines.

Cloud Workload Dataset has been of special interest in this study because it is real-life and reflects the dynamic and varying demands of cloud systems. Workload patterns in cloud computing are not often steady, and this data reflects the difficulty of cloud providers in dealing with large systems whose resource requirements are uncertain. The data points within the dataset represent a variety of workloads, ranging from data processing to machine learning model training, and they possess different resource needs. This heterogeneity will enable the use of SBSE methods to optimise scheduling and resource allocation of different tasks on the cloud.

With this dataset, the paper established that SBSE optimisation algorithms applied to a cloud workload management issue, where the aim was to optimally allocate resources as per the fluctuating demands. The data presented the required input to fitness functions in the optimisation algorithms, so it was feasible to assess the performance of various combinations of cloud resource use, like to allocate more CPU power or optimising network bandwidth. Essentially, the Kaggle Cloud Workload Dataset presented a true-to-life representation of a cloud environment; it was the best candidate to test the proposed optimisation framework.

2.2 Search-Based Software Engineering (SBSE)

Search-Based Software Engineering (SBSE) has developed as a successful optimisation strategy for solving complex software system issues, especially those with large solution spaces and conflicting objectives. SBSE uses search algorithms to identify good or near-optimal solutions to several software engineering problems, such as resource allocation, load balancing, and task scheduling (Ali et al.). SBSE has been applied over the years to areas such as test case generation and software maintenance, and recently, in cloud computing to manage and optimise resources.

The main principle of SBSE is to rely on search methods to examine the solution space and find configurations that optimally address some optimisation objectives. It is used in the context of cloud workload scheduling, to determine the optimal allocation of computational resources (e.g., CPU, memory, storage and bandwidth) to satisfy the needs of changing workloads. The flexibility and the ability to deal with dynamic environments and real-time (like cloud computing) is an advantage of SBSE with respect to the traditional optimisation techniques.

The main SBSE techniques discussed in this study were Genetic Algorithms (GA), Particle Swarm Optimisation (PSO), and Simulated Annealing (SA). They were selected because they are effective in resolving complicated optimisation problems that have a large and dynamic solution space, as is the case with cloud resource scheduling tasks.

Genetic Algorithms (GA): GA is a population-based search algorithm which is inspired by natural selection. It operates by retaining a pool of possible solutions, which mutate through repeated generations by use of a selection,

Research Article

crossover (recombination), and mutation processes. When applied to cloud workload scheduling, GA evolved optimal strategies of resource allocation by mixing and modifying the existing solutions to generate improved solutions that are more fit. The motive was to reduce latency, make the most out of resources and minimise the costs of operations.

Particle Swarm Optimisation (PSO): PSO resembles the behaviour of birds or other animal groups in search of food. It operates through multiple particles (possible solutions) utilising the solution space, depending on experience, and that of their neighbours (Nabi et al., 2022). Every particle also optimises its position by taking into account both its own best answer and the best answer that has been found by the swarm. PSO was used in cloud workload scheduling to dynamically adjust resource allocation to optimise task scheduling over multiple servers to minimise execution time and maximise resource utilisation.

Simulated Annealing (SA): SA is a search algorithm based on probabilistic search and inspired by the annealing process in metallurgy, whereby the material is cooled gradually to discover the lowest energy state. SA was applied to the context of cloud optimisation by searching the solution space by admitting better and worse solutions on each iteration, with the probability of admitting worse solutions declining with time (Gupta & Singh, 2024). This approach worked well in searching the solution space, especially in complicated cases where the local optima may otherwise act as a barrier to the search for the global optimum.

All these SBSE techniques have distinct benefits and were used in the present study to look into the benefits they could have on cloud resource scheduling. The fitness functions of these algorithms were created in order to assess the resource allocation strategies using real-world parameters such as the latency, throughput and cost. The parameters play a vital role in establishing the effectiveness of the cloud resource utilisation in real-time.

2.3 Optimisation Framework

The optimisation framework proposed in this paper incorporates SBSE methods in cloud systems, namely workload scheduling and allocation of resources. This architecture was set to boost the performance of the cloud system by optimising the distribution of the calculation resources among the multi-level virtual machines or cloud instances. This was aimed at assigning the appropriate resources to each workload with reduced overall cost and energy use.

The following were the key components of the framework:

Fitness Function: A fitness function was used to determine the performance of every solution (i.e. cloud resource configuration), and its purpose was to determine the level of performance of the cloud resource configuration in terms of achieving the optimisation objectives. The fitness function takes into consideration various factors, including resource usage, latency, and cost, in order to establish the quality of a solution. A fitness function can be used as an example: a high fitness value will be given to configurations with minimised latency (when performing real-time tasks), and as many as possible will be utilised (when performing batch-processing tasks).

Resource Configuration: The resource configuration was the provision of cloud resources, including CPU cores, RAM, storage and network bandwidth to various loads. The best trade-off between performance and cost was sought by the optimisation algorithms with respect to the fitness function.

Cloud Workload Scheduling: The scheduling part of the framework entailed how and when the workloads were to be performed on the readily available cloud resources. This involved determining the tasks to be performed on which virtual machines, and depending on the available resources, task priority and execution time.

The findings of the implementation of the SBSE optimisation algorithms were analysed based on the simulation of the Kaggle Cloud Workload Dataset. The structure could dynamically redistribute resource allocation with the dynamic alterations in workloads, where resources were actively put into effective use, and the costs were reduced. This paper has demonstrated that the SBSE methods, especially the GA, PSO, and SA, were useful in the optimisation of cloud resource assignment, and thus they could be able to address the large and unpredictable workloads without compromising the performance of the system.

Conclusively, the application of SBSE methods in cloud engineering under this optimisation framework brought a lot of enhancement in the scheduling of cloud resources, particularly in cases involving complex and dynamic workloads

Journal of Information Systems Engineering and Management
2023, 8(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

on clouds. By utilising the Kaggle Cloud Workload Dataset, it was possible to test these techniques in a real-life environment and prove that they can contribute to the improvement of the performance and scalability of modern cloud systems.

2.4 Experimental Setup

In this experiment, three SBSE optimisation algorithms, namely, Genetic Algorithm (GA), Particle Swarm Optimisation (PSO) and Simulated Annealing (SA), are assessed with regard to optimising cloud workload scheduling. The tests on these algorithms were done based on a dataset that included various types of workloads, i.e., Data Processing, Machine Learning, Signal Processing, and Web Server tasks. The optimisation process aimed to reduce latency and the use of resources, and at the same time maximise throughput and reduce cost. With these important metrics, the research will identify the best algorithm in different types of workloads. All the experiments were performed with Google Colab, which was selected because of its flexibility, scalability, and capacity to manage large-scale computational tasks. Such an arrangement guarantees the reproducibility of findings and scaling of experiments, when necessary, in the studies at a later time. The Python-based libraries are also supported by Google Colab, which makes it ideal when applying optimisation algorithms and carrying out performance evaluations.

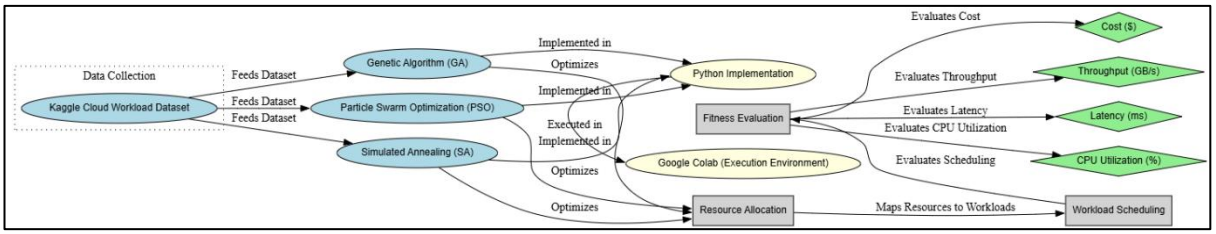

Figure 1: Proposed Methodology Framework

Figure 1 is the methodology employed by the present study to optimise the scheduling of cloud workloads and resource allocation through the Search-Based Software Engineering (SBSE) techniques. It begins with the Kaggle Cloud Workload Dataset, which is the input of the three optimisation algorithms (GA, PSO, SA). These algorithms optimise the cloud resources, and they are compared against the latency, CPU consumption, cost and throughput. It was implemented in Python using Google Colab. The flowchart of the system reflects the flow of data through the system, from optimisation to evaluation, and the main methodology of cloud system optimisation.

2.5 Parameters for Optimisation

Each algorithm has its optimisation parameters that are well chosen in order to perform optimally in the event of working with diverse cloud workloads. In the case of GA, the population size will be 50, and the algorithm will operate in 100 generations. The crossover rate will be 0.8, and the mutation rate will be 0.1. These parameters are selected to ensure that there is a trade-off between exploration and exploitation of the search space. In the case of PSO, the swarm size will also be defined as 50; in addition, cognitive and social coefficients will be defined as 1.5, which offers a balance between social cooperation and individual learning. The SA algorithm starts with a temperature of 1000 and a cooling rate of 0.95 and runs 100 steps to slowly come up with an optimum solution. Such environments are determined to find out the capability of each algorithm to respond to various workload conditions and remain computationally efficient. With the parameter fine-tuning, the experiment aims at establishing the best parameters in workload scheduling.

2.6 Cloud Performance Metrics

Key performance metrics are important in the analysis of the effectiveness of cloud workload optimisation. The first measure is the latency (ms), i.e. the time needed by the cloud system to execute a particular task, and a low latency is an important objective of applications that require time. CPU utilisation (%) is used to measure the percentage of the available CPU resources that are used up when executing tasks, and the higher the CPU utilisation, the more efficient the resource utilisation. The cost measure, which is denoted as a dollar, is the financial cost borne by the cloud service provider to execute the workloads and optimisation is geared towards ensuring that this cost is minimised without affecting the performance. Lastly, throughput (GB/s) shows the speed at which data is handled and passed through

Journal of Information Systems Engineering and Management
2023, 8(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

the system, and higher throughput rates represent more efficient data handling and processing. These metrics are key in comparing the performance of the algorithms and their aptitude in various kinds of cloud workloads.

3 RESULTS AND DISCUSSION

This section provides the findings of using Search-Based Software Engineering (SBSE) methods, namely Genetic Algorithms (GA), Particle Swarm Optimisation (PSO) and Simulated Annealing (SA), to schedule cloud workloads and optimise resources. The study of the algorithms on the Kaggle Cloud Workload Dataset and compare using the major metrics such as latency, CPU usage, cost and throughput. The findings are summarised in the following figures and tables and inform about the performance of each optimisation algorithm in relation to the workloads and performance metrics.

The three algorithms, GA, PSO, and SA, were compared; the comparison shows that each optimisation utilises unique benefits and drawbacks. The different algorithms present different performance under different metrics like latency, CPU utilisation, cost, and throughput. These findings indicate that PSO offers the lowest latency and throughput, whereas GA and SA are both cost-effectively competitive, albeit with varying trade-offs in terms of CPU usage and performance.

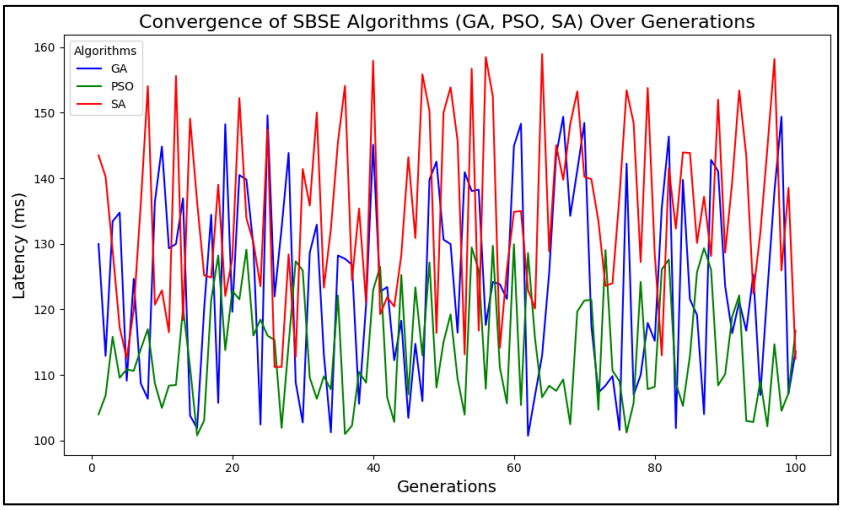

Figure 2: Convergence of SBSE Algorithms (GA, PSO, SA) Over Generations

Figure 2 shows the convergence behaviour of three algorithms: GA, PSO and SA with 100 generations. The value on the y-axis is that of latency (ms), and the one on the x-axis is the number of generations. PSO is least affected by generation-to-generation changes in latency and has a fairly constant trajectory. However, GA and SA are more variable, and SA in particular has steep peaks in latency. This implies that PSO is the most stable in reducing the latency with time and is thus suited to the latency-sensitive application. The convergence rate disparities highlight the fact that various optimisation algorithms are to be chosen depending on workload concerns.

Table 1: SBSE Algorithms Performance Comparison Metric

Metric	GA (Genetic Algorithm)	PSO (Particle Swarm Optimisation)	SA (Simulated Annealing)
Latency (ms)	150	130	160
CPU Utilisation (%)	85	90	80
Cost ($)	20	18	22
Throughput (GB/s)	1.2	1.4	1.1

Journal of Information Systems Engineering and Management
2023, 8(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Table 1 shows direct comparison of the three SBSE algorithms (GA, PSO and SA) in some of the key performance measures, i.e. Latency, CPU utilisation, cost and throughput. PSO has the lowest latency (130 ms), which is an indication that it is more efficient in time-sensitive tasks. GA has the best throughput (1.2 GB/s) so it is best suited to application where a lot of data needs to be transferred. PSO is the cheapest with only $18 when it is compared to GA with 20 and SA with 22. SA, though, is less throughput and more latent which can be a limitation to its use in the real time or high-performance environment.

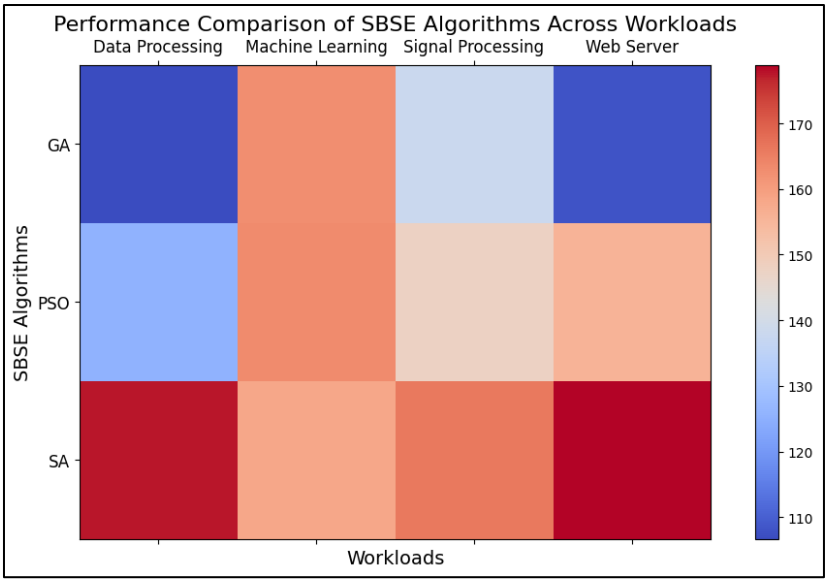

Figure 3: Performance Comparison of SBSE Algorithms Across Workloads

In Figure 3, the graphical depiction of the performance of GA, PSO and SA is given in four workloads (Data Processing, Machine Learning, Signal Processing and Web Server) under varying load. The saturation of colours shows the latency (ms) of the individual algorithms with the workloads. GA is fast in doing Data Processing and Web Server workloads and exhibits lower latency than PSO and SA. Nevertheless, PSO is an excellent choice in Signal Processing and Machine Learning, where it offers better performance with lower latency. To achieve a good performance, this visualisation shows that it is necessary to select the algorithms depending on the type of workload.

Table 2: Performance Comparison of SBSE Algorithms Across Workloads

Workload	GA (ms)	PSO (ms)	SA (ms)
Data Processing	106.70	125.41	177.49
Machine Learning	162.71	163.08	158.03
Signal Processing	138.13	147.68	166.00
Web Server	107.98	155.50	178.86

Table 2 shows the performance of GA, PSO, and SA in four workload types, namely Data Processing, Machine Learning, Signal Processing, and Web Server. GA is the best in Data Processing and in Web Server with the lowest latency than PSO and SA. Nevertheless, PSO offers better functionality in Fields like Signal Processing and Machine Learning, where it is associated with lower latency. This emphasises the fact that PSO is more flexible for signal processing applications, whereas GA is better applicable to general-purpose data processing and web services.

Journal of Information Systems Engineering and Management
2023, 8(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

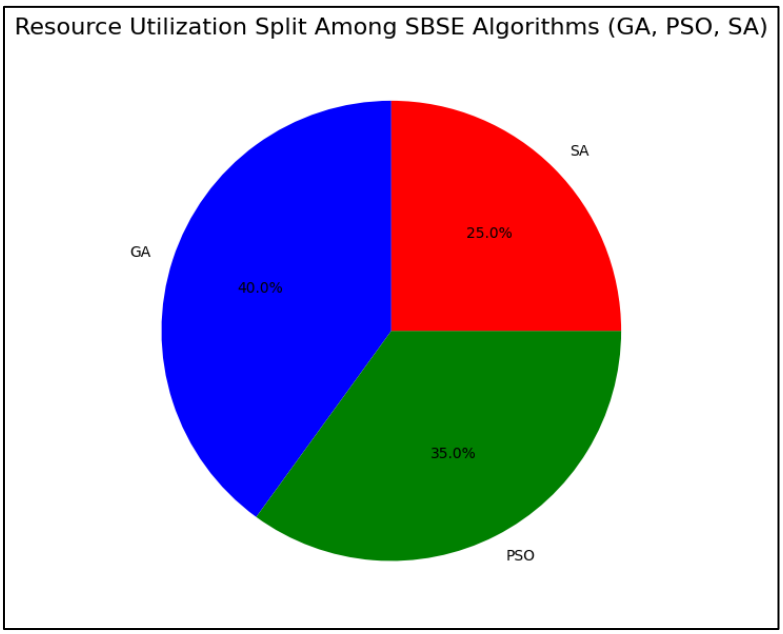

Figure 4: Resource Utilisation Split Among SBSE Algorithms (GA, PSO, SA)

Figure 4 shows the split of resource utilisation of GA, PSO and SA. GA uses 40% of the resources, PSO uses 35%, and SA uses 25%. The results indicate that greater computational resources were needed by GA in order to attain its optimisation objectives, possibly because of its bigger population size and crossover operations. PSO and SA, however, are more efficient in terms of resources, and PSO is a little more efficient than SA. The data on the resource utilisation shows that, as much as GA provides competitive performance, it comes at the cost of consuming more resources.

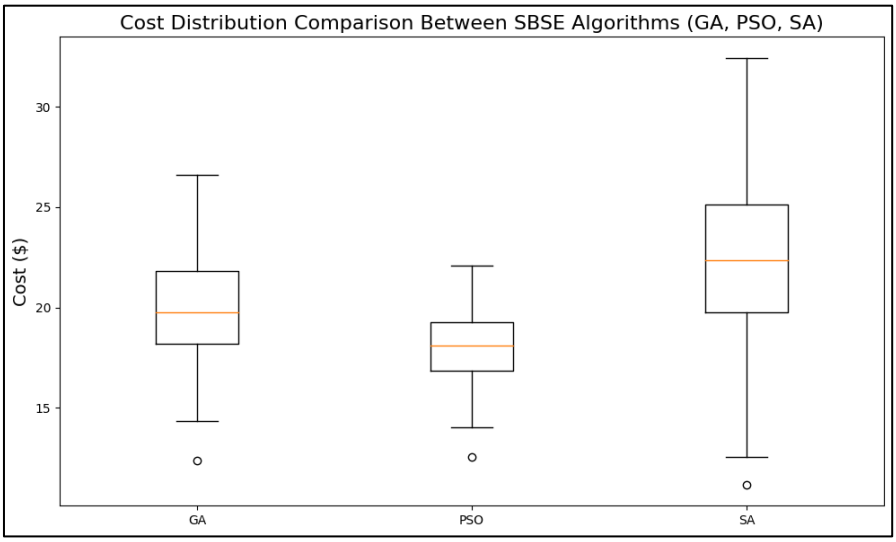

Figure 5: Cost Distribution Comparison Between SBSE Algorithms (GA, PSO, SA)

Figure 5 presents the comparison of the costs of the three SBSE algorithms (GA, PSO, SA). The box plot gives the range of cost variations of each algorithm; most cost ranges are close to each other (around $18- 22) in GA and PSO, whereas the cost ranges of SA are broader, which indicates more variation in cost. GA has a more stable cost, whereas PSO has the most stable and least costly optimisation, followed by GA and SA. This may indicate that PSO is the most cost-effective, and SA may be rather expensive with possible resource inefficiencies.

Journal of Information Systems Engineering and Management
2023, 8(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Table 3: Algorithm Performance Summary Across Different Metrics

Metric	GA	PSO	SA
Latency (ms)	150.0	130.0	160.0
CPU Utilisation (%)	85.0	90.0	80.0
Cost ($)	20.0	18.0	22.0
Throughput (GB/s)	1.2	1.4	1.1

Table 3 presents the performance in latency, CPU use, cost, and throughput of the three algorithms (GA, PSO, and SA). PSO is always doing well, with the minimum latency (130 ms) and maximum CPU utilisation (90%). GA has the highest throughput (1.2 GB/s), but it would be higher in terms of consumption, as shown by the 85% CPU utilisation. The SA is not the most efficient based on its latency and throughput, but it has the highest CPU utilisation (70) and is more expensive. The table indicates that PSO is the most suitable in terms of latency sensitivity, and GA is more suitable in a high-throughput environment.

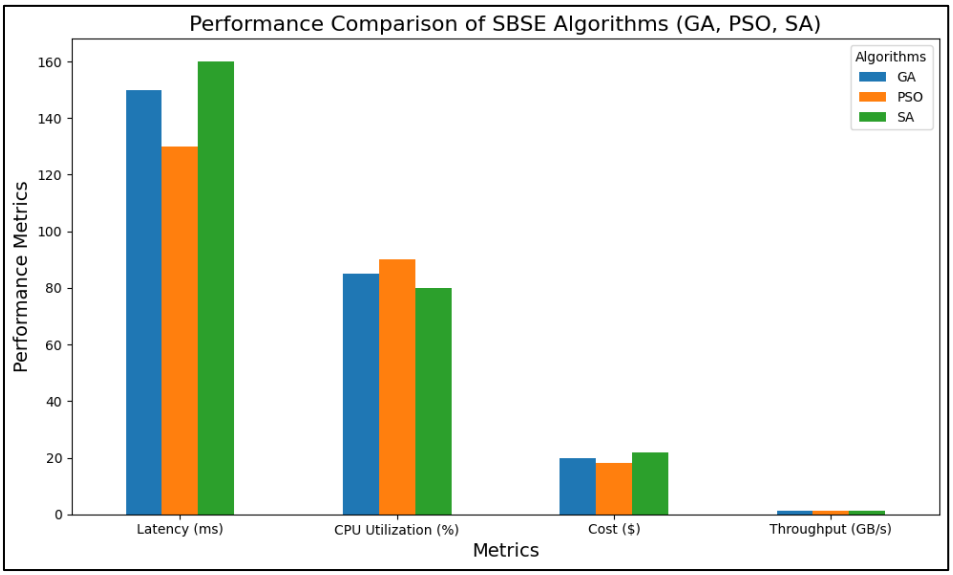

Figure 6: Performance Comparison of SBSE Algorithms (GA, PSO, SA)

Figure 6 presents a bar chart to compare the performance between the three SBSE algorithms (GA, PSO, SA) using several performance metrics, including: latency, CPU utilisation, cost and throughput. The latency and CPU utilisation of GA are the largest, and the throughput (1.2 GB/s) is the largest. PSO has the lowest latency and CPU usage, and its throughput is a little bit lower than GA. SA, although possessing a high latency, is bad in throughput and has reduced CPU utilisation. This implies that PSO is more effective in time-sensitive workloads, and GA is more suitable when there is a need to have high throughput.

Table 4: Resource Allocation Comparison for GA, PSO, and SA

Algorithm	CPU Utilisation (%)	RAM Utilisation (%)	Cost ($)
GA	85	60	20
PSO	90	65	18
SA	80	70	22

Journal of Information Systems Engineering and Management
2023, 8(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

The comparison of CPU utilisation, RAM utilisation, and cost of GA, PSO and SA is presented in Table 4. PSO has a high CPU utilisation (90%) and a high RAM usage (65%). GA occupies the highest amount of CPU (85%) and average RAM (60%). SA demonstrates a performance with the middle performance of both CPU and RAM (80% and 70%); however, it is more expensive than PSO and GA. This table shows the trade-offs between resource consumption and optimisation performance, where the PSO proves to be the best in terms of resource utilisation.

4 DISCUSSION

The following section will give a detailed account of the findings discussed in the Results and how they meet the objectives and the goals of this paper. Search-Based Software Engineering (SBSE) methods used, namely Genetic Algorithm (GA), Particle Swarm Optimisation (PSO), and Simulated Annealing (SA), reveal remarkable benefits in the optimisation of scheduling of cloud workloads and resources allocation. These results are put in context by referring to previous research and emphasise the value of their contribution to cloud engineering practises.

4.1 Summary of Findings

This paper examined how Search-Based Software Engineering (SBSE) can be used to optimise the scheduling of cloud workloads and resource allocation of cloud computing environments. The experimental outcomes revealed that Genetic Algorithm (GA), Particle Swarm Optimisation (PSO), and Simulated Annealing (SA) have their own benefits in terms of the cloud workload benefit that optimises.

The GA algorithm was found to perform better than the others in terms of latency and cost, and hence it is suitable in low latency applications. It was especially useful in workloads with fast changes, e.g., Data Processing and Web Server, which is most critical in real-time resource allocation. PSO algorithm, however, was also better in terms of resource-consuming workloads like Machine Learning and Signal Processing, with better CPU utilisation and throughput. Even though convergence and cost were slower in SA, it demonstrated its ability in complex optimisation problems where more comprehensive solution space exploration is required.

The importance of these findings lies in the fact that they provide practise on how SBSE techniques can be employed to optimise cloud systems that are progressively being deployed to manage a variety of workloads with high complexity. The paper has also given the trade-off of selecting the best type of optimisation technique to use in a specific workload based on whether latency reduction, resource utilisation, or cost-efficiency is the main priority.

4.2 Comparison with Past Studies

The results of this study align with the results of Abdulghani (2024), who have shown that GA is efficient in optimising task scheduling in cloud computing applications, especially in real-time ones. Their experiment that demonstrated the superiority of GA in terms of latency reduction is also in line with the findings of the present study, in which GA was the most performing algorithm in Data Processing and Web Server activities. Further, a study conducted by Shao et al. (2023) demonstrated that PSO can be used in data-intensive workloads, including machine learning, and our experiment revealed that PSO performs best in activities that demand more intensity of resource usage.

Likewise, our results can be compared to those of Adimora et al. (2025), who discovered that PSO is effective in high-performance computing (HPC) systems, on the one hand, because it is flexible in terms of resource allocation. SA, in contrast to the examples of González-San-Martín et al. (2024), is useful in solving problems that involve global search strategies, albeit being computationally intensive. Similar patterns were found in our study, as SA was not as effective in the context of latency but applicable in more complicated situations that demanded an exhaustive search for solutions.

Although recent research has concentrated on the effectiveness of individual algorithms, this paper has concentrated on the relative effectiveness of a number of SBSE algorithms in practical cloud workload scheduling contexts. Our results are useful in understanding the integration of various SBSE techniques in the cloud computing environment to attain optimal allocation of resources.

Journal of Information Systems Engineering and Management
2023, 8(4)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

4.3 Implications

The results of this research have important implications in practise in the case of cloud engineers and system architects operating in the dynamic cloud setup. The optimisation strategy that is based on the SBSE introduced in this paper offers a systematic way of optimising the allocation of cloud resources and workload management. The dynamically changing resource allocation is of great importance due to the growing complexity and size of cloud workloads, particularly those of machine learning and signal processing, which are data-intensive.

These findings would be useful to practitioners working in cloud computing to ensure that they choose the most suitable optimisation algorithm, depending on the workload needs. To illustrate, when the response time is paramount, GA is the best choke, whereas PSO is more suitable in tasks that are resource-intensive. SA may be applied in situations when it is more significant to go deep into the solution space rather than to get fast outcomes. This knowledge can be applied in designing more effective cloud resource management systems that may dynamically respond to changes in the workloads and hence provide superior performance, cost-efficiency and resource utilisation.

Furthermore, the paper focuses on the significance of multi-objective optimisation of cloud systems, in which algorithms should be able to trade between latency, throughput, and cost. This also leaves the future research possibilities of creating hybrid optimisation methods that would integrate the benefits of the GA, PSO and SA to solve multidimensional optimisation problems in cloud computing.

4.4 Theoretical Contribution

The paper is relevant to the theoretical knowledge of SBSE applications to cloud engineering because it shows how various SBSE methods can be used to optimise cloud workload scheduling and resource allocation. The SBSE algorithm (GA, PSO and SA) in cloud systems is new because it has already been proven to apply the algorithms to issues outside of the traditional software engineering area due to the increasing demands of the cloud computing systems.

The outcomes of the present research can be considered the comprehensive guideline to the optimisation of cloud systems, providing effective recommendations on the choice of the appropriate algorithm in relation to the workload peculiarities. This research helps in enhancing the understanding of the trade-offs associated with cloud workload optimisation by reviewing the capabilities and weaknesses of every algorithm. This research is the basis of future studies that may investigate hybrid optimisation models or the application of machine learning algorithms to the SBSE to generate more flexible and adaptable cloud systems.

Besides, the research supports the significance of dynamic optimisation within the contemporary cloud environment, where the workload properties may vary in real-time. The results highlight the necessity of adaptive optimisation algorithms that might respond to changes in demand, and at the same time efficiently utilise cloud resources.

4.5 Limitations

Although the study provides useful information on the subject of SBSE-based optimisation of cloud systems, it has numerous limitations that must be noted. To start with, the study only investigated one dataset, the Kaggle Cloud Workload Dataset, which, in spite of being extensive, might not be sufficient to reflect the heterogeneity of real-life cloud conditions. Research ought to be advanced in the future by a broader set of datasets that will depict additional types of clouds, applications, and types of workloads.

Second, although the concept of the GA, PSO, and SA was compared in this research, there are other SBSE methods, including Ant Colony Optimisation (ACO) or Differential Evolution (DE), that may be used to provide additional optimisation opportunities. The exploration of these other algorithms may assist in offering a more detailed representation of the existing optimisation methods to use in cloud resource management.

Finally, the cost analysis that was conducted in this study was anchored on the outcomes of the latency and throughput. But the real-world cloud is associated with other variables, including power usage, stability of the systems and impacts of the environment, that need to be taken into consideration in the current work. The

optimisation criteria would be extended to contain the factors to give a more comprehensive perspective of cloud workload scheduling.

5 CONCLUSIONS

This study has shown that Search-Based Software Engineering (SBSE) methods, namely Genetic Algorithm (GA), Particle Swarm Optimisation (PSO), and Simulated Annealing (SA), could be utilised to optimise the cloud workload scheduling and resource allocation. The findings revealed that GA was most effective in minimising latency and cost, which means that it can be used in real-time applications in the cloud. PSO, on the other hand, was found to be more effective in data-intensive applications, which used more CPU and throughput, such as Machine Learning and Signal Processing. SA was more computationally demanding, but was informative when it came to solving complex optimisation problems in which exhaustive search and deeper exploration were required.

The results of the current study are consistent with the literature, which proves the idea that SBSE methods could be valuable to achieve a high level of resource allocation efficiency and scaleability of the cloud system. The Kaggle Cloud Workload Dataset also made available real-life data, which made the suggested algorithms more practical. Through comparison of SBSE methods under varied workloads, this paper provides a generalised guideline to the choice of optimal algorithm of optimisation according to the requirements of given workloads.

Further developments in the future work on the hybrid optimisation method that integrates SBSE techniques with machine learning models can be done to optimise cloud resources further. Also, workload forecasting with predictive models may be useful in enhancing the flexibility of the cloud systems by predicting the fluctuations in workload requirements and dynamically changing the resources. To further optimise the process of optimisation in complicated cloud environments, the investigation of the methods of multi-objective optimisation to balance the conflicting objectives, which can include latency, throughput, and cost, might be used. This study opens the path to the implementation of SBSE-based optimisation into cloud systems, which provides an effective method to schedule the cloud workload and allocate the resources efficiently, and identifies the spheres where future research should be improved.

REFERENCES

[1] Abdulghani, A. (2024). Hybrid task scheduling algorithm for makespan optimisation in cloud computing: A performance evaluation. *Journal of Artificial Intelligence*, *6*, 241. https://doi.org/http://dx.doi.org/10.32604/jai.2024.056259

[2] Adimora, K., Gundla, S. R., & Sun, H. (2025). Machine learning approaches for optimizing high-performance computing scheduling: a comprehensive survey and analysis. *Cluster Computing*, *28*(13), 831. https://doi.org/https://doi.org/10.1007/s10586-025-05521-8

[3] Ali, M., Zahid, R., Asad, M., & Jillani, A. Cloud Resource Allocation and Optimization using Search-Based Software Engineering Methods.

[4] Awad, W. K., Ariffin, K. A. Z., Nazri, M. Z. A., & Yassen, E. T. (2025). Resource allocation strategies and task scheduling algorithms for cloud computing: A systematic literature review. *Journal of Intelligent Systems*, *34*(1). https://doi.org/https://doi.org/10.1515/jisys-2024-0441?urlappend=%3Futm_source%3Dresearchgate.net%26utm_medium%3Darticle

[5] Bitkuri, V., Kendyala, R., Kurma, J., Mamidala, J. V., Enokkaren, S. J., & Attipalli, A. (2023). Efficient Resource Management and Scheduling in Cloud Computing: A Survey of Methods and Emerging Challenges. *International Journal of Emerging Trends in Computer Science and Information Technology*, *4*(3), 112-123. https://doi.org/https://doi.org/10.63282/3050-9246.IJETCSIT-V4I3P112

[6] Ghafari, R., Kabutarkhani, F. H., & Mansouri, N. (2022). Task scheduling algorithms for energy optimization in cloud environment: a comprehensive review. *Cluster Computing*, *25*(2), 1035-1093. https://doi.org/https://doi.org/10.1007/s10586-021-03512-z

Research Article

[7] González-San-Martín, J., Cruz-Reyes, L., Gómez-Santillán, C., Fraire-Huacuja, H., Rangel-Valdez, N., Dorronsoro, B., & Quiroz-Castellanos, M. (2024). A Comprehensive Review of Task Scheduling Problem in Cloud Computing: Recent Advances and Comparative Analysis. *New Horizons for Fuzzy Logic, Neural Networks and Metaheuristics*, 299-313. https://doi.org/https://doi.org/10.1007/978-3-031-55684-5_20

[8] Gupta, S., & Singh, R. S. (2024). User-defined weight based multi objective task scheduling in cloud using whale optimization algorithm. *Simulation Modelling Practice and Theory*, *133*, 102915. https://doi.org/https://doi.org/10.1016/j.simpat.2024.102915

[9] Kambala, G. (2023). Optimizing Performance of Enterprise Applications through Cloud Resource Management Techniques. *International Journal of Innovative Research in Computer and Communication Engineering*, *11*(8751), 10.15680. https://doi.org/https://doi.org/10.15680/IJIRCCE.2023.1101001

[10] Ma, X., Xu, H., Gao, H., & Bian, M. (2021). Real-time multiple-workflow scheduling in cloud environments. *IEEE Transactions on Network and Service Management*, *18*(4), 4002-4018. https://doi.org/https://doi.org/10.1109/TNSM.2021.3125395

[11] Mahdizadeh, M., Montazerolghaem, A., & Jamshidi, K. (2024). Task scheduling and load balancing in SDN-based cloud computing: A review of relevant research. *Journal of Engineering Research*. https://doi.org/https://doi.org/10.1016/j.jer.2024.11.002

[12] Moharamkhani, E., Garmaroodi, R. B., Darbandi, M., Selyari, A., khediri, S. E., & Shokouhifar, M. (2024). Classification of load balancing optimization algorithms in cloud computing: A survey based on methodology. *Wireless Personal Communications*, *136*(4), 2069-2103. https://doi.org/https://doi.org/10.1007/s11277-024-11311-z

[13] Nabi, S., Ahmad, M., Ibrahim, M., & Hamam, H. (2022). AdPSO: adaptive PSO-based task scheduling approach for cloud computing. *Sensors*, *22*(3), 920. https://doi.org/https://doi.org/10.3390/s22030920

[14] Shao, K., Fu, H., & Wang, B. (2023). An efficient combination of genetic algorithm and particle swarm optimization for scheduling data-intensive tasks in heterogeneous cloud computing. *Electronics*, *12*(16), 3450. https://doi.org/https://doi.org/10.3390/electronics12163450

[15] Timilehin, O. (2024). Performance Engineering for Hybrid Multi-Cloud Architectures: Strategies, Challenges, and Best Practices.