Bima Journal of Science and Technology, Vol. 4(2) Dec, 2020 ISSN: 2536-6041

A SURVEY AND ANALYSIS OF METAHEURISTIC BASED TASK SCHEDULING ALGORITHMS IN CLOUD COMPUTING ENVIRONMENT

[1]MOHAMMED ABDULLAHI, [2]BARROON ISMAEEL AHMAD, [3]A. F. DONFACK KANA

[1,2,3]Department of Computer Science, Ahmadu Bello University, Zaria – Nigeria

Corresponding Author: abdullahilwafu@abu.edu.ng

ABSTRACT

Cloud service providers wants to schedule as many user applications as possible on each resource to maximize the utilization of resources, while the consumers wish to have their requests served at minimal cost. Since Cloud Computing is an economic setting, cloud service providers and consumers tries to increase their income and return on investment (ROI) by optimal resource. Efficient scheduling of resources becomes central to meeting Service Level Agreements (SLAs) in delivering effective cloud services. When the resource provisioning is overestimated, it leads to under-utilization of resources and loss of revenue. Due to the practical applications and challenges of executing large scale applications, task scheduling of applications on the large scale have become an emerging research in cloud computing and have attracted significant attention of researchers in recent times. Moreover, various heuristics have been applied to solve task scheduling problems which generate optimal solutions for small size problems. However, the quality of solutions produced by these techniques degrades woefully as the problem size and number of variables to be optimized increases. Also, these heuristic methods do not have provisions and support for meeting various Quality of Service (QoS) requirements like response time, makespan time, reliability, availability, energy consumption, cost, resource utilization. In contrast, many cloud users requires certain QoS satisfaction especially for scientific and business domain applications. In recent times, attempts have been made to address task scheduling problems using metaheuristic algorithms to address this problem. Using metaheuristic algorithms for solving task scheduling problems in cloud have shown promising improvements in achieving efficiency, by reducing the solution search space. This paper provides the analysis of metaheuristic algorithms, which we hope to be of great interest to the upcoming researchers in the field of optimizing cloud service resource provisioning.

Keywords: Cloud Computing; Task Scheduling; Metaheuristics; Heuristics; Service Level Agreement.

INTRODUCTION

Cloud Computing provides on-demand access to shared pool of physical or virtual resources that are scalable and elastic which can be rapidly provisioned based on pay-per-usage model. Cloud resources include servers, operating platforms, networks, softwares, and storage. Cloud computing services are hosted in data centers, managed by specialized service providers and accessed by service consumers through client devices. Furthermore, cloud service models can be categorized into: Software-as-a-Service (IaaS), Platform-as a-Service (PaaS), and Infrastructure-as-a-Service (IaaS) (Buyya*etal.,*2009; Zhang et *al.,* 2010). SaaS model enables users to utilize applications like word processing softwares, email, Cloud storage services; however, users do not have control over PaaS and IaaS models. PaaS model provides tools and platform for software development, testing, deployment and related tools, this make SaaS and PaaS unsuitable for hosting large scale applications. Whereas, IaaS provides access to flexible and scalable computing resources for large scale application deployment. The virtualized compute resources called virtual machines (VMs) with pre-configured CPU, storage, memory, and bandwidth are leased to users by paying for what they use only. Various VM instances are available to the users at different prices to serve their various application needs, this give the users the freedom to control compute resource at their disposal.

Furthermore, Clouds are generally categorized as private, community, public, and hybrid, based on their exposure, ownership, and deployment model (Zhang et *al.,* 2010;Srinivasan et *al.,* 2015). Private cloud is used by only one organization and the services are provided on in-house data center, the private cloud services are not accessible to the general public. In contrast, usage of public cloud infrastructure is unrestricted, whereas the community cloud make its services shared among a number of organizations. Hybrid clouds provides services deployed on two or more clouds, and it permits application and data interoperability among the participating clouds (Zhang *et al.,* 2009; Li *et al.,* 2013).

Cloud service providers and cloud service consumers are the two central parties involved in Cloud Computing environment. Providers own high computing resources in their data centers and lease them to consumers on pay-per-use model. Whereas, the consumers lease resources from providers to execute their applications. On one hand, the target of the provider is to maximize return on investment as much as possible. To that effect, providers want to schedule as many user applications as possible on each resource to maximize the utilization of resources. On the other hand, consumers wish to have their requests served at minimal cost. Since Cloud Computing is an economic setting, cloud service providers and consumers increase their income and return on investment (ROI) by optimal resource scheduling (Armbrust et *al.,* 2010; Chen et *al.,* 2015a). Efficient scheduling of resources becomes central to meeting Service Level Agreements (SLAs) in delivering effective cloud services (Morshedlou and Meybodi, 2014). SLA the is terms of the contract between the cloud provider and consumer, it contains QoS requirements of user and penalties for violating the agreed terms. When the provisioning of resources are underestimated, it results to broken SLAs and high payment for penalties. Likewise, when the resource provisioning is overestimated, it leads to under-utilization of resources and loss of revenue (Dikaiakos et *al.,* 2009).

Due to the practical applications and challenges of executing large scale applications, task scheduling of applications on the large scale have become an emerging research in cloud computing and have attracted significant attention of researchers in recent times. Moreover, various heuristics have been applied to solve task scheduling problems which generate optimal solutions for small size problems (Chen et *al.,* 2013;Ming and Li, 2012; Mao et *al.,* 2014; Patel et *al.,* 2015). However, the quality of solutions produced by these techniques degrades woefully as the problem size and number of variables to be optimized increases. Also, these heuristic methods do not have provisions and support for meeting various QoS requirements. In contrast, many cloud users requires certain QoS satisfaction especially for scientific and business domain applications. In recent times, attempts have been made to address task scheduling problems using metaheuristic algorithms to address this problem (Hameed et *al.,* 2014;Wu *etal.,* 2015; Singh and Chana, 2016b). Utilizing metaheuristic algorithms for solving task scheduling problems in cloud have shown promising improvements in achieving efficiency, by reducing the solution search space. However, metaheuristic algorithms incur high computational time and in some cases return local optimum solution especially when dealing with large solution space, also, these techniques may suffer from premature convergence and imbalance between local and global search (Tsai and Rodrigues, 2014;Guzek *et al.,* 2015;Kalra and Singh, 2015; Zhan *et al.,* 2015;Xue*et al.,* 2016;Meena*etal.,* 2016). These limitations result to sub-optimal task schedule solutions which affects the performance of service provision in terms of meeting the desired QoS objectives.

Metaheuristic based task scheduling are classified into optimization techniques and number of objectives. These criteria are the main parts of metaheuristic based task scheduling methods. Metaheuristic approaches are primarily used as the solution search techniques in task scheduling. The most popular metaheuristic algorithms applied to task scheduling problems are Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO) and other metaheuristic algorithms recently applied to task scheduling include League Championship Algorithm (LCA) , Cokoo Search (CS), and Cat Swarm Optimization (CSO) (Tsai and Rodrigues, 2014;Kalraand Singh, 2015; Zhan et *al.,* 2015; Singh and Chana, 2016a). Based on the number of objectives metaheuristic based task scheduling algorithms are categorized into single and multi-objective algorithms. The multi-objective algorithms represent the techniques trying to obtain Pareto optimal trade-off solutions. The algorithms that aggregate the number of objectives into single weighted fitness function are regarded as single objective optimization algorithms in this thesis.

The typical metaheuristic techniques are discussed in Section 1.2.2, where each particular technique is discussed in a subsection. Section 1.4 discussed issues and challenges of metaheuristic based task scheduling algorithms in handling large scale scheduling problems.

Task Scheduling in Cloud Computing Environment

In this section, different types of task scheduling optimization techniques in cloud computing are discussed with some examples of applied techniques for task scheduling optimization and discussion on the kinds of the problems tackled by these techniques. The aim of cloud service provider is to allocate as few resources as possible to service the workloads of the cloud service consumers in order to increase return on investment. On the other hand, cloud service consumers aim to get their workloads executed at minimal cost and high QoS satisfaction. One way to satisfy both requirements of cloud service providers and consumers is to employ optimization techniques. Most of the resource management design decisions found in cloud computing development relate to meeting resource usage or application requirements that target to optimize task scheduling. Based on the cloud stakeholders design decisions, optimization techniques might target challenges in:

i. managing user QoS, from cloud service provider point of view

ii. managing computing resources, from cloud service point of view

iii. managing cloud computing operating environments

Managing user QoS from the cloud service provider point of view involves scheduling decisions to satisfy various user QoS requirements like make-span, execution cost, budget, deadline, and response time. The QoS based scheduling approaches makes scheduling decisions to ensure adherence to QoS requirements and terms of SLA. The heterogeneity and dynamics of users and their QoS requirements which play a crucial in managing user QoS, which affects scheduling decisions (Singh and Chana,2016a). Managing computing resources from the perspective of the service provider focuses on the maximum utilization of resources, minimal energy consumption and low carbon emission (Kaur and Chana, 2015; Zhao et *al.,* 2016). Cloud computing operating environment consists of heterogeneity of computing resources, dynamism of the computing environment introduced by pay-as-you model, and deployment density of applications (Rodriguez and Buyya, 2014).

Quality of Service based Task Scheduling Objectives

The QoS requirements is one the important factor in task scheduling on cloud, the success rate of task execution depends on meeting the required QoS objectives like makespan, cost, reliability, and security subject to certain imposed constraint like deadline and budget (Ranaldo and Zimeo, 2009; Chen et *al.,* 2013;Alkhanak et *al.,*2015). Makespan is the total time to execute the entire user application by putting into consideration the finish time of the last task (Wu et *al.,* 2012;Netjinda et *al.,* 2014a). Deadline is the total time required to execute all the tasks (Abrishami et *al.,* 2012;Xueand Wu, 2012), users usually specify a deadline for the whole application. Budget is the cost bound a user offer to pay a cloud provider for the desired services, budget depends on the selected deadline to offer required QoS at minimum cost (Abrishami et al., 2013;Liu*etal.,* 2011). Reliability is the probability that a task assigned to a computing resource can be completed successfully and effectively (Malawski*etal.,* 2015). Security deals with the

Confidentiality of tasks execution and determine the level of trustworthiness of candidate computing resources (Yu and Buyya, 2005;Deelman et *al.,* 2015). The best effort task scheduling are the common kind of task scheduling techniques (Liu et *al.,* 2010b;TilakandPatil, 2012; Wang et *al.,* 2013). The best effort scheduling only minimizes makespan thereby ignoring other factors like cost of execution and other QoS requirements. On the other hand, QoS aware task scheduling techniques try to maximize system performance under some QoS objectives under certain. As a result, various QoS requirements and constraints needs to be considered when designing efficient task scheduling approaches in cloud computing environment (Varalakshmi et al., 2011;Abrishami et *al.,* 2012).

Metaheuristic Techniques for Task Scheduling

The large scale scheduling problem make the solution search space complex for optimization techniques by introducing several local optima, and the presence of many local optima make it difficult for the scheduling techniques to find near-optimal solutions. The traditional algorithms for solving task scheduling problems on cloud are based on exhaustive search which can efficiently handle small problem sizes (Chen et *al.,* 2013;Ming and Li, 2012; Mao et *al.,* 2014; Patel et *al.,* 2015;Gogos et *al.,* 2016). However, the quality of solutions produced by these techniques degrades woefully as the problem size and number of variables to be optimized increases. Furthermore, these heuristic methods do not have provisions and support for meeting various QoS requirements. In contrast, many cloud users requires certain QoS satisfaction especially for scientific and business domain applications. In recent times, metaheuristic techniques have proven to be effective for solving task scheduling problems. Such techniques include Particle Swarm Optimization (PSO), Genetic Algorithm (GA), and Ant Colony Optimization (ACO) among others. Detailed description of these techniques will be discussed in the following sections.

Genetic Algorithms for Task Scheduling

To improve the performance of Genetic Algorithm (GA) for task scheduling problems, different enhancements have been proposed focusing primarily on search mechanisms, solution representation, and fitness function. Some early works investigated influence of initial population, crossover, mutation, and reproduction operators (Delavarand Aryan, 2011; Kumar and Verma, 2012). Crossover operation exchanges information between the solutions while the mutation operation escapes the search procedure from local optima. However, these works were evaluated with moderate experiments. The way initial solution is constructed is crucial to the convergence rate and quality of the final solution. To speed up the convergence rate and enhance the quality of task schedule using GA, presence of diversity and enhanced initial solution in crucial (DelavarandAryan, 2014; Oxley *et al.,* 2015). Various heuristics such as Min-Min, Just in Time (JIT-C), Best-Fit, Round Robin, and earliest finish time (EFT) have been used to provide initial solution to ensure optimal solution, genetic diversity, and uniform coverage thus improving the global convergence (Delavar and Aryan, 2014; Xu et *al.,* 2014;Meenaet*al.,* 2016). The uniform coverage enables better spread of individuals across the search space while genetic diversity enables the search procedure to reach wider coverage of potential optimal solutions regions.

Novel encoding, decoding, crossover, and mutation operators are designed to various requirements scheduling problem and features of the cloud resource model. Solution encoding is another feature that characterizes the performance of metaheuristic algorithms when solving task scheduling problems. Mapping between tasks and virtual machines can be represented using tree data structure to encode solution for GA, this approach only needs to readjust the chromosomes when there is bound violations (Sawant, 2011;Gu et *al.,* 2012).The works used simple chromosome encoding to represent the relation between the tasks to be executed the land available resources. Another common way of chromosome encoding for task scheduling is using matrix model to record estimated execution time of tasks on available computing resources (Tayal, 2011). A random key representation to keep track of feasibility of chromosomes was presented by Ai et *al.* (2010), however, this solution encoding scheme requires extra encoding and decoding techniques for other operations of GA. Both fitness evaluation and candidate selection of metaheuristic algorithms can be considered as the selection scheme of GA. The designs of fitness functions are dependent on the goal of the scheduling problem at hand. For instance, there are various scheduling goals other than makespan, like energy consumption, resource reliability, resource availability, cost, security (Zhang et *al.,* 2014; Zhan et *al.,* 2015; Sharma et *al.,* 2016). Weighted sum approach can be used to model the impact of each scheduling goal in a composite manner when considering several objectives. Apart from roulette-wheel selection strategy, other selection strategies like tournament, and elitism have been have been used for GA based scheduling algorithms on cloud (Zhao *etal.,* 2009;Gu*etal.,* 2012;Mocanu*etal.,* 2012).

The crossover and mutation operations are used by most GA algorithms to vary the solutions and exchange information among solutions. Crossover operations merge two parents to bring forth new offspring with the aim of the resulting of offspring better than the parents, if the best qualities of the parents are inherited (Ren and Wu,2013). Mutation operations prevent the search procedure from getting stuck in local optima (Nunez et *al.,* 2013). The exchange of information among chromosomes is mostly done with one-point and two-point crossovers.Mocanu et *al.* (2012) remarked that cycle crossover is preferable to both one-point and two-point crossovers (Shen andZhang, 2011; Casas et *al.,* 2016). Swapping of the genes in chromosomes are mostly used for mutation operations in GA to avoid entrapment in local optima(Dutta and Joshi,2011; Casas et *al.,* 2016). The modified crossover and mutation operators improve the population diversity as compared to the classic GA.

To address task scheduling formulations with many objectives and constraints, multi-objective GA based tasks scheduling techniques have been proposed to find optimal trade-offs between various task scheduling objectives (Kessaci et *al.,* 2013; Wang et *al.,*2014, 2016a; Zhang et *al.,* 2017). The optimized objective are either consumer oriented or provider oriented or both, and trade-off solutions are sought by redesigning mutation and crossover operators of GA to enhance the performance of task scheduling techniques. However, these techniques incur height computational time for large scale task.

Particle Swarm Optimization for Task Scheduling

Particle Swarm Optimization (PSO) was originally designed to solve continuous optimization problems, the faster convergence quality of PSO has attracted the many researchers in applying PSO algorithms in tackling task scheduling problem on IaaS Cloud environment (Pandey et *al.,* 2010; Wu et *al.,* 2010; Chen and Zhang,2012). Therefore, various methods have been devised to redesign PSO for handling discrete optimization problems such task scheduling problem. Such methods include transformation, random key representation, and priority-based representation (Yassaet*al.,* 2013;Beegom and Rajasree, 2014; Li et *al.,* 2015b;Ambursa et *al.,* 2016). The core design issues for using PSO in solving scheduling problems is how to redesign solution trial variation operations to accommodate the requirements of the task schedule representation and how to encode the solution for PSO procedure. Encoding the task schedule solution into a particle as a pair (Tj, Rj) where the pair denotes the mapping of task Tjto resource Rj. Particles of PSO used ETC matrix to encode the solutions (Zhaoet*al.,* 2009; Wu et *al.,* 2010).

The fuzzy scheme was introduced by Liu et *al.* (2010a) to keep information about the status of the network, where the size of the fuzzy matrix is m by n as well. When decoding the fuzzy matrix, the selection is based on the maximum element of each column. However, these proposed encoding strategies consider the index of the compute resource which does capture the characteristics of the resource, which makes the particles to wander randomly if they learn using resource indexGuo*et al.*(2012a). Recently, novel encoding strategies that properly particles towards feasible and optimal solution region (Meena et *al.,* 2016; Li et *al.,* 2016). The encoding schemes modeled certain features of the IaaS Cloud to speed up the convergence rate of individuals in the search space while attaining global solution. However, the employed encoding scheme lack adequate information about a resource which may mislead the direction of particles in search space thereby resulting to poor solution, particularly for hard constrained deadlines (Rodriguez and Buyya, 2014).

These performance requirements are defined into fitness function of PSO, the fitness function determines the quality of solution obtained by PSO search procedure. To increase the search diversity of particles and convergence rate thereby improving the scheduling results, PSO suffers from premature convergence and one useful way to avoid this problem is to integrate local search method into the PSO search procedure (Guo et *al*.,2012b). To improve the local search ability of PSO and maintain population diversity, local search techniques and strengths of other metaheuristic algorithms can be integrated into PSO (Xue and Wu, 2012;Zuo et *al.,* 2014; Li et *al.,* 2015c; Nirmala and Bhanu,2016). The crossover and mutation operations improves information sharing among the particles while hill climbing and tabu search local techniques improve the quality of solution obtained by PSO (Xue and Wu, 2012; Sridhar and Babu, 2015). Robust local search techniques like Simulated Annealing (SA) (Yuan et *al.,* 2016), and Variable Neighbourhood Search (VNS) (Netjinda et *al.,* 2014b) have been hybridized with PSO to prevent possible entrapment into local optima (Zuo et *al.,* 2014; Yuan et *al.,* 2016). Furthermore, the use of chaotic sequence in replacement of random components of PSO increases solution diversity in the search space thereby improving global convergence (Li*etal.,* 2015c). However, the proposed techniques still suffers from local entrapment issue especially when solving large scale task which enlarge the search space.

Recently, multi-swarm coevolutionary strategy have employed to obtain optimal trade-off solutions for multi-objective task scheduling problems considering various objectives like energy consumption, makespan, and cost (Li et *al.,* 2015a; Yao et *al.,*2016). The proposed strategies adopted multi-swarm optimization strategy where each swarm is employed to obtain non-dominated solutions using multi-objective PSO. A novel competition and cooperation strategy is designed to avoid swarms getting trapped in local optima. However, competition and cooperation strategy may slow the convergence rate of the proposed approach.

Ant Colony Optimization for Task Scheduling

Ant Colony Optimization (ACO) have been used to solve task scheduling problems on cloud by considering various computing resources such as CPU utilization, memory utilization, and network bandwidth usage

(Lu and Gu, 2011). In addition, various features of VM like MIPS of each processor on a VM, execution time of task on a VM, the bandwidth, and average execution time of a VM can be taken into account when computing the probability for constructing the sub-solutions of ACO (Li et *al.,*2011). Several QoS requirements like reliability, response time, cost, and security can be considered when using ACO for task scheduling problems on cloud (Liu et al., 2011;Guzek et *al.,* 2015; Wu et *al.,* 2015;Mastelic et *al.,* 2015).

Furthermore, a modification to pheromone update rule was suggested by Mathiyalagan*etal.* (2010) to decide when a task is to be mapped to a computing resource, this is achieved by adding extra pheromone to the update table. Local search operation plays a vital role to improve the performance of metaheuristic algorithms, therefore, swapping of sub-solutions (tasks) between computing resources is a direct local search approach to improve the performance of ACO (Kousalya and Balasubramanie, 2009). Grouping of ants can be an effective search strategy to improve the performance of ACO. Kant et *al.* (2010) group ants into red and blank kinds, red ants try to estimate the system resource while the blacks ones determine the resource allocation.

ACO based multi-objective task scheduling technique optimized makespan and cost have been simultaneously optimized while meeting deadline and budget constraints (Zuo et *al.,* 2015). Novel performance and budget constraint handling heuristics are proposed to prevent the search procedure from getting trapped in local optima. The performance of the proposed approach is evaluated using makespan, resource utilization, deadline violation rate, and cost. However, the multiple objectives are converted into a single objective function using weighted sum approach; this can only provide one single solution which is sensitive to the assigned weights.

Other Metaheuristic Techniques for Task Scheduling

Other metaheuristic algorithms that have been applied to task scheduling includes Cat Swarm Optimization (CAT) (Gabi et *al.,* 2016a), League Championship Algorithm (LCA) (Latiff et *al.,* 2016), Simulated Annealing (Moschakis and Karatza, 2015b), Tabu Search (TS)Moschakis and Karatza (2015a), Shuffled Frog Leaping Algorithm (SFLA) (Kaur and Mehta, 2017), and Chemical Reaction Optimization (CRO)Jiang‌et*al.* (2015). Local search techniques such as Taguchi Variable Neighbourhood Search (VNS) improves the convergence speed of CSO and CRO respectively thereby improving makespan and load balance among VMs (Jiang et *al.,* 2015; Gabi et *al.,* 2016b). In the course of task execution on Cloud, task execution failure resulting from either software or hardware faults is likely to occur. Task failures can be minimized using dynamic clustering techniques alongside task migration and fault detector strategies (Latiff et *al.,*2016). However, the above task scheduling techniques were evaluated on small scale datasets which may not reveal its scalability ability.

SA and TS techniques can be utilized to minimize makespan, flowtime, and cost of executing dynamic arriving tasks on interconnected cloud environment using Least Loaded Cloud First (LLCF) to dispatch the incoming application task into different clouds (Moschakis and Karatza, 2015b,a). However, SA and TS have poor global convergence ability which makes then inappropriate for large scale tasks. Moreover, the weighted sum method for handling multiple objectives produces only one solution which may not depicts the requirements of the user.

Analysis of Metaheuristic based Task Scheduling Algorithms

Task scheduling optimization approaches either focused on single objective or multi-objective. The single objective task scheduling optimization approaches, only try to optimize either makespan or cost with some constraints, especially deadline or budget (Zuo et *al.,* 2014; Rodriguez and Buyya, 2014;Netjinda et *al.,* 2014a;Tawfeek et *al.,*2015; Li et *al.,* 2015c, 2016; Nirmala and Bhanu, 2016;Zhong et *al.,* 2016;Meena et *al.,*2016; Liu et *al.,* 2016). The constrained QoS aware algorithms attempted to optimize trade-offs between some QoS objectives without violating user imposed constraints (Lu et *al.,* 2014). However, because of the rapid development of cloud, several QoS objectives and constraints needs to be considered which makes task scheduling a multiobjective optimization problem. The complexity of the multi-objective task optimization formulation arises from the fact that users and providers have different optimization goals. Users are mainly concerned with minimizing makespan and cost while meeting certain imposed constraints, whereas providers want to maximize resource utilization and energy consumption while meeting user QoS requirements. In this situation, task scheduling have to be solved as a multi-objective optimization problem trying to optimize many and yet conflicting objectives, where it is not possible to obtain optimal solution with regards to all objectives. Therefore, a good trade-offs between the objectives need to obtained. Table 1gives examples of some recent metaheuristic based task scheduling approaches found in the literature.

Task Scheduling Optimization with Constraint Requirements

Many task scheduling optimization problems often introduce constraints which could be loose, moderate, or tight, these constraints makes some regions of search space invalid. By convention, metaheuristic algorithms are characterized by solving unconstrained optimization problems, therefore constrained optimization problems needs to be transformed unconstrained form and appropriate penalty factors are applied in the case of constraint violation. Static penalty function is one of the common constraint method handling strategies, static penalty function is usually applied to penalize infeasible solutions by decreasing their fitness values according to their degree of constraint violation. However, finding a suitable value for penalty function is difficult (Chen *etal.,* 2015b; Liu *etal.,* 2016). For instance, Rodriguez andBuyya (2014) presents PSO algorithm for solving deadline constrained cost optimization problem for workflow scheduling on cloud and used static penalty function to identify the particles violate the constraints are inferior to the feasible ones. However, this may result to premature convergence of search procedure which is a common issue with PSO.

Another common approach for constraint handling is eliminating infeasible solutions as the iterative process proceeds. However, some infeasible solutions hold vital information that are essential in guiding search direction, thus they may be useful in next generations of individuals in finding optimal solutions (Kianpisheh et *al.,* 2016;Meena*etal.,* 2016;Ambursa*etal.,* 2016).

Furthermore, Huang (2014) presented improved GA for constrained workflow scheduling problem, in their encoding approach task execution queue on VM is indicated in addition to task to VM assignment. Individuals are first evolved using the objective function and evolved population is changed when there is constrain violation. With this method there is no need to define penalty function for constraint violation. However, the approach needs to evolve for many generations which result to high computation time. To avoid the difficulty of defining problem specific factor for penalty functions, Liu *et al.* (2016) put fort a self-adaptive penalty function handle deadline constraint violation in solving cost optimization based task workflow scheduling problem using coevolutionary GA. The proposed approach is able to accelerate the convergence speed of GA while preventing premature convergence. However, the performance of GA is challenged when traversing large search space. Thus, addressing constrained task scheduling optimization problems is still an active research area.

Multi-Objective Task Scheduling Optimization Approaches

Multi-objective optimization problems involve many conflicting objectives, thus improving one objective lead to deterioration of other objectives. There is no single optimal solution that can optimize MOP with conflicting objectives, rather a set of optimal trade-off solutions known as Pareto optimal solutions. The multi-objective task scheduling optimization algorithms are categorized into aggregation, hierarchical, Pareto, and co-evolutionary approaches.

Aggregation based Multi-Objective Task Scheduling Approaches

The aggregation (weighted) approach is the common method for solving multi-objective task scheduling problems. The approach assigns weights to multiple objectives and sum up the objectives to form single objective function. For instance, Delavar and Aryan (2014) proposed GA based task scheduling algorithm to optimize makespan, reliability, and load balancing of applications by putting into consideration the heterogeneous characteristics of compute resources. Also, Shen et *al.* (2016) developed GA algorithm for adaptive scheduling of tasks considering energy consumption and makespan performance. Casas et *al.* (2016) proposed GA based task scheduling technique for optimizing makespan and cost.Zuo et *al.* (2015)proposed ACO based task scheduling algorithm to optimize budget and deadline constrained task scheduling problems, the proposed approach simultaneously makespan and cost within a given budget and deadline. However, the results of different objectives are dependent on the values of the assigned weights which may not adequately represent the decision of the user. Moreover, the approach produces only solution which is not adequate for multiobjective decision problems.

Table 1: Comparison of metaheuristic based task scheduling optimization algorithms

Reference	Objectives	Multi-Objective Approach	Tasks/ VMs	Strength	Limitation	Implementation
PSO based task scheduling optimization algorithms						
Yassa et *al.* (2013)	Makespan, cost, and Energy consumption	Hierarchical	/ 12	Optimal trade-offs between Make span, cost, and energy consumption while considering heterogeneity	Heuristic information required by hierarchical approach is difficult to determined	Not mentioned
Zuo et *al.* (2014)	Cost; constraint : deadline	Single objective	50/6	Adaptive update of particle velocities using four different velocity updating strategies to improve the capability of search mechanism for effective task scheduling	The effectiveness of the proposed method is dependent on the accurate selection of the update strategy.	Matlab
Rodriguezand Buyya(2014)	Cost; constraint: deadline	Single objective	1000 / 6	Minimized cost while meeting task deadline by considering heterogeneity and elasticity features of cloud resources.	The encoding only strategy only used the index of VM resources which could lead to slow convergence and high computation cost. It difficult to obtain feasible solution	CloudSim
Somasundaram and Govindarajan (2014)	makespan, cost, job rejection ratio; constraint: deadline	Aggregation	1000 / 500	Minimized cost, task completion time and task rejection ratio.	The multi-objective solution approach cannot depict the actual decisions of the user and slow convergence of the optimization technique.	Matlab
Netjinda et *al.*(2014a)	makespan; constraint : deadline	Single Objective	1000 /10	Considers purchasing instances and options, and instance types in minimizing cost while meeting deadline. Used variable neighborhood search to improve the obtained	Considers fixed number of purchasing instances.	Not mentioned.

Li et *al.* (2015c)	cost; constraint: deadline	Single objective	300/4	Integration of chaotic sequence improves the convergence and reduces cost of execution.	The technique is not scalable to handle large scale problem instances.	Not mentioned
Li et *al.* (2015a)	makespan and cost	Co-evolutionary multi-swarm	100/10	The technique found non- dominated solutions for makespan and cost	The underlining optimization techniques still suffers from slow convergence and scalability of the approach is not guaranteed.	Not mentioned
Li et *al.* (2016)	cost; constraints : deadline, risk rate	Single Objective	100	The solution encoding strategy considers the heterogeneity of resources and risk constraints which efficiently minimized the cost while meeting deadline and risk rate constraints.	The constraint handling strategy may eliminate some infeasible individuals that could improve performance in the next generation.	CloudSim
Nirmala and Bhanu (2016)	cost; constraints : deadline	Single Objective	400 / 6	Used catfish particle to escape local optima and avoid premature for efficient minimization of makespan while meeting deadline.	The technique is not scalable	WorkFlowSim
Zhong et *al.*(2016)	makespan	Single Objective	500 /10	Used greedy algorithm to escape local optima and avoid premature convergence for efficient minimization of makespan.	The cost of execution is not considered	CloudSim
Yao et *al.* (2016)	makespan, cost, and energy consumption	Co evolutionary multi-swarm	1000/ 20	Used multi-swarm approach with each swarm optimizing only one objective and used endocrine inspired mechanism to escape local optima and avoid premature convergence of the proposed approach.	Resource reliability is not considered.	WorkFlowSim

Verma and Kaushal (2017)	makespan, cost, energy consumption; constraints: deadline, budget	Pareto	1000 / 20	Uniform spread among Pareto Front and better convergence by hybridization with list based heuristic	Fitness assignment to individuals is difficult with this approach	CloudSim
		GA based task scheduling optimization algorithms				
DelavarandAryan (2014)	Makespan, reliability and load balancing	Aggregation	100 / 30	Best-Fit and Round-Robin heuristics as initial solution seed to improve global convergence of GA algorithm.	Cost is not considered in the optimized objectives.	Not mentioned
Tao et *al.* (2014)	Makespan and energy consumption	Pareto	30/4	Novel crossover operator and external archive update scheme for faster convergence and solution diversity.	The use case library incurs extra search and update procedure which could lead to high computation	Matlab.
Xu *et al.* (2014)	Makespan and priority	Pareto	200 / 4	Used earliest finish time heuristics to generate initial solution for solution diversity and global convergence.	Only small scale problem instances considered.	C#
Ramezaniet*al.* (2015)	Cost, task transfer time, task queue length and Energy consumption	Pareto	200 / 20	Minimized response time, makespan and cloud provider cost	Difficulty is assigning fitness values to individual search agents	CloudSim
Shojafar et *al.*(2015)	Makespan, cost, and load balance	Fuzzy	1000/ 50	Integrate fuzzy logic to improve the performance of GA.	Did not considered reliability ofresources.	CloudSim
Meena et *al.* (2016)	Cost; constraint: deadline	Single Objective	1000/	Considered heterogeneity and pay- per-use model in the solution encoding scheme.	Discarded infeasible solutions during constraint handling could improve generated solution in the subsequent generation.	Not mentioned

Shen et *al.* (2016)	Makespan and energy consumption	Aggregation	500 / 50	Adaptively minimize makespan and energy consumption.	Slow convergence and high computation cost.	CloudSim
Casas et *al.* (2016)	Makespan and cost	Aggregation	1000 / 24	Enhanced crossover and mutation operators improved the convergence speed.	Did not considered cloud pay-per- use pricing model.	VMware-vSphere
Liu et *al.* (2016)	Cost; makespan : constraint	Single objective	1000/	Efficient adaptive penalty function for hard constraint using coevolution to accelerate convergence speed and avoid premature convergence.	The encoding scheme only considers the index of compute resources which slow down the convergence speed.	WorkflowSim
Zhu *et al.* (2016)	Makespan and cost	Pareto	1000/	New encoding strategy to capture the problem specifics. Solution seed by heuristics, new mutation and crossover operators to improve global convergence.	Determining the fitness of individuals under conflicting objectives could lead to poor Pareto Front.	Not mentioned
Zhang et *al.*(2017)	Reliability and energy consumption	Pareto	100/	Used upward rank heuristic to escape from local optima.	Makespan and cost not considered.	jMetal
			ACO based task scheduling optimization algorithms			
Wu et *al.* (2013)	Makespan, and cost	Aggregation	300 / 20	Scale well with tested problem instance	Pheromone update rule did not consider the characteristics of compute resources which could lead to high computation cost.	SwinDeW-C
Zuo et *al.* (2015)	Makespan, and cost ; constraints: budget, deadline	Aggregation	600 /10	Constraint handling functions provide feedback to avoid local optima entrapment.	Pheromone update rule did not consider the characteristics of compute resources which could lead to high computation cost.	CloudSim

Tawfeek*et al.* (2015)	Makespan	Single objective	1000/ 50	Global pheromone update rule provide feedback to avoid local optima entrapment and ensure global convergence.	High computation cost.	CloudSim

Hierarchical based Multi-Objective Task Scheduling Approaches

The hierarchical approaches optimize task scheduling objectives in a sequential order, the optimization ordering of the objectives are determined based on their importance and solution to the objectives are alternately sought based on their ordering. For instance, the approach proposed byTeng et *al.* (2007) used sorting strategy, the objective functions are optimized in sequential order. The optimization of an objective is continuously carried until no further improvement is possible, then next objective is optimized while meeting the constraints of the previous optimized objectives. Similar approach was used by Zhang et *al.* (2014) to optimize makespan and cost. However, these approaches are time consuming especially when there are several objectives with constraints, since it requires several iteration of optimization process. Moreover, the importance of the objectives is dependent on the problem, and performance of the approach may be significantly affected by the ranking of the objectives.

Coevolutionary Multi-Swarm based Multi-Objective Task SchedulingApproaches

To overcome the challenges of fitness assignment problem, new efforts have been reported to use techniques for solving multi-objective task scheduling problems efficiently. These techniques are based on using multiple populations for multiple objectives for solving multi-objective problems where each population optimize one objective (Zhan et *al.,* 2013). Each population is optimized using existing optimization algorithm. Yao et *al.* (2016) proposed endocrine-based co-evolutionary multi-swarm multi-objective algorithm to find optimal trade-offs solutions between energy consumption, makespan, and cost. The proposed strategy adopted multi-swarm optimization strategy where each swarm corresponds to one objective and PSO is used to optimize each objective. A novel competition and cooperation strategy is designed to avoid swarms getting trapped in local optima. Similarly, Li et *al.* (2015a) presents coevolutionary multi-swarm PSO algorithm to obtain optimal trade-off solutions between makespan and cost. Learning between the particles is enhanced using renumber strategy (Li et *al.,* 2015b). However, the proposed techniques cannot scale well since the efficiency of PSO algorithms is challenged by local optima entrapment and imbalance between local and global search. Moreover, efficiently exchanging information between swarms and avoidances of local Pareto Fronts are still challenging issues with coevolutionary multi-swarm multi-objective task scheduling approaches.

Pareto based Multi-Objective Task Scheduling Approaches

To overcome the drawbacks of both aggregation and hierarchical approaches, Pareto-based optimization approaches have been put forth for addressing multi-objective task scheduling problems (Tao et *al.,* 2014;Durillo et *al.,* 2014). The concept of Pareto dominance is applied to assign fitness to individuals. The pareto approach does not require transforming multiple objectives into single objective formulation, and generate several trade-off solutions in a single run. Tao et *al.* (2014) presents a hybrid GA algorithms to obtain Pareto optimal solutions for makespan and energy consumption. Pareto optimal trade-offs between makespan, cost, and energy consumption was solved using list scheduling heuristics and hybrid PSO respectively (Fard et *al.,* 2014;Yassaet*al.,* 2013). Similarly,Verma and Kaushal (2017) presents PSO based multi-objective task scheduling algorithm to obtain optimal trade-offs between makespan, cost, and energy consumption while meeting deadline and budget constraints respectively.Xuet*al.* (2014) put forth multi-objective GA for workflow task scheduling problem to simultaneously minimize makespan and cost while considering the priorities of the tasks.

Moreover, Zhang et *al.* (2017) proposed multi-objective GA algorithm to obtain Pareto optimal trade-offs between energy consumption, and reliability for deadline constrained task scheduling problems. However, with Pareto task scheduling approaches, it is difficult to select appropriate individual for the next generation since Pareto dominance is a partial order (Zhan et *al.,* 2013). Therefore, the solutions obtained may not cover the entire Pareto Front (PF) if the selection operator fails to keep adequate diversity. Thus, developing multi-objective task scheduling that effectively assign fitness to individuals while keeping solution to efficiently estimate the entire PF remains challenging research.

DISCUSSION

In recent times, scientific and business community have been witnessing explosive growth of information in areas like astronomy, data mining, business informatics, and bio-informatics (Wu et *al.,* 2015). This is attributed to the speedy advancements in information and communications technology leading continuant production of large amount of data. In this context, large scale applications have emerged involving several variables which needs to be processed within a short period of time. The complexity of large scale application problems stems from the fact it has huge number of decision variables, different conflicting objectives, various types of constraints, and requires large computational time (Singh and Chana, 2016a,b). Due to the practical applications and challenges of executing large scale applications, task scheduling of applications on the large scale have become an emerging research and have attracted the attention of researchers in recent times.

It is expected that traditional task scheduling algorithms can solve large scale scheduling problems with various requirements within an acceptable period of time. However, the large scale size makes the traditional task scheduling problems inadequate to tackle large scale task scheduling problems. Moreover, traditional task scheduling problems did not have provision for multiple decision variables and objectives. Indeed, the traditional task scheduling algorithms are satisfactory for small size problems but they are not scalable as the problem size increases. Recently, Metaheuristic algorithms have shown remarkable suitability, success, and improvements for addressing task scheduling problems, however, there are still key issues and challenges with existing algorithms. The large scale task scheduling increases the computational cost and demand advanced metaheuristic based search algorithms. Therefore, there need for novel methods and algorithms to tackling large scale task scheduling problems.

Metaheuristic algorithms like PSO, ACO, and GA have shown remarkable suitability, success, and improvements for addressing task scheduling problems, however, there are still key issues and challenges with existing algorithms. The algorithmic design must be carefully done to capture the specific requirements of task scheduling problem and features of underlying Cloud infrastructures. The literature review also paid attention to the scale of task scheduling problem sizes addressed by metaheuristic algorithms. Most of the works evaluate the performance of their algorithms with relatively small number of tasks which raises concern about the scalability and applicability to large scale task scheduling problems. Solving large scale task scheduling problem instances requires high computation time and cost due to large dimension of search space that needs to be traverse by the individuals.

Task scheduling problem is NP-hard and becomes more complex when handling large problem instances (Guzek et *al.,* 2014), this requires new metaheuristic algorithms with robust global search capabilities and existing metaheuristic task scheduling algorithms still have room for further improvement. The new search algorithms should be capable of exploring the entire search space and be able to exploit the local solution regions when required. New metaheuristic algorithms may incorporate local search algorithms to form novel memetic algorithms, hybridization of different metaheuristic algorithms, hybridization of metaheuristic algorithm with heuristic algorithms (Ming andLi, 2012;Ambursa et *al.,* 2016). Hybridizing metaheuristic algorithms could improve task schedule solutions for large scale problems instances (Mezmaz et *al.,* 2011). Also, devising parallel versions of metaheuristic is also another approach to address large scale task scheduling problems (Guzek et *al.,* 2014; Liu et *al.,* 2016; Wang et *al.,*2016b). Recently, a number of metaheuristic algorithms have been proposed to solve large scale task scheduling problems (Wang et *al.,* 2016b; Liu et *al.,* 2016). To solve large scale task scheduling problems, there is need for new methods and new search algorithms. Metaheuristic approaches have shown promising results for large scale global optimization problems (Mahdavi et *al.,* 2015), which provides opportunity to betteraddress large scale task scheduling problems.

Most of the metaheuristic based multi-objective algorithms are designed for continuous optimization problems (Coello, 2006), but task scheduling problem is a discrete optimization problem. The existing metaheuristic based multi-objective task scheduling algorithms do not scale well when handling large scale problems (Durillo*et al.,* 2014; Zhang *et al.,* 2014; Li *et al.,* 2015a; Yao *et al.,* 2016). Furthermore, the two common objectives of task scheduling problem (minimizing makespan and financial cost) are conflicting that is minimizing makespan by allocating more or powerful computer resources will lead to higher cost of execution while minimization of cost by leasing few or less powerful compute will lead to longer makespan. Since the traditional multi-objective metaheuristic algorithms consider the conflicting objectives as whole, it becomes difficult to assign fitness to individuals. An individual may be better with respect to one objective but is worse with regards to another objective (Zhan et *al.,* 2013). This fitness assignment problem is a common challenge with existing metaheuristic based multi-objective algorithms which always cause search inefficiency. Moreover, besides makespan and cost, other objectives like energy consumption, reliability, resource utilization, and availability could as well be considered in multi-objective task scheduling problem. Thus, considering many conflicting objectives makes fitness assignment problem more challenging which necessitate designing suitable method for assigning fitness to individuals (Zhan *et al.,* 2013; Li *et al.,* 2015a).

CONCLUSION

This chapter presents comprehensive review of metaheuristic techniques in solving task scheduling problems, which looked at the commonly used metaheuristic algorithms for solving task scheduling focusing on the factors like solution representation, search techniques, multi-objective solution strategy, and scale of task scheduling problem instance. The associated issues and challenges are discussed. The literatures reviewed indicated that application of various metaheuristic algorithms to task scheduling problems on IaaS Cloud environment have attracted significant attention of researchers. The common approach in GA and PSO is to improve the initial solution of individuals and solution representation to increase the efficiency of search mechanisms in obtaining optimal solutions for task scheduling problems. Different metaheuristic algorithms have different features, the genetic operators search mechanisms provider better fitness for task schedules during the evolutionary, the simple update mechanism of PSO makes it to be computationally cheap, and ACO gradually build up task schedule solution using graph representation. Thus, metaheuristic algorithms or their combinations can be utilized to solve various task scheduling problems. Moreover, the common metaheuristic algorithms have been used to address task scheduling problem with relatively large scale instances. However, the algorithms still suffer from high computational cost and their performance degrades as the task instances increases.

Despite the fact that metaheuristic techniques have attain some success in solving task scheduling problems, the algorithms still face challenges and their potentials need to be investigated further. Scalability is one of the crucial issues since the number of task instances is increasing in many real-world applications due to the trend in big data. The recent developments in metaheuristic algorithms for large scale global optimization problems inspire further investigation on metaheuristic algorithms for large scale task scheduling. Thus, novel search mechanisms and solution representation schemes are necessitated for both single objective and multi-objective task scheduling problems taken into consideration the features of IaaS Cloud.

REFERENCES

Abrishami, S., Naghibzadeh, M. and Epema, D. H. (2012). Cost-driven scheduling of grid workflows using partial critical paths. *IEEE Transactions on Parallel and Distributed Systems.* 23(8), 1400-1414.

Abrishami, S., Naghibzadeh, M. and Epema, D. H. (2013). Deadline-constrained workflow scheduling algorithms for Infrastructure as a Service Clouds.

Future Generation Computer Systems. 29(1), 158-169.

Ai, L., Tang, M. and Fidge, C. J. (2010). QoS-oriented sesource allocation and scheduling of multiple composite web services in a hybrid cloud using a random-key genetic algorithm. In *17th International Conference on Neural Information Processing.* 2225 November. Sydney, N.S.W.: Springer.

Alkhanak, E. N., Lee, S. P. and Khan, S. U. R. (2015). Cost-aware challenges for workflow scheduling approaches in cloud computing environments: Taxonomy and opportunities. *Future Generation Computer Systems.* 50, 3-21.

Ambursa, F. U., Latip, R., Abdullah, A. and Subramaniam, S. (2016). A particle swarm optimization and min-max-based workflow scheduling algorithm with QoS satisfaction for service-oriented grids. *The Journal of Supercomputing,* 1-34. doi: 10.1007/s11227-016-1901-x.

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I. *et al.* (2010). A view of cloud computing. *Communications of the ACM.* 53(4), 50-58.

Beegom, A. A. and Rajasree, M. (2014). A particle swarm optimization based pareto optimal task scheduling in cloud computing. In *International Conference in Swarm Intelligence.* 17-20 October. Hefei, China: Springer, 79-86.

Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J. and Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation computer systems.* 25(6), 599-616.

Casas, I., Taheri, J., Ranjan, R., Wang, L. and Zomaya, A. Y. (2016). GA-ETI: An enhanced genetic algorithm for the scheduling of scientific workflows in cloud environments. *Journal of Computational Science.*doi:http://doi.org/10.1016/j.jocs. 2016.08.007.

Chen, H., Wang, F., Helian, N. and Akanmu, G. (2013). User-priority guided Min-Min scheduling algorithm for load balancing in cloud computing. In *Parallel Computing Technologies (PARCOMPTECH), 2013 National Conference on.* 21-23 February. Bangalore,India: IEEE, 1-8.

Chen, H., Zhu, X., Guo, H., Zhu, J., Qin, X. and Wu, J. (2015a). Towards energy-efficient scheduling for real-time tasks under uncertain cloud computing environment. *Journal of Systems and Software.* 99, 20-35.

Chen, W.-N. and Zhang, J. (2012). A set-based discrete PSO for cloud workflow scheduling with user-defined QoS constraints. In *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC).* Seoul, Korea: IEEE, 773-778.

Chen, Z.-G., Du, K.-J., Zhan, Z.-H. and Zhang, J. (2015b). Deadline constrained cloud computing resources scheduling for cost optimization based on dynamic objective genetic algorithm. In *2015 IEEE Congress on Evolutionary Computation (CEC).* IEEE, 708-714.

Coello, C. C. (2006). Evolutionary multi-objective optimization: a historical view of the field. *IEEE computational intelligence magazine.* 1(1), 28-36.

Deelman, E., Vahi, K., Juve, G., Rynge, M., Callaghan, S., Maechling, P. J., Mayani, R., Chen, W., da Silva, R. F., Livny, M. *et al.* (2015). Pegasus, a workflow management system for science automation. *Future Generation Computer Systems*. 46, 17-35.

Delavar, A. G. and Aryan, Y. (2011). A Synthetic Heuristic Algorithm for Independent Task Scheduling in Cloud System. *IJCSI.*

Delavar, A. G. and Aryan, Y. (2014). HSGA: a hybrid heuristic algorithm for workflow scheduling in cloud systems. *Cluster computing*. 17(1), 129-137.

Dikaiakos, M. D., Katsaros, D., Mehra, P., Pallis, G. and Vakali, A. (2009). Cloud computing: Distributed internet computing for IT and scientific research. *IEEE Internet computing*. 13(5), 10-13.

Durillo, J. J., Nae, V. and Prodan, R. (2014). Multi-objective energy-efficient workflow scheduling using list-based heuristics. *Future Generation Computer Systems*. 36, 221-236.

Dutta, D. and Joshi, R. (2011). A genetic: algorithm approach to cost-based multi-QoS job scheduling in cloud computing environment. In *Proceedings of the International Conference & Workshop on Emerging Trends in Technology.* Mumbai, Maharashtra, India: ACM, 422-427.

Fard, H. M., Prodan, R. and Fahringer, T. (2014). Multi-objective list scheduling of workflow applications in distributed computing infrastructures. *Journal of Parallel and Distributed Computing*. 74(3), 2152-2165.

Gabi, D., Ismail, A. S., Zainal, A., Zakaria, Z. and Abraham, A. (2016a). Orthogonal Taguchi-based cat algorithm for solving task scheduling problem in cloud computing. *Neural Computing and Applications,* 1-19.

Gabi, D., Ismail, A. S., Zainal, A., Zakaria, Z. and Abraham, A. (2016b). Orthogonal Taguchi-based cat algorithm for solving task scheduling problem in cloud computing. *Neural Computing and Applications*, 1-19.

Gogos, C., Valouxis, C., Alefragis, P., Goulas, G., Voros, N. and Housos, E. (2016). Scheduling independent tasks on heterogeneous processors using heuristics and Column Pricing. *Future Generation Computer Systems*. 60, 48-66.

Gu, J., Hu, J., Zhao, T. and Sun, G. (2012). A new resource scheduling strategy based on genetic algorithm in cloud computing environment. *Journal of Computers*. 7(1), 42-52.

Guo, L., Zhao, S., Shen, S. and Jiang, C. (2012a). A particle swarm optimization for data placement strategy in cloud computing. In *Information Engineering and Applications.* (pp. 946-953). vol. 154. Springer.

Guo, L., Zhao, S., Shen, S. and Jiang, C. (2012b). Task scheduling optimization in cloud computing based on heuristic algorithm. *Journal of Networks.* 7(3), 547-553.

Guzek, M., Bouvry, P. and Talbi, E.-G. (2015). A Survey of Evolutionary Computation for Resource Management of Processing in Cloud Computing [Review Article]. *IEEE Computational Intelligence Magazine.* 10(2), 53-67.

Guzek, M., Varrette, S., Plugaru, V., Pecero, J. E. and Bouvry, P. (2014). A holistic model of the performance and the energy efficiency of hypervisors in a high-performance computing environment. *Concurrency and Computation: Practice and Experience*. 26(15), 2569-2590.

Hameed, A., Khoshkbarforoushha, A., Ranjan, R., Jayaraman, P. P., Kolodziej, J., Balaji, P., Zeadally, S., Malluhi, Q.

M., Tziritas, N., Vishnu, A. *et al.* (2014). A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems. *Computing,* 1-24.

Huang, J. (2014). The workflow task scheduling algorithm based on the GA model in the cloud computing environment. *Journal ofSoftware*. 9(4), 873-880.

Jiang, Y., Shao, Z., Guo, Y., Zhang, H. and Niu, K. (2015). DRSCRO: A Metaheuristic Algorithm for Task Scheduling on Heterogeneous Systems. *Mathematical Problems in Engineering.* 2015.

Kalra, M. and Singh, S. (2015). A review of metaheuristic scheduling techniques in cloud computing. *Egyptian Informatics Journal.* 16(3), 275-295.

Kant, A., Sharma, A., Agarwal, S. and Chandra, S. (2010). An ACO approach to job scheduling in grid environment. In *International Conference on Swarm, Evolutionary, and Memetic Computing.* 16-18 December. Chennai, India: Springer, 286-295.

Kaur, P. and Mehta, S. (2017). Resource provisioning and work flow scheduling in clouds using augmented Shuffled Frog Leaping Algorithm. *Journal of Parallel and Distributed Computing.* 101, 41-50.

Kaur, T. and Chana, I. (2015). Energy efficiency techniques in cloud computing: A survey and taxonomy. *ACM Computing Surveys (CSUR).* 48(2), 22.

Kessaci, Y., Melab, N. and Talbi, E.-G. (2013). A Pareto-based metaheuristic for scheduling HPC applications on a geographically distributed cloud federation. *Cluster Computing.* 16(3), 451-468.

Kianpisheh, S., Charkari, N. M. and Kargahi, M. (2016). Reliability-driven scheduling of time/cost-constrained grid workflows. *Future Generation Computer Systems.* 55, 1-16.

Kousalya, K. and Balasubramanie, P. (2009). To Improve Ant Algorithm's Grid Scheduling Using Local Search. *International Journal of Intelligent Information Technology Application*. 2(2).

Kumar, P. and Verma, A. (2012). Independent task scheduling in cloud computing by improved genetic algorithm. *International Journal of Advanced Research in Computer Science and Software Engineering.* 2(5).

Latiff, M. S. A., Madni, S. H. H., Abdullahi, M. *et al.* (2016). Fault tolerance aware scheduling technique for cloud computing environment using dynamic clustering algorithm. *Neural Computing and Applications*, 1-15.

Li, H.-H., Chen, Z.-G., Zhan, Z.-H., Du, K.-J. and Zhang, J. (2015a). Renumber coevolutionarymultiswarm particle swarm optimization for multi-objective workflow scheduling on cloud computing environment. In *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation.* 25-28 May. Sendai, Japan: ACM, 1419-1420.

Li, H.-H., Fu, Y.-W., Zhan, Z.-H. and Li, J.-J. (2015b). Renumber strategy enhanced particle swarm optimization for cloud computing resource scheduling. In *2015 IEEE Congress on Evolutionary Computation (CEC).*11-15 July. Madrid, Spain: IEEE, 870-876.

Li, K., Xu, G., Zhao, G., Dong, Y. and Wang, D. (2011). Cloud task scheduling based on load balancing ant colony optimization. In *2011 Sixth Annual ChinaGrid Conference.* 22-23 August. Dalian, Liaoning, China: IEEE, 3-9.

Li, Q., Wang, Z.-y., Li, W.-h., Li, J., Wang, C. and Du, R.-y. (2013). Applications integration in a hybrid cloud computing environment: Modelling and platform. *Enterprise Information Systems.* 7(3), 237-271.

Li, X., Xu, J. and Yang, Y. (2015c). A chaotic particle swarm optimization-based heuristic for market-oriented task-level scheduling in cloud workflow systems. *Computational intelligence and neuroscience.* 2015, 81.

Li, Z., Ge, J., Yang, H., Huang, L., Hu, H., Hu, H. and Luo, B. (2016). A security and cost aware scheduling algorithm for heterogeneous tasks of scientific workflow in clouds. *Future Generation Computer Systems.* 65, 140-152.

Liu, H., Abraham, A. and Hassanien, A. E. (2010a). Scheduling jobs on computational grids using a fuzzy particle swarm optimization algorithm. *Future Generation Computer Systems.* 26(8), 1336-1343.

Liu, H., Xu, D. and Miao, H. K. (2011). Ant colony optimization based service flow scheduling with various QoS requirements in cloud computing. In *Software and Network Engineering (SSNE), 2011 First ACIS International Symposium on.* 19-20 December. Wuhan, China: IEEE, 53-58.

Liu, K., Jin, H., Chen, J., Liu, X., Yuan, D. and Yang, Y. (2010b). A compromised-time-cost scheduling algorithm in SwinDeW-C for instance-intensive cost-constrained workflows on cloud computing platform. *International Journal of High Performance Computing Applications.* 24(4).

Liu, L., Zhang, M., Buyya, R. and Fan, Q. (2016). Deadline-constrained coevolutionary genetic algorithm for scientific workflow scheduling in cloud computing. *Concurrency and Computation: Practice and Experience.* 29(5).

Lu, G., Tan, W., Sun, Y., Zhang, Z., Tang, A. *et al.* (2014). QoS constraint based workflow scheduling for cloud computing services. *Journal of Software.* 9(4), 926-930.

Lu, X. and Gu, Z. (2011). A load-adapative cloud resource scheduling model based on ant colony algorithm. In *2011 IEEE International Conference on Cloud Computing and Intelligence Systems.* 15-17 September. Beijing, China: IEEE, 296-300.

Mahdavi, S., Shiri, M. E. and Rahnamayan, S. (2015). Metaheuristics in large-scale global continues optimization: a survey. *Information Sciences.* 295, 407-428.

Malawski, M., Juve, G., Deelman, E. and Nabrzyski, J. (2015). Algorithms for cost-and deadline-constrained provisioning for scientific workflow ensembles in iaas clouds. *Future Generation Computer Systems.* 48, 1-18.

Mao, Y., Chen, X. and Li, X. (2014). Max-min task scheduling algorithm for load balance in cloud computing. In *Proceedings of International Conference on Computer Science and Information Technology.* 2123 September. Kunming, China: Springer, 457-465.

Mastelic, T., Oleksiak, A., Claussen, H., Brandic, I., Pierson, J.-M. andVasilakos, A. V. (2015). Cloud computing: Survey on energy efficiency. *ACM Computing Surveys (CSUR).* 47(2), 33.

Mathiyalagan, P., Suriya, S. and Sivanandam, S. (2010). Modified ant colony algorithm for grid scheduling. *IJCSE) International Journal on Computer Science and Engineering.* 2(02), 132-139.

Meena, J., Kumar, M. and Vardhan, M. (2016). Cost Effective Genetic Algorithm for Workflow Scheduling in Cloud Under Deadline Constraint. *IEEE Access.* 4, 50655082.

Mezmaz, M., Melab, N., Kessaci, Y., Lee, Y. C., Talbi, E.-G., Zomaya, A. Y. and Tuyttens, D. (2011). A parallel bi-objective hybrid metaheuristic for energy- aware scheduling for cloud computing systems. *Journal of Parallel and Distributed Computing.* 71(11), 1497-1508.

Ming, G. and Li, H. (2012). An improved algorithm based on max-min for cloud task scheduling. In *Recent Advances in Computer Science and Information Engineering*. (pp. 217-223). Springer.

Mocanu, E. M., Florea, M., Andreica, M. I. and Tapu§, N. (2012). Cloud computing-task scheduling based on genetic algorithms. In *Systems Conference (SysCon), 2012 IEEE International.* 19-22 March. British Columbia, Canada: IEEE, 1-6.

Morshedlou, H. and Meybodi, M. R. (2014). Decreasing impact of SLA violations: A proactive resource allocation approachfor cloud computing environments. *IEEE Transactions on Cloud Computing.* 2(2), 156-167.

Moschakis, I. A. and Karatza, H. D. (2015a). A meta-heuristic optimization approach to the scheduling of Bag-of-Tasks applications on heterogeneous Clouds with multilevel arrivals and critical jobs. *Simulation Modelling Practice and Theory.* 57,1-25.

Moschakis, I. A. and Karatza, H. D. (2015b). Multi-criteria scheduling of Bag-of-Tasks applications on heterogeneous interlinked clouds with simulated annealing. *Journal of Systems and Software.* 101, 1-14.

Netjinda, N., Sirinaovakul, B. and Achalakul, T. (2014a). Cost optimal scheduling in IaaS for dependent workload with particle swarm optimization. *The Journal of Supercomputing.* 68(3), 1579-1603.

Netjinda, N., Sirinaovakul, B. and Achalakul, T. (2014b). Cost optimal scheduling in IaaS for dependent workload with particle swarm optimization. *The Journal of Supercomputing.* 68(3), 1579-1603.

Nirmala, S. J. and Bhanu, S. M. S. (2016). Catfish-PSO based scheduling of scientific workflows in IaaS cloud. *Computing.* 98(11), 1091-1109.

Nunez, A., Merayo, M. G., Hierons, R. M. and Nunez, M. (2013). Using genetic algorithms to generate test sequences for complex timed systems. *Soft Computing.* 17(2), 301-315.

Oxley, M. A., Pasricha, S., Maciejewski, A. A., Siegel, H. J., Apodaca, J., Young, D., Briceno, L., Smith, J., Bahirat, S., Khemka, B. *et al.* (2015). Makespan and energy robust stochastic static resource allocation of a bag-of-tasks to a heterogeneous computing system. *IEEE Transactions on Parallel and Distributed Systems.* 26(10), 2791-2805.

Pandey, S., Wu, L., Guru, S. M. and Buyya, R. (2010). A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments. In *2010 24th IEEE international conference on advanced information networking and applications.* 20-23 April. Perth, Australia: IEEE, 400407.

Patel, G., Mehta, R. and Bhoi, U. (2015). Enhanced Load Balanced Min-min Algorithm for Static Meta Task Scheduling in Cloud Computing. *Procedia Computer Science.* 57,545-553.

Ramezani, F., Lu, J., Taheri, J. and Hussain, F. K. (2015). Evolutionary algorithm-based multi-objective task scheduling optimization model in cloud environments. *World Wide Web.* 18(6), 1737-1757.

Ranaldo, N. and Zimeo, E. (2009). Time and cost-driven scheduling of data parallel tasks in grid workflows. *IEEE Systems Journal.* 3(1), 104-120.

Ren, Y. and Wu, Y. (2013). An efficient algorithm for high-dimensional function optimization. *Soft Computing.* 17(6), 995-1004.

Rodriguez, M. A. and Buyya, R. (2014). Deadline based resource provisioningand scheduling algorithm for scientific workflows on clouds. *IEEE Transactions on Cloud Computing*. 2(2), 222-235.

Sawant, S. (2011). A genetic algorithm scheduling approach for virtual machine resources in a cloud computing environment.

Sharma, Y., Javadi, B., Si, W. and Sun, D. (2016). Reliability and Energy Efficiency in Cloud Computing Systems: Survey and Taxonomy. *Journal of Network and Computer Applications.*

Shen, G. and Zhang, Y.-Q. (2011). A shadow price guided genetic algorithm for energy aware task scheduling on cloud computers. In *International Conference in Swarm Intelligence.* 12-15 June. Chongqing, China: Springer, 522-529.

Shen, Y., Bao, Z., Qin, X. and Shen, J. (2016). Adaptive task scheduling strategy in cloud: when energy consumption meets performance guarantee. *World Wide Web*, 1-19.

Shojafar, M., Javanmardi, S., Abolfazli, S. and Cordeschi, N. (2015). FUGE: A joint meta-heuristic approach to cloud job scheduling algorithm using fuzzy theory and a genetic method. *Cluster Computing.* 18(2), 829-844.

Singh, S. and Chana, I. (2016a). QoS-aware autonomic resource management in cloud computing: a systematic review. *ACM Computing Surveys (CSUR)*. 48(3), 42.

Singh, S. and Chana, I. (2016b). A survey on resource scheduling in cloud computing: Issues and challenges. *Journal of Grid Computing.* 14(2), 217-264.

Somasundaram, T. S. and Govindarajan, K. (2014). CLOUDRB: A framework for scheduling and managing High-Performance Computing (HPC) applications in science cloud. *Future Generation Computer Systems.* 34, 47-65.

Sridhar, M. and Babu, G. R. M. (2015). Hybrid particle swarm optimization scheduling for cloud computing. In *Advance Computing Conference (IACC), 2015 IEEE International.* IEEE, 1196-1200.

Srinivasan, A., Quadir, M. A. and Vijayakumar, V. (2015). Era of cloud computing: a new insight to hybrid cloud. *Procedia Computer Science.* 50, 42-51.

Tao, F., Feng, Y., Zhang, L. and Liao, T. (2014). CLPS-GA: A case library and Pareto solution-based hybrid genetic algorithm for energy-aware cloud service scheduling. *Applied Soft Computing.* 19, 264-279.

Tawfeek, M. A., El-Sisi, A., Keshk, A. and Torkey, F. A. (2015). Cloud task scheduling based on ant colony optimization. *Int. Arab J. Inf. Technol.* 12(2), 129-137.

Tayal, S. (2011). Tasks scheduling optimization for the cloud computing systems. *IJAEST-INTERNATIONAL JOURNAL OF ADVANCED ENGINEERING SCIENCES AND TECHNOLOGIES.* 1(5), 111-115.

Teng, S., Lee, L. H. and Chew, E. P. (2007). Multi-objective ordinal optimization for simulation optimization problems. *Automatica.* 43(11), 1884-1895.

Tilak, S. and Patil, D. (2012). A survey of various scheduling algorithms in cloud environment. *International Journal of Engineering Inventions.* 1(2), 36-39.

Tsai, C.-W. and Rodrigues, J. J. (2014). Metaheuristic scheduling for cloud: A survey. *IEEE Systems Journal.* 8(1), 279-291.

Varalakshmi, P., Ramaswamy, A., Balasubramanian, A. and Vijaykumar, P. (2011). An optimal workflow based scheduling and resource allocation in cloud. In *International Conference on Advances in Computing and Communications*. Springer, 411-420.

Verma, A. and Kaushal, S. (2017). A hybrid multi-objective Particle Swarm Optimization for scientific workflow scheduling. *Parallel Computing.* 62, 1-19.

Wang, W.-J., Chang, Y.-S., Lo, W.-T. and Lee, Y.-K. (2013). Adaptive scheduling for parallel tasks with QoS satisfaction for hybrid cloud environments. *The Journal of Supercomputing.* 66(2), 783-811.

Wang, X., Wang, Y. and Cui, Y. (2014). A new multi-objective bi-level programming model for energy and locality aware multi-job scheduling in cloud computing. *Future Generation Computer Systems*. 36, 91-101.

Wang, X., Wang, Y. and Cui, Y. (2016a). An energy-aware bi-level optimization model for multi-job scheduling problems under cloud computing. *Soft Computing*. 20(1), 303-317.

Wang, Y., Li, Y., Chen, Z. and Xue, Y. (2016b). Cooperative particle swarm optimization using MapReduce. *Soft Computing,* 1-11.

Wu, F., Wu, Q. and Tan, Y. (2015). Workflow scheduling in cloud: a survey. *The Journal of Supercomputing.* 71(9), 3373-3418.

Wu, Q., Yun, D., Lin, X., Gu, Y., Lin, W. and Liu, Y. (2012). On workflow scheduling for end-to-end performance optimization in distributed network environments. In *Workshop on Job Scheduling Strategies for Parallel Processing.* 25 May. Shanghai, China: Springer, 76-95.

Wu, Z., Liu, X., Ni, Z., Yuan, D. and Yang, Y. (2013). A market-oriented hierarchical scheduling strategy in cloud workflow systems. *The Journal of Supercomputing*. 63(1), 256-293.

Wu, Z., Ni, Z., Gu, L. and Liu, X. (2010). A revised discrete particle swarm optimization for cloud workflow scheduling. In *Computational Intelligence and Security (CIS), 2010 International Conference on.* 11-14 December. Guangxi Zhuang Autonomous Region, China: IEEE, 184-188.

Xu, Y., Li, K., Hu, J. and Li, K. (2014). A genetic algorithm for task scheduling on heterogeneous computing systems using multiple priority queues. *Information Sciences.* 270, 255-287.

Xue, B., Zhang, M., Browne, W. and Yao, X. (2016). A survey on evolutionary computation approaches to feature selection. 20(4), 606 - 626.

Xue, S.-J. and Wu, W. (2012). Scheduling workflow in cloud computing based on hybrid particle swarm algorithm. *Indonesian Journal of Electrical Engineering and Computer Science.* 10(7), 1560-1566.

Yao, G., Ding, Y., Jin, Y. and Hao, K. (2016). Endocrine-based coevolutionarymultiswarm for multi-objective workflow scheduling in a cloud system. *Soft Computing,* 1-14.

Yassa, S., Chelouah, R., Kadima, H. and Granado, B. (2013). Multi-objective approach for energy-aware workflow scheduling in cloud computing environments. *The Scientific World Journal.* 2013.

Yu, J. and Buyya, R. (2005). A taxonomy of workflow management systems for grid computing. *Journal of Grid Computing.* 3(3-4), 171-200.

Yuan, H., Bi, J., Tan, W., Zhou, M., Li, B. H. and Li, J. (2016). TTSA: an effective scheduling approach for delay bounded tasks in hybrid clouds. *IEEE transactions on cybernetics.*

Zhan, Z.-H., Li, J., Cao, J., Zhang, J., Chung, H. S.-H. and Shi, Y.-H. (2013). Multiple populations for multiple objectives: A coevolutionary technique for solving multiobjective optimization problems. *IEEE Transactions on Cybernetics.* 43(2), 445-463.

Zhan, Z.-H., Liu, X.-F., Gong, Y.-J., Zhang, J., Chung, H. S.-H. and Li, Y. (2015). Cloud computing resource scheduling and a survey of its evolutionary approaches. *ACM Computing Surveys (CSUR).* 47(4), 63.

Zhang, F., Cao, J., Li, K., Khan, S. U. and Hwang, K. (2014). Multi-objective scheduling of many tasks in cloud platforms. *Future Generation Computer Systems.* 37, 309320.

Zhang, H., Jiang, G., Yoshihira, K., Chen, H. and Saxena, A. (2009). Intelligent workload factoring for a hybrid cloud computing model. In *2009 Congress on Services-I.* 6-10 July. Los Angeles, CA: IEEE, 701-708.

Zhang, L., Li, K., Li, C. and Li, K. (2017). Bi-objective workflow scheduling of the energy consumption and reliability in heterogeneous computing systems. *Information Sciences.* 379, 241-256.

Zhang, Q., Cheng, L. and Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges. *Journal of internet services and applications.* 1(1), 7-18.

Zhao, C., Zhang, S., Liu, Q., Xie, J. and Hu, J. (2009). Independent tasks scheduling based on genetic algorithm in cloud computing. In *2009 5th International Conference on Wireless Communications, Networking and Mobile Computing.* 2426 September. Beijing, China: IEEE, 1-4.

Zhao, Q., Xiong, C., Yu, C., Zhang, C. and Zhao, X. (2016). A new energy-aware task scheduling method for data-intensive applications in the cloud. *Journal of Network and Computer Applications.* 59, 14-27.

Zhong, Z., Chen, K., Zhai, X. and Zhou, S. (2016). Virtual machine-based task scheduling algorithm in a cloud computing environment. *Tsinghua Science and Technology.* 21(6), 660-667.

Zhu, Z., Zhang, G., Li, M. and Liu, X. (2016). Evolutionary multi-objective workflow scheduling in cloud. *IEEE Transactions on Parallel and Distributed Systems.* 27(5), 1344-1357.

Zuo, L., Shu, L., Dong, S., Zhu, C. and Hara, T. (2015). A Multi-Objective Optimization Scheduling Method Based on the Ant Colony Algorithm in Cloud Computing. *IEEE Access*. 3, 2687-2699.

Zuo, X., Zhang, G. and Tan, W. (2014). Self-adaptive learning PSO-based deadline constrained task scheduling for hybrid IaaS cloud. *IEEE Transactions on Automation Science and Engineering.* 11(2), 564-573.