A Multi-Objective Approach With WASPAS Decision-Making for Workflow Scheduling in Cloud Environment

Fatemeh Ebadifard
Department of Computer,
University of Kashan, Kashan, Iran
Ebadifard.fatemeh@gmail.com

Seyed Morteza Babamir*
Department of Computer
University of Kashan, Kashan, Iran
Babamir@kashanu.ac.ir

Received: 2017/07/02 Revised: 2017/12/09 Accepted: 2018/03/13

***Abstract*— A workflow consists of a set of independent tasks, while workflow scheduling in a cloud environment is a proper permutation of these tasks involving virtual machines. Selecting the permutation with minimum completion time from among all of the arrangements, in which the requests and diversity of virtual machines increase, is an NP-hard problem. Given that, in addition to the makespan, other objectives should be considered in the scheduling problem in a real environment, which, in most cases, are conflicting objectives, the scheduling problem becomes more complicated. Therefore, multi-objective heuristic algorithms represent the perfect solution to these problems.**

To this end, we extended a recent heuristic algorithm known as black hole optimization (BHO) and presented a multi-objective scheduling method for a workflow application based on the Pareto optimizer algorithm. Since multi-objective algorithms select a set of permutations with an optimal trade-off from among conflicting objectives, we use a decision-making method – the weighted aggregated sum product assessment (WASPAS) – in the following and select a solution that offers suitable permutation from among all solutions of the Pareto optimal set. Our proposed method is able to consider user requirements, as well as the interests of service providers. Using a balanced and unbalanced workflow, we compare our proposed method with the SPEA2 and NSGA2 algorithms based on conflicting objectives: (1) makespan, (2) cost and (3) resource efficiency.

Keywords—cloud; makespan; Cost; effeciency; WASPAS.

1. INTRODUCTION

Cloud environment provides a huge context of servers in the data center, so that when users request resources, provides them in shared mode [1]. Benefiting from a "pay as you go" model has made cloud services ubiquitous. Cloud computing architecture consists of three different layers namely Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Cloud providers, Service Providers and users are three distinct entities incorporating in SaaS level service provisioning.

A workflow is a common approach to modeling the most scientific applications in distributed systems. Typically, a workflow is displayed as a directed acyclic graph, where each task is associated with a node, while the connection between tasks is displayed using edges. Given the importance of workflow applications, extensive research has been carried out in recent years on the scheduling of the workflow in a cloud environment. The workflow scheduling on the resources is to choose the appropriate resource for a task as it dependent tasks have been already run. This resource selection and task allocation on them depends on the desired quality of service requirements for different users, so that the scheduling is a nondeterministic polynomial time (NP)-hard problem [2].

Most previous works on these issues have considered one of the requirements concerning the quality of service; for example, in most makespan cases, the completion time of the workflow is considered. In addition to the makespan, which is one of the most important factors in the task scheduling of workflows, there are some important factors concerning service quality, such as cost, performance, reliability and security in a cloud environment. Therefore, an appropriate task-scheduling algorithm should strike a balance between some of the objectives relating to the quality of service. This issue is known as multi-objective task-scheduling; there are different approaches to solving it. One of these methods is to use Pareto-based optimizer algorithms, which allow users to select their best outcomes from a proper solution set.

In this paper, we have proposed a multi-objective workflow scheduling algorithm using black hole heuristic algorithm [3]. As the proposed method, it can simultaneously consider the quality of service requirements on the part of both the service provider and the customer. The aim of the proposed algorithm is to use the Pareto optimizer to satisfy the following requirements: 1. reducing the time to complete the workflow (makespan); 2. Reducing the cost for the customer; and 3. Increasing resource efficiency. To achieve an optimal solution for the three above objectives, we present a Pareto-based BHO algorithm.

A single solution (trade-off), which simultaneously optimizes such conflicting objectives, is called a Pareto optimal solution. When there is a set of candidates for the optimal solution, the set is called the Pareto front. Regarding the Pareto front, no solution can dominate another. Since the output of heuristic algorithms, rather than a proper permutation of requests made to virtual machines, involves a set of solutions, selecting the proper permutation for service providers becomes difficult [4]. In this article, we use one of the most recognized processes in the area of decision-making, that is, the WASPAS. This process allows service providers to express their semantic requirements and gives an accurate weight to objectives based on user preferences, thus selecting the best solution from the Pareto optimal set with the help of given priorities.

The WorkflowSim tool [5] has been used to evaluate the proposed method, which is a CloudSim open-source development tool [6]. We have extended the initial core of this tool to provide our algorithm, as well as compared the proposed method with previous Pareto-based algorithms, such as SPEA2 [7] and NSGA2 [8].

Various efficiency metrics have been proposed to measure the quality of the Pareto optimal set. In turn, they have been used to compare the efficacy of different multi-objective optimization algorithms. Ideally, the Pareto optimal solutions must be accurate, well distributed and widely spread [9]. These three common efficacy metrics (distance-based distribution, coverage ratio and maximum spread ratio) have been used to compare the archive collections (Pareto front) obtained from the proposed algorithm and other algorithms. In summary, our main motivations in this article are as follows:

1. Presenting a multi-objective scheduling method based on the black hole algorithm for scheduling the workflow in a cloud environment.
2. Converting the single-objective algorithm of the black hole into a multi-objective algorithm by using Pareto optimization.
3. Considering the interests of service providers and users simultaneously by defining a fitness function to decrease makespan and cost and increase the resource efficiency.
4. Using a weighted aggregated sum product assessment (WASPAS) decision-making process for selecting an optimal solution from among the Pareto optimal set based on the preferences of users.
5. Evaluating through performance metrics on the different workflow for ensuring efficiency of result which obtained through the proposed method.

The rest of the paper consists of the following sections: In Section 2, related work is expressed in terms of workflow scheduling, in Section 3, the mathematical model of workflow scheduling and details of optimization purposes used have been investigated. The forth section expressed the details of the proposed method in single-objective and Pareto-based multi objective. Section 5 introduces a decision-making method for finding optimal solution in Pareto-individuals named WASPAS. Section 6 demonstrates simulation results and evaluation of the proposed method and Section 7 concludes the paper and discusses some future work.

2. Related Work

Workflow scheduling in distributed systems has been given much attention in recent years. Finding fully optimized solutions for task scheduling issues is almost impossible due to the NP-hardness. The purpose of the existing algorithms is to provide proper solutions which are close to the optimal state. Many algorithms have been provided aimed at finding appropriate solutions to meet the quality of service. Each of these solutions provides one or more of the requirements of the quality of service. In the following, some of the solutions presented in recent years using heuristic algorithms are discussed, considering two categories: single-objective and multi-objective algorithms.

2.1. Single-Objective Heuristic Algorithms:

The main aim of the single-objective algorithm is to provide one of the requirements of the quality of service for users or service providers. Pandey et al. [10] presented a PSO-based workflow scheduling method aimed at reducing executive costs for the workflow. Load balancing methods were also used in this algorithm to balance the workloads on virtual machines. Yu et al. [11] used a genetic algorithm for workflow scheduling in the grid environment. The purpose of this algorithm was to reduce task completion time by considering the constraint of the user's budget. Keshanchi et al. [12] used a genetic algorithm for workflow scheduling in the cloud environment using priority queues. The purpose of the proposed algorithm was to reduce the makespan by considering a heuristic-based HEFT search to assign subtasks to processors.

2.2. Multi-Objective Heuristic Algorithms:

In multi-objective scheduling algorithms, several objectives are considered at the same time in order to provide an efficient scheduling method. Bligaiyan et al. [13] presented a workflow scheduling algorithm using a cat swarm optimization algorithm for workflow in the cloud environment. The purpose of the presented algorithm was to reduce makespan, cost and the idle time of the processor. The authors compared the proposed method with a MOPSO algorithm, and simulation results in MATLAB showed that the proposed method was better than MOPSO. Udomkasemsub et al. [14] presented a multi-objective scheduling method using an ABC algorithm with respect to objectives such as cost and makespan, using a Pareto optimizer algorithm. Wu et al. [15] proposed a multi-objective scheduling method using an RDPSO algorithm for scheduling workflow in the cloud, aimed at reducing the cost and makespan. Yassa et al. [16] presented a multi-objective workflow scheduling method using a MODPSO algorithm to reduce costs, energy and makespan. These authors used a DVFS technique to reduce the cost and then compared the proposed method with HEFT. Most previous studies have used the cost and makespan as two important factors in scheduling. Kaur et al. [17] presented a workflow scheduling method using an augmented shuffled frog-leaping algorithm to reduce execution cost while meeting the specified deadline. The proposed method was compared with a PSO algorithm, and simulation results in WorkflowSim showed that this method was better than the PSO method for reducing the overall execution cost of the considered workflows.

Khalili et al. [18], in addition to taking into account the interests of users, considered the requirements of quality service for service providers. Using a GWO algorithm and a

Pareto optimizer, a multi-objective scheduling algorithm was developed with the aim of reducing makespan, time and costs and increasing throughput, and compared with an SPEA2 algorithm. We have used a black hole heuristic algorithm in our proposed method; this algorithm has not previously been used for workflow scheduling problems. We formed this algorithm as a multi-objective algorithm using a Pareto optimizer function. Using this, we have provided a proper method to investigate the problem of task scheduling in the cloud environment to reduce costs and makespan and to increase the efficiency of resources.

3. Problem Formulation

A workflow application is indicated by W = (T, E) as a discrete acyclic graph where T = {t_1, t_2, …, t_n} is a set of tasks and E is a set of edges. If there is an edge e_{ij} between two tasks t_1 and t_2, then t_i is the parent and t_j is the child. According to this definition, a child may not be run unless the parent is running. Figure 1 shows an example workflow, where each node represents a task and each edge shows the connection between tasks. The numbers on the edges are the costs of the connections between each pair of nodes. In the following, the quality of service objectives that we have applied to the scheduling problem will be explained.

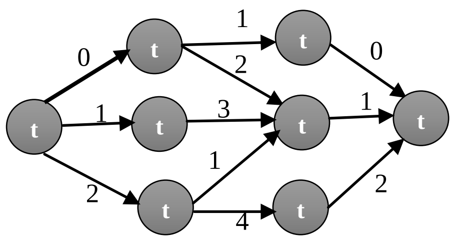

Fig. 1. The workflow sample

Makespan

Suppose VM = {VM_1, VM_2, ..., VM_m} is a set of virtual machines and Task = {t_1, t_2, ..., t_n} is a set of tasks which can be run on the resources. If the request completion time ti on VM_j is indicated by $CT_{ti}^{VM_j}$, the makespan is defined by Equation (1).

$$Makespan = \max_{1 \le j \le m} \sum_{i=1}^{n} CT_{ti}^{VM_j} \times x_{ij} \qquad (1)$$

If the request t_i is run on VM_j, the value of x_{ij} is equal to one, otherwise it is zero. For example, if there are three virtual machines and requests are run on the machines according to Figure 2, the makespan is determined as follows.

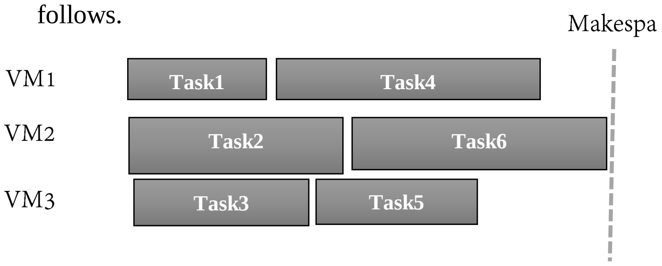

Fig. 2. Completion time of a typical workflow

Since the worst value of makespan is given by the FCFS algorithm [19], we consider the upper limit of makespan as its value in the FCFS algorithm.

Cost

In task scheduling problems in the cloud, the computational cost for each customer is determined based on his use of the resources at any time. The cloud provider takes into account three types of costs for each request, as follows.

3.1. Computational Cost:

This cost is calculated based on the number of millions of instructions per second (MIPS) for each request. For example, for request t_i, the computational cost is calculated using Equation (2).

$$C_p(t_i) = ET_{ti}^{VMj} \times \mathrm{Cos}\,tPer\,\mathrm{Pr}\,occes\sin ginVmj \qquad (2)$$

Here, ET_{ti}^{VMj} is the execution time of the request ti on the virtual machine VM_j, which is calculated using Equation (3):

$$ET_{t_i}^{VM_j} = \frac{MI(t_i)}{MIPS(VM_j)} \qquad (3)$$

where *MI* (t_i) is the number of instructions for request *i* and *MIPS* (VM_j) is the number of instructions that machine *j* runs in one second.

3.2. Hosting Time per Second:

This cost is calculated based on the duration of tasks on the virtual machine, according to Equation (4):

$$C_s(t_i) = (ET_{ti}^{VMj} + WT_{t_i}^{VMj}) \times \mathrm{Cos}\,tPerStoragInVMj \qquad (4)$$

where $WT_{t_i}^{VMj}$ is the waiting time of request ti on the virtual machine VM_j, which depends on the provision of the required files from its parent and is calculated according to Equation (5).

$$WT_{ti}^{VMj} = \frac{\max input(t_i)}{BW} \qquad (5)$$

3.3. The Cost of Data Transmission:

The cost that request ti must pay for transfer of its file set to the children is calculated according to Equation (6).

$$C_T(t_i) = \frac{\sum Output(t_i)}{BW} \times \mathrm{Cos}\,tPerTransfer \qquad (6)$$

The total cost is calculated using Equation (7).

$$C_{total}(t_i) = C_p(t_i) + C_s(t_i) + C_T(t_i) \qquad (7)$$

The cloud provider calculates the operating cost of request ti based on the total cost mentioned above. To find the upper limit of the cost, it has been assumed that all the requests are assigned to machines with high processing power, high-speed memory storage and high-power transmission.

Resource Efficiency

For each virtual machine, resource efficiency is defined according to Equation (8). If request ti is run on VMj, xij is equal to one, otherwise it is zero.

$$TH_{VM_j} = \frac{\sum_{i=1}^{n} MI(t_i) \times x_{ij}}{\sum_{i=1}^{n} CT_{t_i}^{VM_j} \times x_{ij}} \tag{8}$$

4. THE PROPOSED APPROACH

This section first introduces standard black hole algorithms, and then discusses how they are used to solve the multi-objective scheduling problem using a Pareto optimizer algorithm for the workflow.

-Standard Black Hole Algorithm

The black hole algorithm was first presented in 2013 by Hemmatloo [3]. A black hole algorithm is a population-based method that has some features in common with other population-based methods. Like other population-based algorithms, a community is produced via a candidate solution for the problem and is distributed randomly in the search space. Population-based algorithms include the populations generated toward the optimal solution via certain mechanisms. In the black hole algorithm, the evolution of the population is achieved by moving all the candidates toward the best candidate, called the black hole, in each iteration. Those candidates which come within the scope of black holes will be replaced by new candidates produced in the search space.

Like other population-based algorithms, in the proposed black hole algorithm, a population produced by the candidate solutions (stars) is generated randomly in the search space. After initialization, the population fitness is assessed and the best candidate in the population (the candidate with the best fitness) is selected as the black hole. The other candidates are normal stars. After setting the black holes and stars, the black holes begin to attract the stars around them, and all the stars start moving towards a black hole. Stars move towards a black hole, according to Equation (9):

$$x_i(t+1) = x_i(t) + rand \times (x_{BH} - x_i(t)) \quad i = 1, 2, ..., N \tag{9}$$

where x_i is the i^{th} star at time t and t + 1, x_{BH} is the position of the black hole in the search space, “rand” is a random number in the interval [0,1] and N is the number of stars (candidate solutions).

Moving towards the black hole, a star may obtain a place at a lower cost than a black hole. In this case, the black hole moves toward the star’s position and vice versa. Then the black hole algorithm will continue from a new position and stars will move towards this new position.

The possibility of passing the horizon also exists during the movement of the stars toward black holes. Each star that passes the horizon of a black hole will be sucked in by the black hole. When a star dies, (i.e., is sucked in by the black hole), a new star is born and is randomly distributed in the search space, and a new search is started. This action is performed in order to keep the number of candidate solutions constant. The next iteration begins after all the stars have moved. The radius of the horizon in the black hole algorithm is calculated using Equation (10).

$$R = \frac{f_{BH}}{\sum_{i=1}^{n} f_i} \tag{10}$$

F_{BH} is the fitness of the black hole, f_i represents the fitness of the i^{th} star and N is the number of stars. When the distance between a candidate solution (star) and a black hole is less than R, that candidate falls in and a new candidate will be created and randomly distributed in the search space. The steps of black hole algorithm can be summarized in the form of pseudocode as shown in Algorithm 1.

Initialize a population of stars with random locations in the search space
Loop
For each star, evaluate the objective function
Select the best star that has the best fitness value as the black hole
Change the location of each star according to Eq. (9)
If a star reaches a location with lower cost than the black hole, exchange their locations
If a star crosses the event horizon of the black hole, replace it with a new star in a random location in the search space
If a termination criterion (a maximum number of iterations or a sufficiently good fitness) is met, exit the loop
End loop

Algorithm1: The black hole algorithm pseudo code [3]

-Pareto-Based Black Hole Algorithm

For solving workflow multi-objective scheduling problems, we have extended the black hole algorithm using a Pareto optimizer algorithm, thus producing the proposed PBHO algorithm. For transforming BH to PBHO for each star, two merit values, *R* and *S*, are considered. According to the number of stars which dominate in the population and the archive set, the power S is assigned, and the R-value is assigned according to Equation (11) for each star based on the merit value S of stars which dominate it. If the star overruns the upper limit of the runtime and the upper limit of the cost, its value of R is a large number.

$$R(i) = \sum_{\substack{j=1 \\ j \in p+a \\ j \succ i}}^{n} (S_j) \tag{11}$$

j> i in the above equation is called Pareto overcome symbol.

In the equation, stars with a lower R-value have higher fitness, because they are dominated by stars with less power. The star with the lowest value of *R* is considered to be the BH. Based on the BH position in each stage, the position of all the stars of the initial population are updated and then merged with the archive set, and the merit values of *S* and *R* are calculated for all members of the archive and initial population sets. Members of the newly formed population are sorted in terms of *R* in ascending order. Members of the population that have the same R-value are sorted in terms of *S*. The archive and population set members are thus primarily sorted based on *R* and at the second level in terms of *S*. At the next stage, a number of members equal to the number of archive members are selected from the above sorted list and transferred to the archive set. This cycle of steps is iterated until the ending conditions are fulfilled.

Non-dominated answers obtained from solving multi-objective optimization problems (archived) are often known as the Pareto front. Depending on the conditions each can be considered as the optimal decision, and none of the answers on the other side of the Pareto front is considered a priority. Algorithm 2 displays the pseudocode of the PBHO algorithm.

```
PBHO Algorithm:
Initialize the population of star and create an empty archive
    While (t< Max number of iterations)
      Calculate the fitness vector of all stars based on objective functions
      by Eq.(1), (7),(8)
      Calculate fitness of Pareto based on Eq. (11)
       Copy top ten non-dominated stars in population to the archive
       Select Black hole from the archive
      For each star
            Update the position of current star using Eqs.(9)
            Calculate fitness of Pareto based on Eq. (11)
            If a star crosses the event horizon of the black hole, replace it
            with a new star in a random location in the search space
      End while
Return the archive
```

Algorithm2: PBHO algorithm

5. Applications of WASPAS Method in Multi-Criterion Decision-Making

Multi-objective scheduling methods provide a set of optimal solutions. Each of these solutions represents a trade-off between conflicting objectives. In such cases, the user should make proper choices from among the Pareto optimal set based on his or her preferences and requirements. Hence, a mechanism is required for weighting the criteria based on priorities. Weighted aggregated sum product assessment (WASPAS) is one of the most efficient methods in the field of decision-making. This method was first developed by Zavadskas et al. [20] and it has the capability of accurately ranking the alternatives in all the considered selection problems.

-WASPAS Method

The main steps of WASPAS include determining attributes and alternatives, performing calculations to determine the weights of attributes, decision-making and selecting the definitive solution.

Determining Attributes and Alternatives:

Every multi-criterion decision-making problem starts with the following decision matrix:

$$A = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{21} & \dots & x_{2n} \\ & & & \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix}$$

where n is the number of attributes that should be optimized (in this case study the three objectives of makespan, cost and resource efficiency) and m represents candidate alternatives (in this study the members of the Pareto front) from which one alternative should be selected.

Weight Calculation:

Weights (w_i) of attributes are calculated using Equation (12).

$$W_j = \frac{S_j}{\sum_{j=1}^{n} S_j} \tag{12}$$

Here, S_j is the degree of importance of the jth attribute, and j is number of attributes. In this paper we consider three attributes. Therefore, n = 3 and the value of S_j can be between one and three where three is the highest degree of importance. Therefore, the attribute with $S_j = 3$ has the highest degree of impact in finding the optimal solution and that with $S_j = 1$ is the least preferable attribute.

Decision-Making and Selecting a Definitive Solution:

The final step of WASPAS involves determining the rank of each decision-making alternative. In the initial step for determining the rank of each alternative, we apply the normalizer function for the values of each of the attributes in accordance with Equation (13) if maxi xij is preferable, or using Equation (14) if this is not the case.

$$\bar{x}_{ij} = \frac{x_{ij}}{\underset{i}{Max}\ x_{ij}} \tag{13}$$

$$\bar{x}_{ij} = \frac{\underset{i}{Min}\ x_{ij}}{x_{ij}} \tag{14}$$

The WASPAS method is a combination of two well-known MCDM approaches. The first criterion of optimality is a weighted sum model (WSM), calculated using Equation (15) for all alternatives. The other criterion is a weighted product model (WPM), calculated using Equation (16).

$$Q^1 = \sum_{j=1}^{n} \bar{x}_{ij} \times w_j \tag{15}$$

Where w_j is weight of of j^{th} attribute.

$$Q^2 = \prod_{j=1}^{n} (\bar{x}_{ij})^{w_j} \tag{16}$$

Therefore, the final equation of WASPAS is Equation (17).

$$Q_i = 0.5Q^1 + 0.5Q^2 \tag{17}$$

Now, the candidate alternatives are ranked based on the Q-values, and the member that has the highest Q-value has higher priority among the group members.

6. Evaluation of Results

We have used open-source, WorkflowSim tools to evaluate the proposed methods, and compared our results with three known radiation optimizer-based algorithms NSGA2 [8], SPEA2 [7] and PGWO [18].

We used the real workflow library presented by Bahraini et al. [21] to evaluate the proposed method. This library enables the structure of five real workflows to be studied: 1. Montage (astronomy), 2. Cybershake (seismology), 3. Epigenomics (biological sciences), 4. LIGO (gravitational physics) and 5. Sipht (biology). We used two workloads: The Epigenomics balanced workload and the Montage unbalanced workload. Figure 3 shows a small sample of these two workflows.

Our experiment environment includes a data center that consists of twenty similar hosts with virtualization capability. In fact, it is assumed that virtualizers like Xen are installed on them that they can share sources. Host specifications are according to Table 1.

We have put 60 virtual machines on this data center; characteristics of this virtual machine are given in Table 2. Parameters for simulation for each of the algorithms PBHO, NSGA2 and SPEA2 are also given in Table 3.

We evaluated the proposed method in two parts. Firstly, the proposed method was evaluated on the basis of metrics which evaluate multi-objective algorithms. Secondly, the proposed method was evaluated based on makespan, cost, and resource efficiency objectives.

-Evaluation Using Metrics:

There are various metrics for assessing the quality of the Pareto optimal set in multi-objective algorithms enabling multi-objective algorithms to be compared with each other [23-25]. The three metrics used for most multi-objective algorithms are: 1. the coverage ratio, 2. the maximum spread and 3. the distance-based distribution. Therefore, we have also used these metrics to evaluate the quality of the Pareto optimal set in the NSGA2, PBHO, SPEA2 and PGWO [18] algorithms. We explain the results and give further details in the following sections.

Coverage Ratio:

The coverage ratio is used to compare two Pareto fronts and is defined as the fraction of solutions in the PBHO Pareto front that dominate other solutions in the other algorithms. It is calculated using Equation (18).

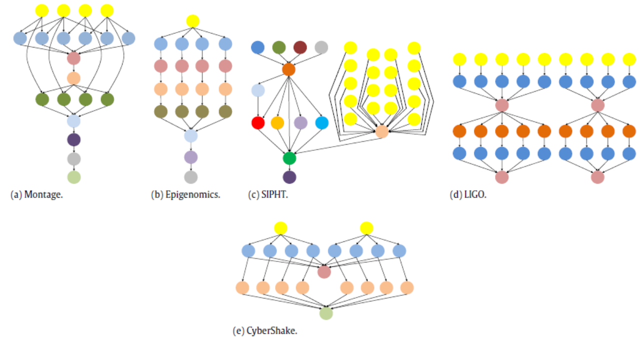

Fig. 3. The structure of realistic scientific workflows[22]

TABLE 1:HOSTS' TECHNICAL SPECIFICATIONS TABLE 1 HOSTS' TECHNICAL SPECIFICATIONS

Host ID	Number of processing Cores	Processing speed (Mips)	Ram (MB)	Hard (MB)	Bandwidth (Mbps)
1-20	8	2000000	2048000000	1000000	10000

TABLE 2 VIRTUAL MACHINE' TECHNICAL SPECIFICATIONS

Host ID	Number of processing Cores	Processing speed (Mips)	Ram (MB)	Hard (MB)	Bandwidth (Mbps)
1-15	1	1000	512	10000	1000
16-30	1	1000	512	10000	1000
31-45	2	1000	1024	20000	2000
46-60	4	2000	1024	20000	2000

TABLE 3- THE ALGORITHM PARAMETERS

Parameter	Value
Population Size (PBHO, NSGA2,SPEA2)	50
Archive Size (PBHO ,NSGA2,SPEA2)	10
Maximum Iteration (PBHO ,NSGA2,SPEA2)	20
Maximum Generation (SPEA2, NSGA2)	100
Mutation Probability (SPEA2, NSGA2)	0.5
Crossover Probability (SPEA2, NSGA2)	0.9

$$c(S_1, S_2) = \frac{\left| \forall s_2 \in S_2 ; \exists s_1 \in S_1 : S_1 \succ S_2 \right|}{\left| S_2 \right|} \tag{18}$$

C (S_1, S_2) = 1 indicates that all solutions of S_1 dominate solutions of S_2 and C (S_1, S_2) = 0 indicates that no solution of S_1 dominates a solution of S_2. Figure 4 shows coverage ratio values for the Epigenomics and the Montage workflows for three sizes: small, medium and large. A larger coverage ratio value indicates better performance.

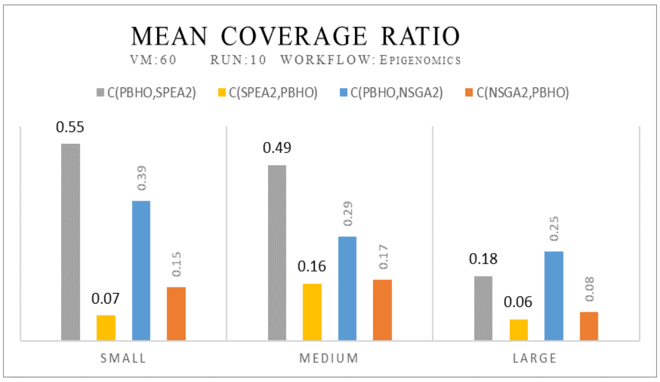

Fig. 4.a. The Coverage ratio values for Epigenomics

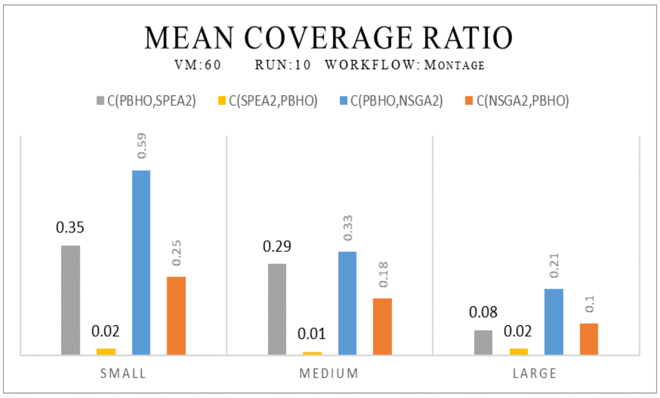

Fig. 4.b. The Coverage ratio values for Montage

According to the coverage ratio values for both the Epigenomics and the Montage workflows in the three sizes, we conclude that the values of C (PBHO, SPEA2) and C (PBHO, NSGA2) were better than the values of C (SPEA2, PBHO) and C (NSGA2, PBHO). This means that the Pareto optimal set obtained from the PBHO algorithm could dominate optimal solutions obtained from the SPEA2 and NSGA2 algorithms.

Maximum Spread

The maximum spread of the Pareto front S is the distance between solutions of the front (Equation (19)). A higher value of this index indicates that the border points have been well covered, and hence the algorithm is more efficient in selecting the optimal solution.

$$MS = \sqrt{\sum_{m=1}^{M} \left(\max_{i=1to|s|} f_m^i - \min_{i=1to|s|} f_m^i\right)^2} \qquad (19)$$

S is the Pareto front, *M* is the number of objectives and fm is the m^{th} objective function for solution *i*.

Figure 5 shows maximum spread values for the Epigenomics and Montage workflows in three sizes: small, medium and large.

According to the maximum spread values for the Epigenomics and the Montage workflows in the three sizes, we conclude that the PBHO algorithm has higher maximum spread values compared to the SPEA2 and NSGA2 algorithms. In other words, the PBHO algorithm is able to

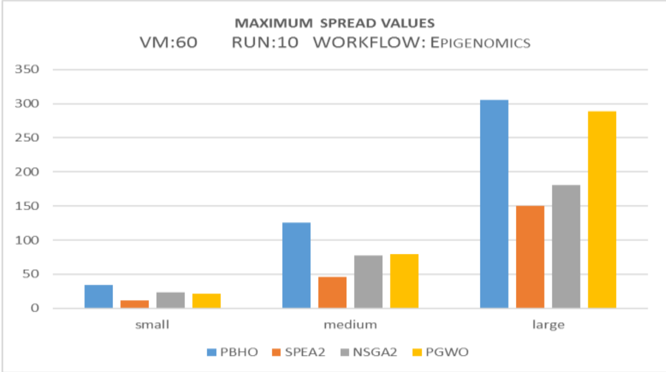

Fig. 5.a.The maximum spread values for Epigenomics

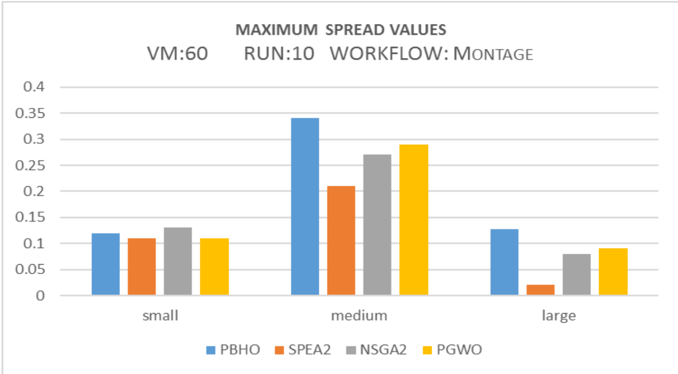

Fig. 5.b.The maximum spread values for Montage

cover more border points compared to the NSGA2 and SPEA2 algorithms. This is due to the ability of the black hole algorithm to remove solutions within a given radius of the optimal points and replace them with new random solutions. This ability enables he proposed method to avoid getting stuck at local optimal points and enables it to seek borders well.

Distance-Based Distribution (DBD)

One of the performance evaluation metrics for the Pareto optimal set is the distribution of the solutions obtained in the search space of the problem. One index of distribution is the distance-based distribution, as indicated by the value of *SP*. The distance di for the i^{th} solution of the optimal set S is equal to the lowest total difference between that solution and other solutions in line with each axis. The distance-based distribution reflects the diversity or accuracy rate in the Pareto optimal set and is obtained from Equation (20). Performance and efficiency are better when this metric has a lower value.

$$SP = \sqrt{\frac{1}{|S|-1}(d_i - \bar{d})^2} \qquad (20)$$

$$d_i = \min \sum_{m=1}^{M} \left| f_m(s_i) - f_m(S_k) \right|$$

$$S_k \in S \ and \ S_k \neq S_i$$

Figure 6 shows Distance Based Distribution (DBD) values for Epigenomics and Montage workflows in three sizes, small, medium and large.

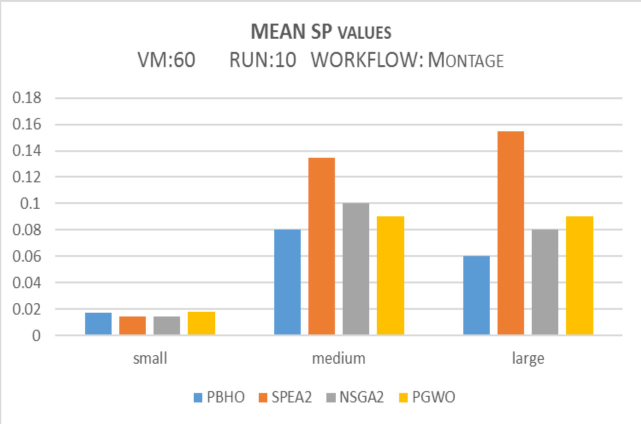

Fig. 6.a. Distance Based Distribution (DBD) Values For Montage

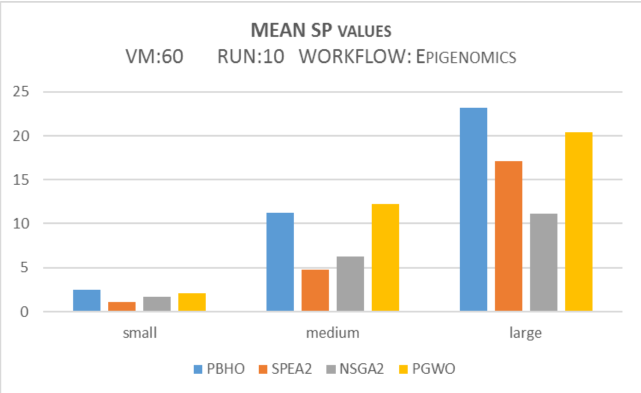

Fig. 6.b. Distance Based Distribution (DBD) Values For epegenomic

The value of SP shows the diversity and accuracy of the chosen solution, and the SP shows that fewer algorithms are more efficient. The simulation results show that the diversity rate of solutions in the PBHO algorithm for the Montage workload (unbalanced workflow) for the medium and large sizes, has more precision and diversity compared to the other algorithms, but in the Epigenomics workflow (balanced workload) SP for the SPEA2 and NSGA2 algorithms is better than for the PBHO algorithm, and therefore the diversity of solutions is greater in these algorithms.

-Evaluation Using Objectives

There is no scalar assessment method for investigating a multi-objective optimization algorithm which can sort the obtained results and choose the best answer. In such cases, the user should make proper choices from among Pareto optimal set based on preferences and requirements. We have introduced two scenarios to evaluate the results using objectives. In one scenario, different weights have been considered for the objectives of cost, resource efficiency and makespan, and in the other scenario the same weights were used for all the objectives. We obtained the optimal solution from the Pareto front according to these weights with the help of WASPAS. We repeated our experiments 10 times and calculated the average of the results for the three factors of makespan, cost and throughput for the two Epigenomics and Montage workloads in small, medium and large sizes.

A. The First Scenario: Objectives with the Same Weight:

If each objective is of equal importance, the weights of the objectives are equal, i.e., W = (0.33, 0.33, 0.33). We obtained the optimal solution from each Pareto front according to the weight vector *W* using Equation (17) and calculated the average of the values of the objective functions for those solutions. Table 4 shows the objective function values for two workloads in scenario 1 for three sizes: small, medium and large.

According to the values of the objective function for the unbalanced Montage workflow (Table 4-a) and the balanced Epigenomics workflow (Table 4-b) the following conclusions can be drawn.

For the Epigenomics workload in the large, medium and small states and the Montage workload in the medium and large states, the makespan in the black hole algorithm is lower than in the other algorithms. In the Montage workload in the small state, the makespan value in the NSGA2 algorithm is lower than in the other algorithms.

In the Epigenomics workload in large and medium states and in the Montage workload in the large state, the cost of the PBHO algorithm is lower than that of the other algorithms. In the other cases, there is little difference in the cost between the optimal value and the near-optimal states.

The efficiency for both Montage and Epigenomics workloads in large, medium and small states in the PBHO algorithm is greater than in the other algorithms.

TABLE 4.A RESULT OF PBHO, NSGA2 AND SPEA2 FOR MONTAGE WORKFLOW

Type	*Algorithm*	*Makespan*	*Cost*	*Troughput*
Small	Spea2	31040	144594	757.98
	Nsga2	27656	152681	773.7
	PBHO	31380	155961	781.9
medium	Spea2	89363	1466098	1197.77
	Nsga2	91890	1526879	1195.47
	PBHO	85864	1499458	1208.64
Large	Spea2	167317	15520129	1248.68
	Nsga2	165317	17520129	1248.58
	PBHO	154077	13995587	1288.77

TABLE 4.B: RESULT OF PBHO, NSGA2 AND SPEA2 FOR EPIGENOMICS WORKFLOW

Type	*Algorithm*	*Makespan*	*Cost*	*Troughput*
Small	*Spea2*	*119.62*	*1954*	*909*
	Nsga2	*119.56*	*1947*	*907*
	PBHO	*95.52*	*1950*	*915.72*
medium	*Spea2*	*124.92*	*4127*	*1229.6*
	Nsga2	*125.33*	*4143*	*1246*
	PBHO	*104.1*	*4003*	*1266*
Large	*Spea2*	*401.06*	*45598*	*1246.38*
	Nsga2	*381.67*	*45102*	*1246.33*
	PBHO	*337.82*	*44064*	*1256.64*

Generally, when objectives are not superior to each other, the PBHO algorithm performs better than the SPEA2 and NSGA2 algorithms and produces better optimal values for the makespan, cost and throughput objectives.

B. The Second Scenario: Objectives with Different Weights:

If cost is more important for the user than the other objectives, and efficiency has the least importance, the weights of the objectives are given by W = (cost: 0.5, makespan: 0.33, efficiency: 0.16). We obtained the optimal solution in each Pareto front according to the weight vector *W* using Equation (17) and calculated the average of the values of the objective functions for those solutions. Table 5 shows the objective function values for two workloads in scenario 2 in three sizes: small, medium and large.

According to the values of the objective function for the unbalanced Montage workflow (Table 5-a) and the balanced Epigenomics workflow (Table 5-b) the following conclusions can be drawn.

For the Epigenomics workload in the large, medium and small states and the Montage workload in the medium and large states, the makespan in the black hole algorithm is lower than in the other algorithms. In the Montage workload in the small state, the makespan value in the NSGA2 algorithm is lower than in the other algorithms.

The cost in both the Montage and Epigenomics workloads in the large, medium and small states in the PBHO algorithm is greater than in the other algorithms.

The throughput in both the Montage and Epigenomics workloads in the large, medium and small states in the PBHO algorithm is less than in the other algorithms and there is little difference between the optimal value and the near-optimal states.

As is clear from the results in Tables 5-a and 5-b, the cost for both the workloads in all sizes in the PBHO algorithm is greater than in the other algorithms. The proposed algorithm can find solutions with the lowest cost from among the total possible solutions due to the fact that the importance of cost for the user is higher than the importance of the other objectives.

7. Conclusions and Future Work

The main purpose of this paper was to present a new heuristic algorithm for solving the problem of workflow scheduling in the cloud environment that was able to optimize the user's minimum expected service quality and could also increase profitability for the service provider. We proposed a multi-objective Pareto-based black hole optimizer (PBHO) by combining the Pareto concept with the heuristic algorithm. The proposed method performed better than previous algorithms for some criteria.

We selected an optimal solution from the Pareto optimal set using the WASPAS method according to user requirements. As the simulation results show, the diversity rate of solutions in the black hole algorithm for the balanced workload was less than in the SPEA2 and NSGA2 algorithms. We intend to improve our technique and increase the precision and diversity rate of solutions for the balanced workload in future work. In addition, we plan to provide dynamic scheduling for workflow applications using this algorithm.

TABLE 5.A: RESULT OF PBHO, NSGA2 AND SPEA2 FOR MONTAGE WORKFLOW

Type	*Algorithm*	*Makespan*	*Cost*	*Troughput*
Small	Spea2	30125	144140	777.0
	Nsga2	29876	145328	777.7
	PBHO	31235	132456	761
medium	Spea2	87564	1583798	1177.19
	Nsga2	90123	1526879	1185.43
	PBHO	85349	1438698	1108.06
Large	Spea2	163125	15234875	1248.07
	Nsga2	159765	18126429	1245.45
	PBHO	153120	13336768	1238.11

TABLE 5.B: RESULT OF PBHO, NSGA2 AND SPEA2 FOR EPIGENOMICS WORKFLOW

Type	*Algorithm*	*Makespan*	*Cost*	*Troughput*
Small	Spea2	115.99	1924	911
	Nsga2	118	1984	906
	PBHO	94.12	1894	901
medium	Spea2	127.4	4234	1221.7
	Nsga2	145.4	4456	1254.2
	PBHO	108.56	4003	1201
Large	Spea2	421	46756	1246.87
	Nsga2	389.78	44984	1246.45
	PBHO	340.32	42134	1234.12

References

[1] F. Ebadifard and S. M. Babamir, "Dynamic task scheduling in cloud computing based on Naïve Bayesian classifier," Proceedings of the International Conference for Young Researchers in Informatics, Mathematics and Engineering Kaunas, Lithuania, April 28, 2017, vol. 1852, 2017.

[2] F. Ebadifard and S. M. Babamir, "A PSO-based task scheduling algorithm improved using a load-balancing technique for the cloud computing environment," Concurrency and Computation: Practice and Experience, pp. e4368-n/a, Art. no. e4368, 2017.

[3] A. Hatamlou, "Black hole: A new heuristic optimization approach for data clustering," Information Sciences, vol. 222, pp. 175-184, 2/10/ 2013.

[4] L. Liu and M. Zhang, "Multi-objective Optimization Model with AHP Decision-making for Cloud Service Composition," KSII Transactions on Internet &Information Systems, vol. 9, no. 9, 2015.

[5] W. Chen and E. Deelman, "Workflowsim: A toolkit for simulating scientific workflows in distributed environments," IEEE International Conference on EScience (e-Science), pp. 1-8, 2012.

[6] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," Software: Practice and Experience, vol. 41, no. 1, pp. 23-50, 2011.

[7] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm,"Technical Report 103, Gloriastrasse 35, CH-8092 Zurich, Switzerland, 2001.

[8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182-197, 2002.

[9] D. R. Araújo and et. al., "A performance comparison of multi-objective optimization evolutionary algorithms for all-optical networks design," IEEE Symposium on Computational Intelligence in Multicriteria Decision-Making, pp. 89-96, 2011.

[10] S. Pandey, L. Wu, S. M. Guru, and R. Buyya, "A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments," in Proc. IEEE Int. Conf. Adv. Inform. Netw. Appl., 2010, pp. 400–407.

[11] J. Yu and R. Buyya, "A budget constrained scheduling of workflow applications on utility grids using genetic algorithms," Workflows in Support of Large-Scale Science, pp. 1-10, 2006.

[12] B. Keshanchi, A. Souri, and N. J. Navimipour, "An improved genetic algorithm for task scheduling in the cloud environments using the priority queues: Formal verification, simulation, and statistical testing," Journal of Systems and Software, vol. 124, pp. 1-21, 2017/02/01/ 2017.

[13] S. Bilgaiyan, S. Sagnika, and M. Das, "A Multi-objective Cat Swarm Optimization Algorithm for Workflow Scheduling in Cloud Computing Environment," in Intelligent Computing, Communication and Devices: Proceedings of ICCD 2014, Volume 1, 2015, pp. 73-84.

[14] O. Udomkasemsub, X. Li, and T. Achalakul, "A multiple-objective workflow scheduling framework for cloud data analytics," in 2012 Ninth International Conference on Computer Science and Software Engineering (JCSSE), 2012, pp. 391-398.

[15] Z. Wu, Z. Ni, L. Gu, and X. Liu, "A Revised Discrete Particle Swarm Optimization for Cloud Workflow Scheduling," in 2010 International Conference on Computational Intelligence and Security, 2010, pp. 184-188.

[16] S. Yassa, R. Chelouah, H. Kadima, and B. Granado, "Multi-Objective Approach for Energy-Aware Workflow Scheduling in Cloud Computing Environments," The Scientific World Journal, vol. 2013, p. 13, 2013.

[17] P. Kaur and S. Mehta, "Resource provisioning and work flow scheduling in clouds using augmented Shuffled Frog Leaping Algorithm," Journal of Parallel and Distributed Computing, vol. 101, pp. 41-50, 2017/03/01/ 2017.

[18] A. Khalili and S. M. Babamir, " Optimal Scheduling workflows in Cloud Computing Environment Using Pareto based Grey Wolf Optimizer" In Concurrency and Computation: Practice and Experience, http://mc.manuscriptcentral.com/cpe, 2016.

[19] B. Johannes, "Scheduling parallel jobs to minimize the makespan", Journal of Scheduling, Springer, 9:433-452, 2006.

[20] E. Zavadskas, Z. Turskis, J. Antucheviciene, and A. Zakarevičius, Optimization of Weighted Aggregated Sum Product Assessment. 2012.

[21] S. Bharathi, A. Chervenak, E. Deelman, G. Mehta, M. H. Su, and K. Vahi, "Characterization of scientific workflows," in *2008 Third Workshop on Workflows in Support of Large-Scale Science*, 2008, pp. 1-10.

[22] Pegasus Workflow Management System, available at: http://pegasus.isi.edu/workflow_gallery/index.php (Dec. 2015).last visited : 2017/11/17.

[23] T. Okabe, Y. Jin, and B. Sendhoff, "A critical survey of performance indices for multi-objective optimisation," in Evolutionary Computation, 2003. CEC '03. The 2003 Congress on, 2003, pp. 878-885 Vol.2.

[24] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," IEEE Transactions on Evolutionary Computation, vol. 3, pp. 257-271, 1999.

[25] E. Zitzler, J. Knowles, and L. Thiele, "Quality Assessment of Pareto Set Approximations," in Multiobjective Optimization: Interactive and Evolutionary Approaches, J. Branke, K. Deb, K. Miettinen, and R. Słowiński, Eds., ed Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 373-404.

Fatemeh Ebadifard received her B.S. degree in Software Engineering from Qom university, Iran, in 2010 and her M.Sc. degree in network Engineering from University of Science & Technology, Tehran, in 2014 and Currently she is a PhD student in software Engineering at Kashan University. Her research interests include Cloud computing, Task scheduling and network programming.

Seyed Morteza Babamir received BS degree in Software Engineering from Ferdowsi University of Meshhad and MS and PhD degrees in Software Engineering from Tarbiat Modares University in 2002 and 2007, respectively. Now, he is an associate professor with Department of Computer Engineering at University of Kashan, Kashan, Iran. He authored one book in Software Testing, four book chapters, 40 journal papers and more than 50 international and internal conference papers.