Journal of Algorithms and Computation

journal homepage: http://jac.ut.ac.ir

A security aware workflow scheduling in hybrid cloud based on PSO algorithm

Maedeh Mehravaran*[1], Fazlollah Adibnia†[2] and Mohammad-Reza Pajoohan‡[3]

[1,2,3]Department of Computer Engineering, Yazd University, Yazd, Iran.

ABSTRACT

In real world, organization's requirements for high performance resources and high capacity storage devices encourage them to use resources in public clouds. While private cloud provides security and low cost for scheduling workflow, public clouds provide a higher scale, potentially exposed to the risk of data and computation breach, and need to pay the costs. Task scheduling, therefore, is one of the most important problems in cloud computing.

In this paper, a new scheduling method is proposed for workflow applications in hybrid cloud considering security. Sensitivity of tasks has been considered in recent works; we, however, consider security requirement for data and security strength for resources. The proposed scheduling method is implemented in Particle Swarm Optimization (PSO) algorithm. Our proposed

Keyword: Cloud computing; Task scheduling; Security requirements; Resource; PSO.

AMS subject Classification: 05C78.

ARTICLE INFO

Article history:
Received 11, october, 2019
Received in revised form 05, May 2020
Accepted 21 May 2020
Available online 01, June 2020

*m.mehravaran@stu.yazd.ac.ir
†Corresponding author: F. Adibnia. Email: fadib@yazd.ac.ir
‡pajoohan@yazd.ac.ir

1 Abstract continued

algorithm considers minimizing security distance, that is maximizing similarity of security between data and resources. It, meanwhile, follows time and budget constraints. Through analysis of experimental results, it is shown that the proposed algorithm has selected resources with the most security similarity while user constraints are satisfied.

2 Introduction

Recently, many researchers have considered the benefits of scheduling tasks on cloud computing. Clouds provide the virtual resources according to the requirements of the application. Therefore, the proper mechanism for scheduling applications (workflow) on efficient resources is required.
Workflow is described by a Directed Acyclic Graph (DAG) in which each task is represented by a node, and each data between tasks is represented by a directed edge. If there is an edge between the two nodes, the first task execution must be done before the second task, because of the requiring data from first task. The amount of data being transferred between two tasks is determined by the weight of the edge.
Mapping each task to a suitable resource to satisfy some performance criterion is Workflow scheduling.
There is no known algorithm for this problem to find the best solution in polynomial time. So many researches have been focused on the workflow scheduling in the cloud computing which obtain the approximated solutions by heuristics and meta-heuristics algorithms [14].
Hybrid cloud includes a private cloud with limited, secure, low cost resources and public clouds with a higher scale, potentially exposed to the risk of data and computation breach, and need to pay the costs. So When private resources are not sufficient to satisfy the requirements of users, the public cloud infrastructure can support it[16].
Several challenges can be faced in hybrid cloud [7] such as security consideration, privacy preservation, virtualization machines, flexibility and energy consumption, dynamic allocation and allocation with preemptive resources.
Many papers focused on the workflow scheduling schemes in the cloud computing environment based on the type of the algorithm. Various objectives and properties are compared on these schemes. One of the objective which have been addressed in recent workflow scheduling schemes is security [13]. Considering security attacks against schedulers and other cloud components such as the VMs signify the subject.
In previous works security and privacy of tasks were considered. For example, they defined two sets of tasks [5,10–12], sensitive tasks, which are related to the private information of the organization, and insensitive tasks, which are not sensitive and can be scheduled on the public cloud resources.
In this paper, we consider security for tasks and data interaction between tasks and also

security strength for resources. The proposed algorithm use Particle Swarm Optimization (PSO) which is one of the metaheuristic algorithms.
The remainder of this paper is organized as follows: Section 2 gives an overview of existing approaches to scheduling workflows. In Section 3, we define the workflow scheduling problem and describe the basis of our algorithms which considers security. Section 4 shows the evaluation of the performance of our algorithm. Section 5 is the conclusions of the paper with future directions.

3 Related work

Liu and et al. [11] proposed a "security and cost aware scheduling "(SCAS) algorithm for scientific workflows in clouds. This proposed algorithm is based on the particle swarm optimization (PSO). The objective of this strategy is to minimize the total workflow execution cost while meeting the risk rate constraints. Risk constraint has three modes. In Secure mode, schedule tasks only on high secure computing units. In risky mode, it's free to schedule tasks on any available resources so take all possible risks. C-risky mode, schedule tasks on resources which have at most C risk, where C is parameter between 0 and 1 [11].
Chen et al. [5] proposed scheduling approach with selective tasks duplication which have two important phases. First, selecting the tasks which should be duplicated. Second, schedule this tasks on the idle time slots of resources. They evaluate proposed approach using both real workflows and randomly generated ones. The result shows the minimization of makespan and cost because of eliminating data encryptions for intermediate data.
"A security-aware intermediate data placement strategy "is proposed by Liu et al. [12]. They build a security overhead model to measure the sensitive data security overheads. Then, place the intermediate data for the scientific workflows. The data placement problem solved by using Ant Colony Optimization (ACO).
Li et al. [10] proposed a model which considers security services on task execution process. This is shown in Figure 1. They assumed that task t_i is the successor of task t_{i-1} and task t_{i-1} will be submitted data to task t_i. Authentication services are deployed to data transferring from successor tasks [10]. The authenticated algorithms depicted in Table 1 [5]. Since integrity services ensure that no one can modify the datasets without being detected, they are used to cope with threats of alteration. Integrity algorithms depicted in Table 2 [5]. Finally, confidentiality services used to ensure that data will be not available to unauthorized persons. Table 3 shows the confidentiality algorithms [10].
The security strength for this algorithms, depicted in Tables 1,2,3 shows by coefficient between zero and one. Zero shows minimum security strength and one shows the max security strength. whoever if the algorithm have more strength, more complicated computing needs.
Abrishami et al. [2] proposed "Workflow scheduling on Hybrid Cloud to maintain Data Privacy (WHPD)" algorithm to solve scheduling problems. This

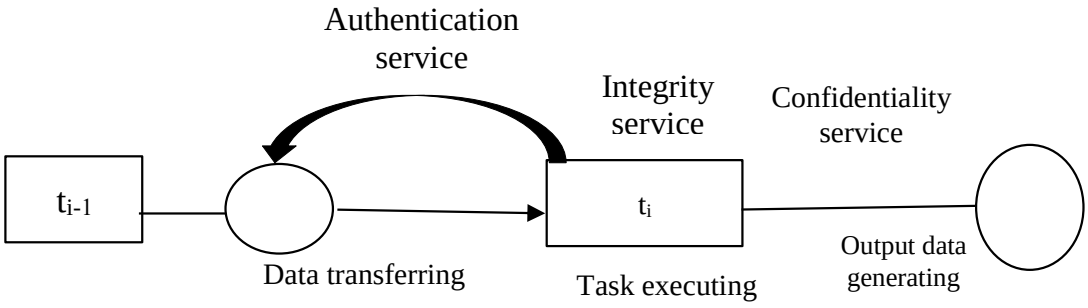

Figure 1: Security service [10]

Authentication Algorithms	**Strength Security**	**Speed(Mb/s)**
CBC-MAC-AES	0.9	163
HMAC-SHA-1	0.6	148
HMAC-MD5	0.3	90

Table 1: Hash Function [5]

Hash Function	**Strength Security**	**Speed(Mb/s)**
TIGER	1.00	48.03
RIFDMD-160	0.77	71.27
SHA-1	0.63	80.67
RIFDMD-128	0.36	86.97
MD5	0.26	138.12

Table 2: Hash Function [5]

Encryption Algorithms	**Strength Security**	**Speed(Mb/s)**
IDEA	1.00	17.34
DES	0.9	18.21
Rijndael	0.64	39.88
RC4	0.36	39.96

Table 3: Confidentiality algorithm [10]

	τ_{t1}	τ_{t2}	τ_{t3}
τ_{s1}	1	0	0
τ_{s2}	1	1	0
τ_{s3}	1	1	1

Table 4: Privacy map allocation [15]

algorithm uses sensitivity for tasks and schedule sensitive tasks on private cloud. They uses heuristic algorithms to solve the problem of scheduling workflows on hybrid clouds. This algorithm assigned levels to tasks and then distributes deadline among levels. The objective of this paper is to minimize budget while sub-deadline limitation is satisfied.
Bittencourt and Madeira [4] proposed the Hybrid Cloud Optimized Cost scheduling algorithm (HCOC) in order to schedule a workflow on multicore systems on the hybrid cloud. This method minimized cost while the user-defined deadline satisfied. This method starts scheduling tasks on private resources while the user-defined deadline violates, new resources are leased from the public cloud and the algorithm reschedules the workflow.
Sharif et al. [15]presented an algorithm that preserves privacy in workflow scheduling, while considering users' deadlines and cost. Three levels of privacy is considered for workflow tasks as $\Gamma t = (\tau_{t1}, \tau_{t2}, \tau_{t3})$. τ_{t1} is the first level that tasks can be deployed on both public and private resources. τ_{t2} is the second level which tasks can be scheduled on private instances and some selected public resources. The third level, τ_{t3}, tasks that executed only on private resources. They also assign privacy to the resources noted as $\Gamma s = (\tau_{s1}, \tau_{s2}, \tau_{s3})$. τ_{s1} denoted as public instances, τ_{s2} and τ_{s3} have higer security, selected public and private instances, respectively. Table 4. depicted maps each task to permissible resources.
"A Novel Deadline-Constrained Scheduling to Preserve Data Privacy in Hybrid Cloud" [1], was proposed by Abrishami et al. In this algorithm, a new scheduling model is proposed for workflow scheduling in hybrid cloud architecture in order to maintain privacy of tasks. The proposed method schedules the sensitive tasks on the private cloud, which is secure and under control of the organization, and the other non-sensitive tasks on hybrid cloud. In this model, the scheduler tries to find resources with minimum cost [1].
Sooezi et al. [17]proposed scheduling algorithm in multi-cloud environment. In this proposed algorithm, they partitioned the workflow into sub-graphs which have the largest data transmission. Then each sub-graph schedules on the selective cloud. Since the data transfer rate between the instances of a cloud is very high, so the makespan and cost of workflow is minimized.
The scheduling algorithm proposed by Fernandez et al. [6] made use of metaheuristic algorithms to minimize the energy consumption of task scheduling on resources as well as the amount of time the entire task requires.
Our algorithm focus on security for data and resource which is discuss in the next Section.

4 Problem Definition

Before we formulate the scheduling problem for workflow applications in Hybrid cloud computing environments, the environment components need to be introduced. The security constraint model is also presented for the workflow scheduling problem. Lastly, the performance metrics are expressed.

4.1 Environment description

Different terminologies are used in literatures. Here we define the key terms employed in formulating the scheduling problem.

- Resource (Virtual Machine) is a part of computational machine with their computational capacity including the number of CPUs, amount of memory, storage space and other specializations. A set of m VMs in the computing environment is denoted as $(VM_1, VM_2, \ldots, VM_m)$.

- Workflows are the most widely used models for representing scientific computations [14]. A Directed Acyclic Graph (DAG) is represented a workflow. Using a DAG, a workflow is defined as a graph $G = (T, E)$ where $T = (t_0, t_1, \ldots, t_n)$ is denoted a set of tasks represented by vertices and $E = (e_{i,j}|t_i, t_j \in T)$ is a set of directed edges denoting data transfer between tasks. An edge $e_{i,j} \in E$ represents a directed arc between two tasks t_i and t_j where task t_j can start only after completing the execution of task t_i with all data received from t_i . In other word, task t_i is the parent of task t_j, and task t_j is the successor or child of task t_i. Each task can have one or more parents. Task t_i cannot start until all parents have been completed.

- Hybrid cloud is an infrastructure for scheduling tasks. The combination of public and private clouds is known as hybrid cloud. The private cloud provides low cost and privacy for workflow s execution because they are on the control of the users. However, an organization's requirements to high performance resources and high capacity storage devices encourage them to utilize public clouds. Public cloud leases resources or services from providers. So that, this model is potentially exposed to the risk of data and computation breach and is less secure in comparison to a pure private cloud environment.

- Scheduling problem: A schedule is the mapping of the tasks to specific time intervals on specific virtual machines which various constraints consider. In this article we consider security constraint model for our scheduling problem.

To explain definitions and our proposed model, we use following notations described in Table 5.

Definition	Notation
The ith task of workflow	t_i
The entry task of workflow	t_{entry}
The exit task of workflow	t_{exit}
The size of data transferred from task t_i to t_j	$d_{i,j}$
The size of bandwith between resource i and j	$bw_{i,j}$
The communication time of task t_i and t_j	$c_{i,j}$
The immediate predeccesor of task t_i	Pred(t_i)
The immediate successor of task t_i	Succ(t_i)
The jth instance(resource)	vm_j
The Cost per hour of vm_j	Cost(vm_j)
The rank of task t_i on Eq.2	Rank(t_i)
The Finished time of t_i	FT(t_i)
The execution time of task t_i on vm_j	$W(t_i, vm_j)$
The execution time of task t_i on the fastest resource	$W(t_i)$
The Earliest finish time of task t_i	EFT(t_i)
The Earliest start time of task t_i	EST(t_i)
The cost of executing t_i on vm_j	TaskCost (t_i, vm_j)
The Security strength of resource vm_i on service $l\epsilon(a, c, i)$	$SS_{l\epsilon(c,i,a)}(vm_i)$
The Overhead of security strength on service $l\epsilon(a, c, i)$	$SO_{l\epsilon(c,i,a)}(vm_i)$
The Security requirment of data transfer between task t_i and t_j on service $l\epsilon(a, c, i)$	$SR_{l_\epsilon(c,i,a)}(t_i, t_j)$
The Security distance of task t_i and resource vm_j	∂_{i,vm_j}
Indicate authentication service	A
Indicate integrity service	I
Indicate confidentiality service	C
Indicate acceptable min distance	α
Indicate acceptable max distance	W

Table 5: Definitions of notation.

	Processing capacity(MIPS)	Cost per hour	Security strength for confidentially	Security strength for integrity	Security strength for authentication
1. 4.2xlarge	8800	0.035$	0.2	0.2	0.5
2. 4.xlarge	17600	0.2$	0.2	0.3	0.3
3. m4.large	17600	0.1$	0.5	0.3	0.5

Table 6: Resources with security parameters

4.2 Security constraint model

We present a security constraint model to explain our proposed algorithm next. Related research can be found in [5,10–12].

Definition 3.1[Security strength for VMs] Security services for VMs can be defined by three parameters. These parameter are maximum security strength of the resource to satisfy the security services, i.e. confidentially, integrity and authentication. We refer to these security services as security strength aspects in the rest of this paper. The parameter values are calculated based on the overhead of algorithms used in VMs. In other words, these parameters are normalized of the inverted overhead values. As these parameters are division of two values with the same unit, they have no unit. Despite of previous works considering security for data center, we consider security strength for each resources. Related algorithms used for security services are depicted in Tables 1, 2 and 3 respectively. The specification of our proposed VMs is shown in table 6. As one can see, in addition to the capacity and cost, the security strength aspects are added to each resource. Resources in private cloud have (1, 1, 1) security strength aspects shows maximum protection.

Definition 3.2[Security requirements]

The security requirements of data are represented by three coefficients representing the requirement of data to security services that is confidentiality, integrity and authentication. We refer to these coefficients as security requirements aspects. A graph shown in Figure 2 depicts security requirement aspects of data transmitted between tasks.

As an example, $SR_{T12} = (0.14, 0.36, 0.52)$ represents security requirement of data between task 1 and task 2 which implies least security requirements to confidentiality, integrity and authentication services. If security requirement aspects of data is (1, 1, 1) then this data must be scheduled on private resources.

By defining strength security aspects for resources and security requirement aspects for data, a task in the workflow tries to be executed on resources with the most similar security. Although scheduling tasks on resources with higher security is usually preferred, it is violated in some circumstances. In other words if users apply better security services, it will incur longer processing time which also result in more monetary cost and larger makespan. Hence, we should select the resource with the most similar security to task [10].

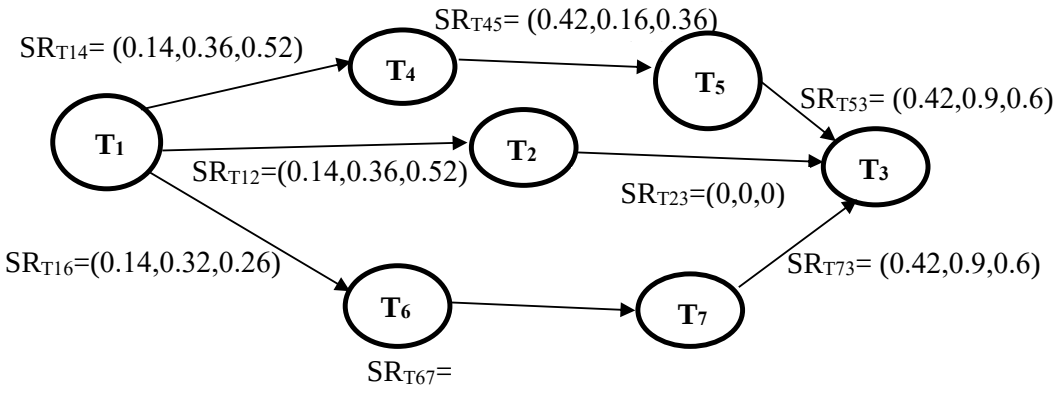

Figure 2: Security requirement of data on security services

Li et al. [10] defined risk probability for scheduling tasks. The risk probability is a function of security levels and the distribution of risk rate for any time interval which follows a Poisson probability distribution calculating by Eq. (1). In this Equation p indicated security risk, sr shows tasks security requirements and sl is for security strength. $\{a, g, c\}$ indicates authentication, integrity and confidentially service respectively.
Poisson has complex calculations and does not correctly answer for some examples which is expressed in Example 1.

$$p(t_i, sl_i^l) = 1 - \exp(-\gamma(sr_i^l - sl_i^l)), l \in (a, g, c) \quad (1)$$
$$p(t_i) = 1 - \prod_{l \in (a,g,c)} p(t_i, sl_i^l)$$

example 1. Drawback of Poisson
Supposed two resources with security strength $SS_1 = (0.14, 0.36, 0.52)$ and $SS_2 = (0.3, 0.36, 0.36)$ and data with security requirements $SR = (0.14, 0.36, 0.52)$. Based on Eq. (1), Poisson function calculates the same values for risk probability of data and resources. It means that the security risk for scheduling data on SS_1 and SS_2 is the same value -one-. As we can see, SS_1 is completely satisfied the security requirement aspects of data. Meanwhile, SS_2 have different security strength aspects with security requirements of data.
Definition 3.3[Security Distance] Security distance is the distance between security requirement aspects of data (input or output) and security strength aspects of resources. It can be used to find the most similar security of data and resource. We use *Manhattan* distance for calculating security distance.
The Manhattan distance between two items is the sum of the differences of their corresponding components. Security distance (∂) of task i and resource j is defined based on Eq. (2). This is the fitness function of our proposed model. The less the ∂ the security

similarity of the task and resources is more.

$$\begin{aligned}\partial_{t_i,vm_j} &= \sum_{t_{p\in Pred(t_i)}} |SS(vm_j) - SR(t_p,t_i)| \\ &+ \sum_{t_{s\in Succ(t_i)}} |SS(vm_j) - SR(t_i,t_s)||SS(vm_j) - SR(t_p,t_i)| \\ &= \beta * |(SS_c(vm_j) - SR_c(t_p,t_i)| \\ &+ \beta * |(SS_a(vm_j) - SR_a(t_p,t_i))| \\ &+ \beta * |(SS_i(vm_j) - SR_i(t_p,t_i)|\end{aligned} \tag{2}$$

For computing the security distance between task and resource, we consider the security of input (output) data of task and that of resources. This is because when the task to be scheduled on resource, it should decode the input data to execute the task. So, the input data is accessible by resource, and the security requirement aspects of data should be similar to the security strength aspects of resource as more as possible. The output data of task is accessible to resource as well. Therefore, security similarity between resource and output data should be considered. In computing security distance in Eq. (2), β is considered. To the definition of β, we should explain two definitions first.
Definition 3.4[Maximum security factor (α)] If the security strength of resource is less than the security requirements of data, the difference must be less than a user defined threshold acceptable for him which is defined as maximum security factor represented by a in Eq. (3). In secure scheduling, we prefer to select resources with security strength higher than security requirement of tasks. This condition, however, is not always possible. So, we may be forced to select resources with security strength less than the security requirements. But the difference should not be more than the maximum security factor.
Definition 3.5[Source depletion factor (w)]
Source depletion factor implies the depletion when one hesitates between selecting a resource with security strength higher than security requirement in contrast to selecting resource with security strength less than security requirement. Recall that in Eq.(2), the difference is multiplied by w when security requirement is larger than security strength. β in Eq.(2) has two possible values represented in Eq. (3). β is 1, if the difference between security strength and security requirement is positive. It means that the security strength of resource is more than the security requirement of data which is good. However, if the security strength of resource is less than the security requirement of data, then the value w is assigned to β. w denotes source depletion factor which is defined in Def. 5. The difference of security strength and security requirement should be less than maximum security factor (a) which is defined as follows.

$$\beta = \begin{cases} w & if(SS_c(vm_j) - SR_c(t_p,t_i) < a \\ 1 & else \end{cases} \tag{3}$$

example 2. Calculating security distance between task and resource

Supposed the data security requirement is $SR = (0.4, 0.3, 0.3)$ and two resources with security strength exist, i.e. $SS_1 = (0.6, 0.3, 0.3)$ and $SS_2 = (0.3, 0.3, 0.3)$. We supposed to schedule task 2 on Figure 2, on resource with security strength $(0.3, 0.36, 0.36)$. For calculating security distance of task 2 and resource, we should consider the input and output data of task. Security requirement of these data are $SR_{T23} = (0, 0, 0), SR_{T12} = (0.14, 0.36, 0.52)$. If source depletion factor sets to 2, based on Eq. 2 security distance calculates as follows.

$$\begin{aligned}
\partial_{t_2,vm_2} &= \beta * |SS(vm_2) - SR(t_1, t_2)| + \beta * |SS(vm_2) - SR(t_2, t_3)| \\
|SS(vm_2) - SR(t_2, t_3)| &= (0.3 - 0) + (0.36 - 0) + (0.36 - 0) = 1.02 \\
|SS(vm_2) - SR(t_1, t_2)| &= (0.3 - 0.14) + (0.36 - 0.36) + 2 * (|0.36 - 0.52|) = 0.48 \\
\partial_{t_2,vm_2} &= 1.02 + 0.48 = 1.5
\end{aligned}$$

4.3 Particle swarm heuristics for scheduling problems

Since task scheduling is a one of the problems that have no polynomial solutions, so the Particle Swarm Optimization (PSO), the Meta-heuristic optimization technique, is used in this article. We use PSO as it have a faster convergence rate and fewer primitive mathematical operators than other Meta-heuristic algorithms and also this algorithm is less dependent on parameter fine-tuning.

Particle Swarm Optimization (PSO) is a swarm-based intelligence algorithm [9] influenced by the social behavior of animals such as a flock of birds finding a food source. A particle in PSO is a bird flying through a search space that usually are initialized randomly. The movement of each particle is coordinated by a velocity. Each particle position in each generation is influenced by its best position "pbest" and the position of the best particle in a problem space "gbest". The particle performance in each generation is measured by a fitness value and the velocity and the position of particles will be updated as in Eq. (4 and 5), respectively.

$$population[i] = velocity[i] + population[i] \tag{4}$$

$$velocity[i] = velocity[i] + c1 * r1 * (pbest[i] - population[i]) + c2 * r2 * (gbest - population[i]) \tag{5}$$

The brief description of Eq. ((6), (7) where :

velocity[i]	velocity of particle i
pbest[i]	best position of particle i
gbest	position of best particle in a population
population[i]	current position of particle i
c1, c2	acceleration coefficients
r1, r2	random number between 0 and 1

Our proposed algorithm, PSO Workflow Scheduling algorithm considering Security (PSOWFS) is described as follows.

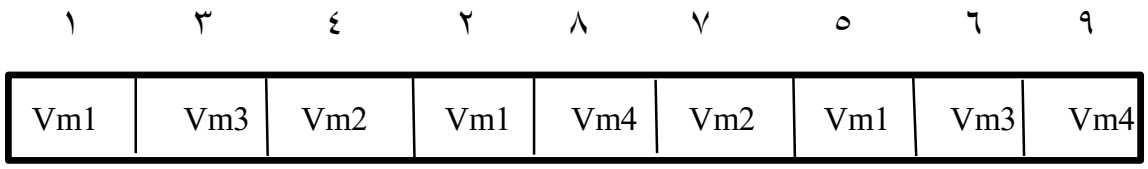

Figure 3: Sample particle

In this problem, each particle is one of the solutions and the dimension of the particles is the number of tasks in a workflow. The value assigned to each dimension of particles are the computing resources indices. Thus the particle represents a mapping of resource to a task. The order of tasks in particles are based on priority. Assigning priority to the tasks of the workflow is done using the upward ranking [3]. Eq. (8) is the recursive function of computing ranks. $w(i,r)$ refers to average execution time of task i on resources.

$$rank(t_i) = \begin{cases} w(i,r) & t_i = t_{exit} \\ w(i,r) + max_{t_{p \in childsof(t_i)}(rank(t_i)+c_{i,p})} & otherwise \end{cases} \quad (6)$$

As an example, suppose the workflow with 9 tasks and 4 resources. Each particle is 9-D because of 9 tasks and the content of each dimension of the particles is the computing resource assigned to that task. So as we have 4 VMs (Virtual Machine), the sample particle could be represented as Figure 6.
The pseudo code for this algorithm is described as tabel 7, and the following paragraphs provide the steps in the PSO in detail.
There are parameters in PSO that are dependent to the problem. One parameter is the number of the population which is in the range 20-50. In our work we set the number of population to 20. Other parameters are the dimension of the particles and the range in which they are allowed to move, which is modeled in paragraph before. The algorithm starts with random initialization of particle's position and velocity (line 4-8). In initialization, the known algorithm HEFT (Heterogeneous Earliest Finish Time) is used. So we benefit this algorithm.
From line 10, the iteration of algorithm starts. The evaluation of each particle is performed by the fitness function given in Eq. (2). The particles calculate their velocity using Eq. (4) and update their position according to Eq. (5). The velocity and position updates are liable to cause particles to exceed the boundaries of the feasible conditions of the problem. This algorithm modified the values to keep the particle within the search space. So, when a new position goes beyond its boundaries, then the value of its corresponding boundary (either the lower or the upper boundary) alternates the calculated value. The evaluation is carried out until the specified number of iterations (user-specified stopping number of iteration).
In line 13, we check user-specified constraint. Each particle violates the constraint, the particle should be recreated. After all, the best answer is in gbest.

Algorithm 1. (The security workflow scheduling in hybrid cloud on PSO)

BEGIN
1.Set the number of particles to p and the dimension of the particles to n;// $p \in [20 - 50]$
2. Set the gbest and pbest to zero.
3.//Initialize the population with random particles and random velocities;
4. for each particle i=1 to p
5. Randomly create particle population[i] and velocity[i]
6 pbest[i]= population[i] ;
7. gbest= the best solution on current population
8. end for
9. $t = 0$;
10. while (t <number of iteration)
11. for each particle $i = 1$ to p
12. Update the velocity and position of particle on Eq.(6,7)
13. Check constraint function()
14. Calculate fitness function for each particle on Eq.(2);
15. If (fitness function(current particle i) is better than fitness function(pbest[i]))
16. pbest[i]= current particle;
17. end for
18. calculate the gbest;
19. $t = t + 1$;
20. end while
END

Table 7: The pseudo code of proposed algorithm

4.3.1 Fitness Function

To calculate the fitness function, security distance, the sum of security distance of all data transferred between tasks and resources should be obtained. The lower the security distance, the more the similarity in security. The objective function of our problem summarized as follows:

$$Min\partial = \sum_{t_i \in allTasks} \partial_i$$

4.3.2 Constraint Function

In this algorithm, time and budget constraint are considered. It means that the entire time of the execution of all tasks must be under the user-defined deadline and the constraint budget also must be under the user-defined MaxCost. The constraint is devised in Eq. (7).

$$\begin{aligned} Subjectto: makespan &< Deadline \\ TotalCost &< MaxCost \\ TaskCost(t_i, vm_j) &= W(t_i, vm_j) * Cost(vm_j) \\ TotalCost &= \sum_{t_{i \in workflow}} TaskCost(t_i, vm_j) \end{aligned} \tag{7}$$

To calculate makespan, we obtain the finish time of each task and the maximum is the response. The finish time of task is calculated in Eq. (8).

$$EFT(t_i, vm_j) = EST(t_i) + W(t_i, vm_j) + SC(t_i, vm_j) \tag{8}$$

The earliest finish time $EFT(t_i, vm_j)$ is computed in terms of the total processing time and the start time of the task. The processing time consists of the execution time $W(t_i, vm_j)$ and security overhead computing $SC(t_i, vm_j)$ which is devised next.
For each unscheduled task ti, EST(ti) is defined as its earliest start time. This is computed as follows. The first term is for maximum EFT of the parents. It means that the task execution should wait until the parents execution finished. The second term is represented data time transfer (DT) which is needed for this task. Note that the transfer time between two tasks executed on the same VM is zero.

$$EST(t_i) = \max_{t_{p \in parrentsof(t_i)}} (EFT(t_p) + DT_p, i))$$

$$DT_{p,i} = \begin{cases} \frac{d_{p,i}}{bw_{n,m}} & \text{if } t_p \text{ on } vm_n \text{ and } t_i \text{ on } vm_m \\ 0 & \text{if the resource is the same} \end{cases}$$

Security services introduce overheads to the existing computing systems. For confidentiality, integrity and authentication services, the amount of security overhead

computing mainly depends on the service overhead and the size of data. Hence, the security overhead computing of these services are denoted as multiplying the data size and security overhead (SO). Calculating security overhead on Eq. (9), have two terms. First term calculates overheads for input data which needed algorithms for decoding and the second term calculates overheads for output data which needed algorithms for encrypting data.

$$SC_l(t_i, vm_j) = \sum_{t_p \in parrentsof(t_i)} d_{p,i} * SO_l(vm_i) + \sum_{t_p \in childsof(t_i)} d_{i,p} * SO_l(vm_i) \quad l \in \{c, i, a)$$
$$SC(t_i, vm_j) = \sum_{l \in (c,i,a)} SC_l(t_i, vm_j) \tag{9}$$

After all, the consraint model is explained and the performance evaluation on sample graphs are expressed in the next section.

5 Performance Evaluation

In this section, we evaluate the performance of the proposed algorithm, task scheduling in hybrid cloud considering security. This algorithm implemented by PSO, used the workflowSim[1] framework to simulate a cloud environment. The time complexity of this algorithm is $O(M * N^2)$, where N is the number of tasks and M is the size of population.

The proposed algorithm is compared with three algorithms, HEFT, RANDOM and DHEFT, which were simulated in workflowSim by adding security parameters to them. In HEFT, the earliest completion time of all tasks on all VMs is calculated according to tasks priority. The VM that finishes the task first will be the best candidate for execution. DHEFT schedules task on VMs so that the makespan is below the user-defined deadline. Finally, based on RANDOM policy, tasks are randomly picked from the ready list and scheduled on random VMs.

In order to select a proper deadline for each workflow, we calculate the fastest execution time based on HEFT algorithm. As we know, the makespan increases when the user-defined deadline increases. It is clear that increasing the deadline causes to the resources with different properties be selected. So, based on the tasks or resources security aspects, we free to select the best resources according to the defined constraints. In other words, if the deadline does not violated, we can select the resources with lower speeds, higher similar security resulting to the lower cost and security difference.

The other parameters used in PSO algorithm are as follows.

we define $c1 = 2.05, c2 = 2.05$ and the number of particles and iterations are set to 20 and 1000 respectively, as in related works[11].

[1] https://github.com/WorkflowSim

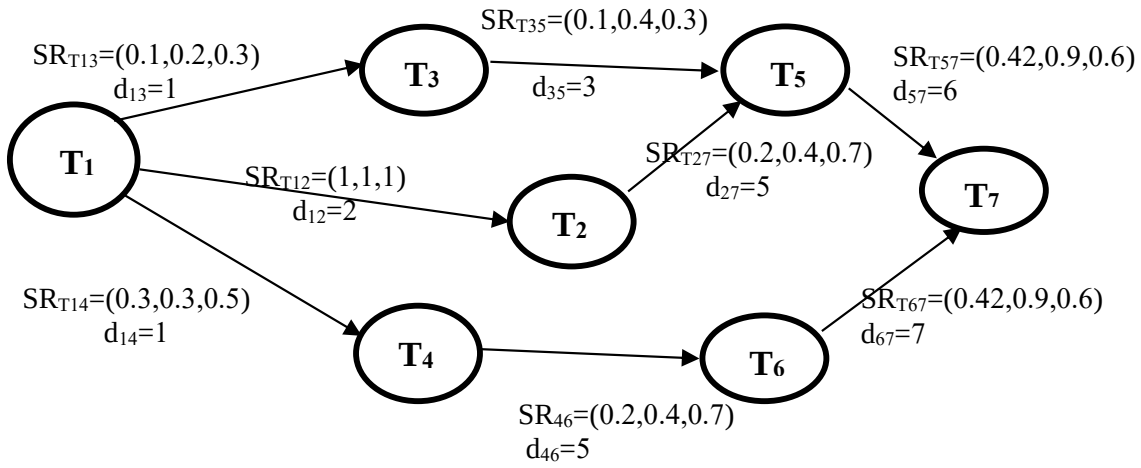

Figure 4: Sample workflow

The value of maximum security distance (a) is defined by the user. This parameter should satisfy its constraint. That is if the minimum difference of the security requirement and security strength aspects are more than (a), it is not feasible.
The source depletion factor (w) could set to 1, 2, 3. We can show by a counterexample that 1 is not suitable. Consider two resources with the security strength aspects $SS1 = (0.3, 0.4, 0.6)$ and $SS2 = (0.1, 0.6, 0.6)$ and suppose security requirements of data is $(0.2, 0.5, 0.6)$. If w is 1 then the value of fitness function for two resources become the same. It is, however, clear that the resource 1 is better than resource 2. If the parameter w set to 2 and 3 then the suitable resource can be selected. We set w with 2 for the implementation setup.

example 3.(Evaluating our proposed algorithm on a sample graph):
Supposed workflow with 7 tasks and security requirements aspects for data transmitted between tasks depicted in Figure 4. $d_{i,j}$ shows the size of the output data of each task. As we can see, the sensitive data, the output data of task 1, have security requirement $(1, 1, 1)$. Three resources (VMs) consider for this example. According to Table 8, VM1, VM2 and VM4 consider as we named in this example VM1, VM2 and VM3. The resources includes one private and two public one. The execution time of tasks on resources are depicted in Figure 5. We consider 0.3 and 2 for the maximum security distance (a) and the source depletion factor (w) respectively. For constraints values, user-defined deadline and maxCost, we supposed 40 and 10.
We run our proposed algorithm (PSOWFS) on sample workflow shown in Figure 4. The results are shown in Figure 5.
The most effectiveness of our algorithm is the selection of the resources with the most similar security to input (output) data of tasks, meanwhile the constraints are satisfied. When the security requirements of data are (1, 1, 1) – i.e. max sensitivity- it must be definitely scheduled on private resources. Then, task 1 with sensitive output and task 2 with sensitive input are scheduled on VM3, which is in private cloud, as it is shown in

	T1	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**
VM_1	14	13	11	13	12	13	7
VM_2	7	6.5	5.5	6.5	6	6.5	3.5
VM_3**(private)**	14	13	11	13	12	13	7

Table 8: The execution time of tasks on resources

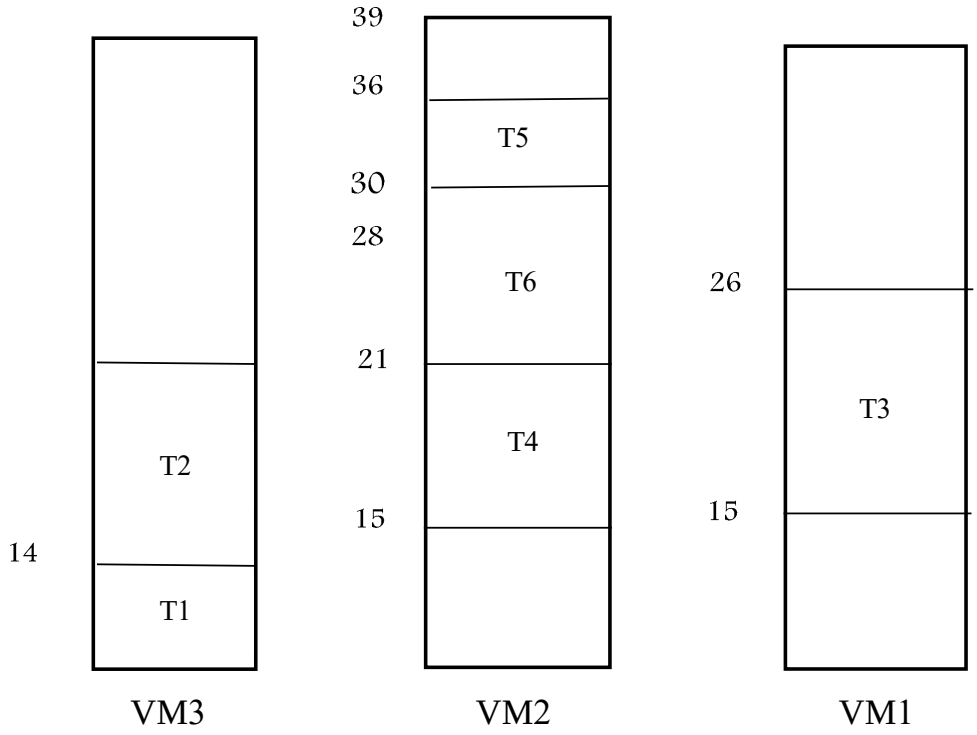

Figure 5: The result of PSOWFS on workflow in Figure 4

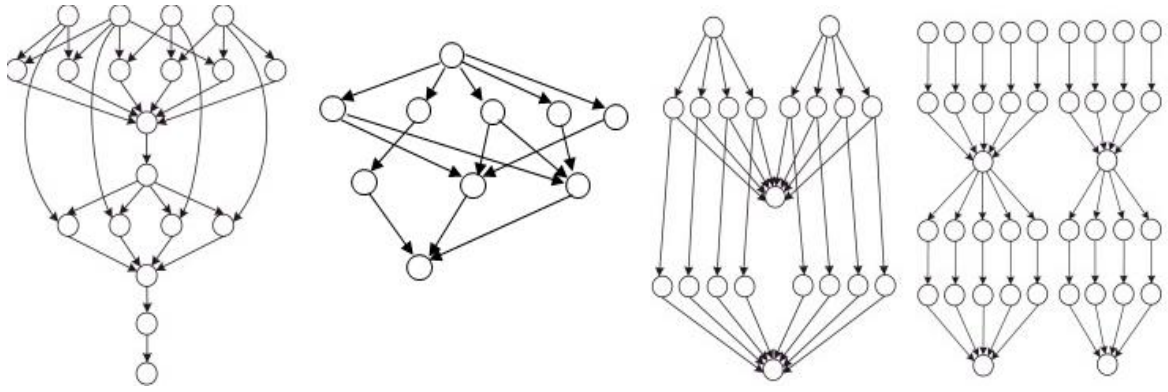

Figure 6: Sample Graphs

Figure 5. The other tasks should be scheduled on resources with security as similar as possible. For example task 3 is scheduled on VM2 that have minimum security distance and is more similar to it than the other VMs. In other words, we let algorithm to be relax to pay a more cost and time and meet the constraints, so that the security of data and resource become more similar. After all, the makespan and cost of our algorithm are below the user-defined constraints.

The simulation and evaluation of our proposed algorithm on real workflows is illustrated in next paragraph.

In our simulation, to evaluate the proposed algorithm, we run our algorithm and HEFT, DHEFT and RANDOM on real workflows i.e. Inspiral, Cybershake, HEFT and Montage[2] shown. Each of these workflows has specific structures as seen in Figure 6. The full description of these workflows is presented in [8]. For evaluating, we added security requirements to these workflows which are in XML format.

The resources that we consider for this experiment was presented in Table 9. As we cansee, VM1, VM2 and VM3 have different properties and are in public cloud. VM4 and VM5 are resources in private cloud which have high security strength aspects (1, 1, 1) and no costs. VM1 and VM3 have the same processing capacity but because of the different security strength aspects have different costs.

As VMs are not in the same cloud datacenter, the bandwidth between VMs is also considered in the range [1500, 3000] Mb/s.

The results of our proposed algorithm (PSOWFS) and the other algorithms are presented in Figures 7,8,9 and 10.

As expected, our PSOWFS algorithm has a minimum security distance, shown in Figures 7,8,9 and 10.Thus, our algorithm is better than the other three algorithms by selecting the best data security distance and satisfying the constraints. It means that our algorithm allowed the makespan to increase until the violation of user-defined deadline didn't occur. So it can select the resource with the most security similarity. As far as we know,

[2]http://pegasus.isi.edu/schema/DAX

	Processing capacity(MIPS)	**Cost per hour**	**Security strength for confidentially**	**Security strength for integrity**	**Security strength for authentication**
VM1	1000	0.3$	0.2	0.3	0.5
VM2	2000	0.6$	0.1	0.2	0.3
VM3	1000	0.4$	0.3	0.4	0.6
VM4	1000	0$	1	1	1
VM5	1000	0$	1	1	1

Table 9: The properties of resources used in this experiment

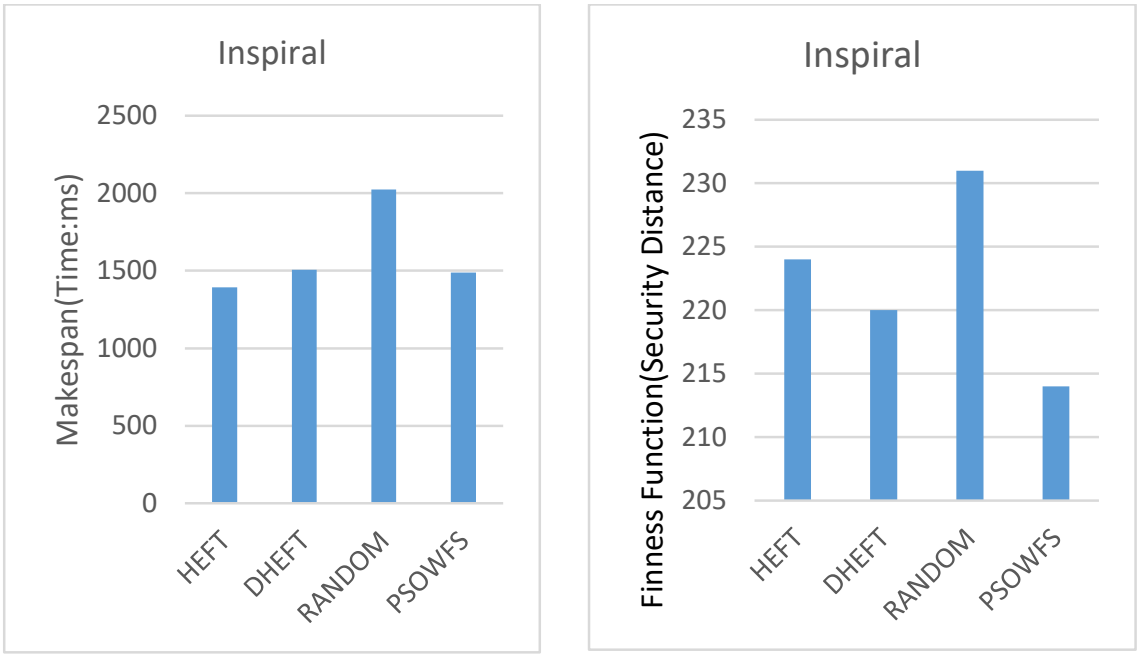

Figure 7: Makespan and Fitness function (security distance) on Inspiral graph

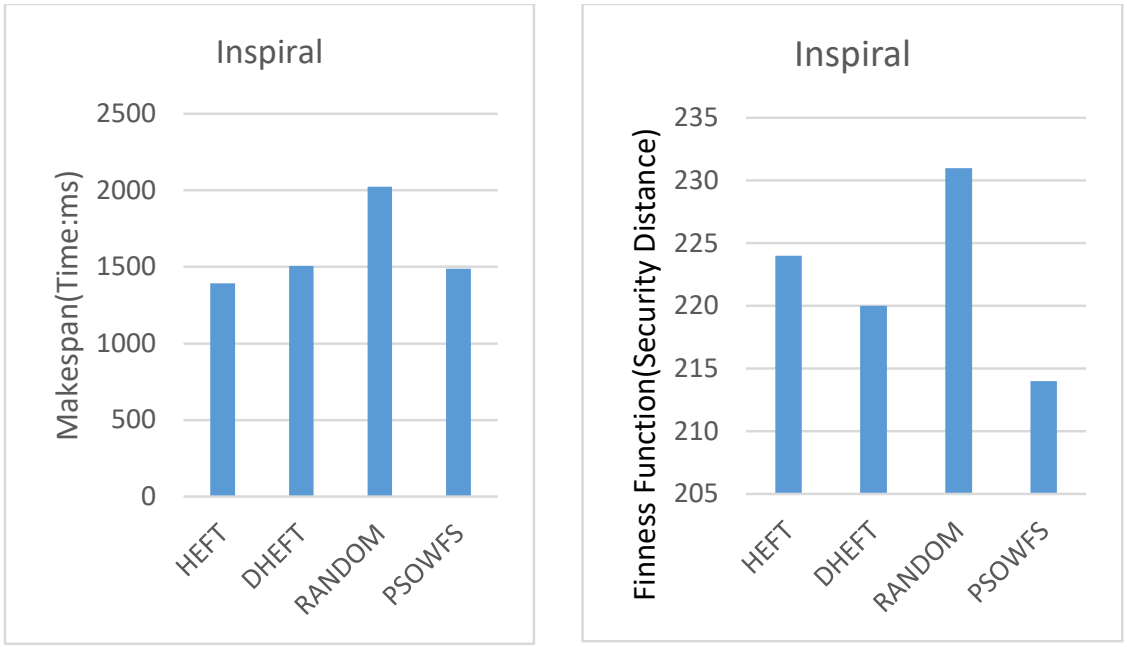

Figure 8: Makespan and Fitness function (security distance) on CyberShake-50 graph

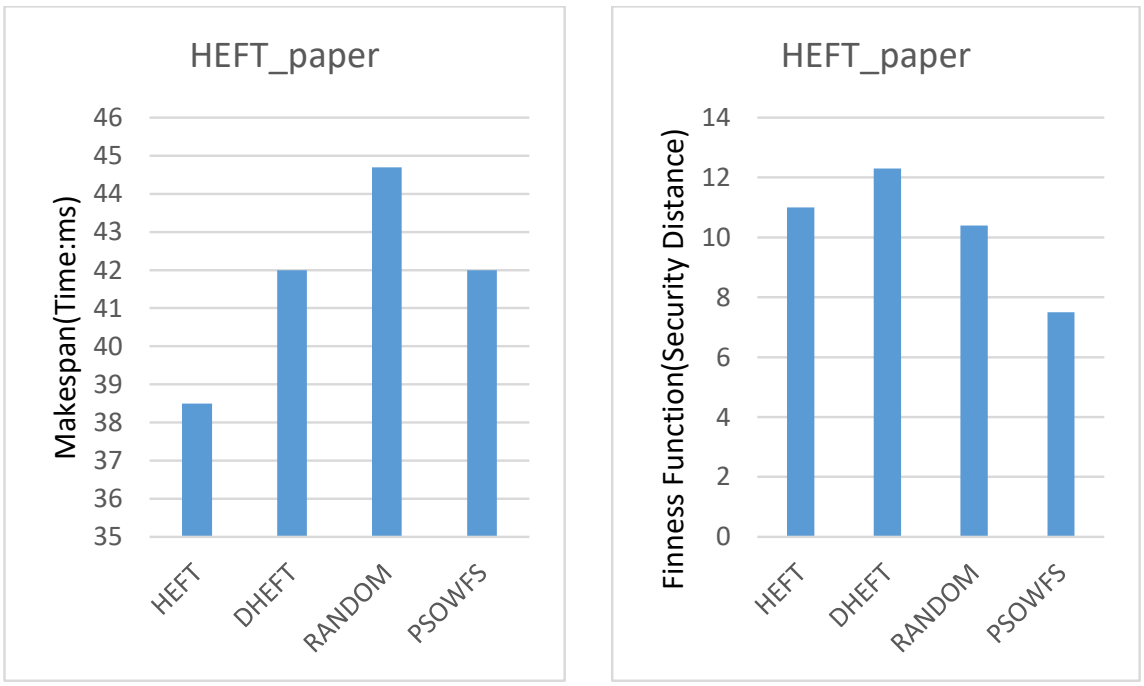

Figure 9: Makespan and Fitness function (security distance) on HEFT_paper graph

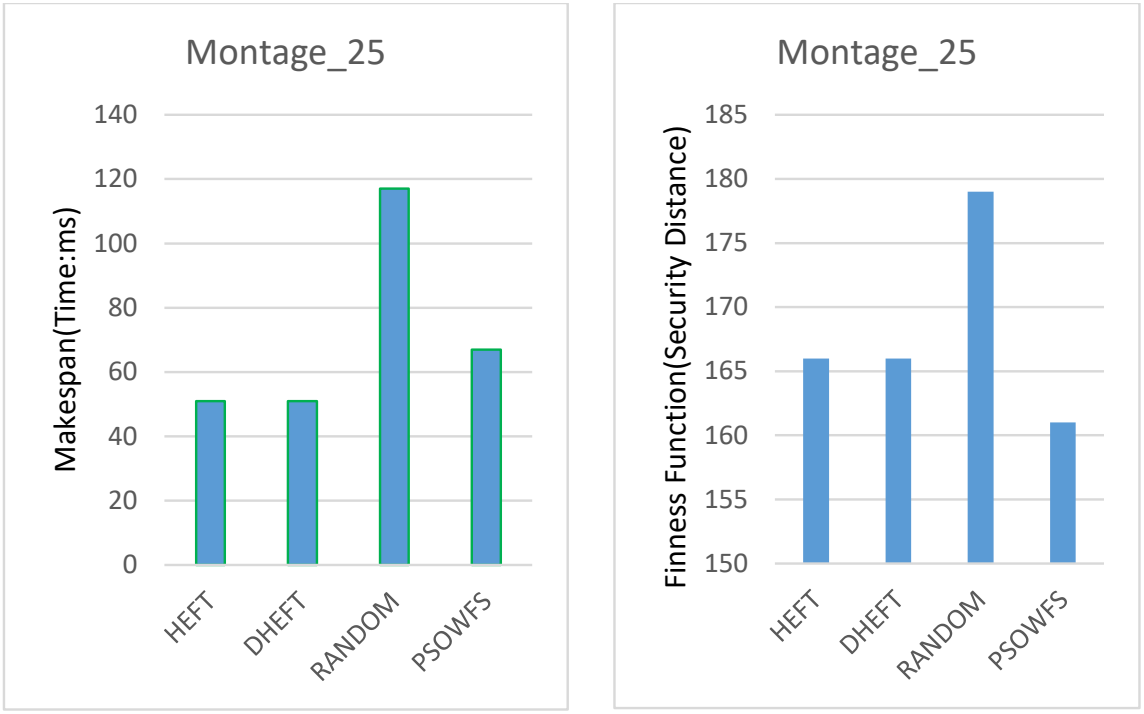

Figure 10: Makespan and Fitness function (security distance) on Montage_25 graph

similarity function for security of data and resources- like this paper- are not considered before. Therefore, in addition to preventing the data from tampering maliciously and accessing illegally, similarity between security of task and resourecs can be helpful. However, applying better security services to the workflow incures longer processing time, as it results in more cost and larger makespan. Hence, algorithm should tradeoff between the cost and makespan to more security similarity for the workflow execution.

6 Conclusions and Future Work

The core issue of cloud computing is task scheduling that has been widdly concerned. The main work for this paper is focused on the following points.
First, in our proposed algorithm, security strength is considered for resources. It means in addition to proccessing and cost, we defined security strength aspects on security services i.e confidentiality, integrity and authentication.
Second, we considerd sensitivity for data transfered between tasks. Sensetivity is defined by the security requirement aspects on security services expressed before.
Third, our proposed algorithm is based on the PSO (Particle Swarm Optimization)- one of the frequently used metahuristic algorithms- which aims to minimize the security distance while meeting the deadline and cost constraints. Minimizing security distance means the most similarity of security between task requirement and resource strength which is our

goal.
Moreover, extensive experiments using three real-world scientific workflow structures demonstrate the effectiveness of our algorithm.
The research work of this paper provides a new contribution for the research of cloud computing task scheduling problem, but there is still a lot of work for improvement. Future studies in this research can be performed in the following directions. First, we will study the security framework for multi clouds [17]. Second, experiments will be held to analyze the performance of the scheduling algorithm in these scenarios using game theory. Third, we will be using our model to interaction graphs on the cloud.

References

[1] Abrishami, H., Rezaeian, A., and Naghibzadeh, M., A novel deadline-constrained scheduling to preserve data privacy in hybrid cloud. In *2015 5th International Conference on Computer and Knowledge Engineering (ICCKE)* (2015), IEEE, pp. 234–239.

[2] Abrishami, H., Rezaeian, A., and Naghibzadeh, M., Workflow scheduling on the hybrid cloud to maintain data privacy under deadline constraint. *Journal of Intelligent Computing Volume 6*, 3 (2015), 93.

[3] Arabnejad, V., Bubendorfer, K., and Ng, B., Scheduling deadline constrained scientific workflows on dynamically provisioned cloud resources. *Future Generation Computer Systems 75* (2017), 348–364.

[4] Bittencourt, L. F., and Madeira, E. R. M., Hcoc: a cost optimization algorithm for workflow scheduling in hybrid clouds. *Journal of Internet Services and Applications 2*, 3 (2011), 207–227.

[5] Chen, H., Zhu, X., Qiu, D., Liu, L., and Du, Z., Scheduling for workflows with security-sensitive intermediate data by selective tasks duplication in clouds. *IEEE Transactions on Parallel and distributed systems 28*, 9 (2017), 2674–2688.

[6] Fernández-Cerero, D., Jakóbik, A., Grzonka, D., Kołodziej, J., and Fernández-Montes, A., Security supportive energy-aware scheduling and energy policies for cloud environments. *Journal of Parallel and Distributed Computing 119* (2018), 191–202.

[7] Gupta, R., Above the clouds: a view of cloud computing. *Asian Journal of Research in Social Sciences and Humanities 2*, 6 (2012), 84–110.

[8] Juve, G., Chervenak, A., Deelman, E., Bharathi, S., Mehta, G., and Vahi, K., Characterizing and profiling scientific workflows. *Future Generation Computer Systems 29*, 3 (2013), 682–692.

[9] Kennedy, J., and Eberhart, R., Particle swarm optimization. In *Proceedings of ICNN'95-International Conference on Neural Networks* (1995), vol. 4, IEEE, pp. 1942–1948.

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

[11] Liu, H., Abraham, A., Snášel, V., and McLoone, S., Swarm scheduling approaches for work-flow applications with security constraints in distributed data-intensive computing environments. *Information Sciences 192* (2012), 228–243.

[12] Liu, W., Peng, S., Du, W., Wang, W., and Zeng, G. S., Security-aware intermediate data placement strategy in scientific cloud workflows. *Knowledge and information systems 41*, 2 (2014), 423–447.

[13] Masdari, M., ValiKardan, S., Shahi, Z., and Azar, S. I., Towards workflow scheduling in cloud computing: a comprehensive analysis. *Journal of Network and Computer Applications 66* (2016), 64–82.

[14] Naghibzadeh, M., Modeling and scheduling hybrid workflows of tasks and task interaction graphs on the cloud. *Future Generation Computer Systems 65* (2016), 33–45.

[15] Sharif, S., Taheri, J., Zomaya, A. Y., and Nepal, S., Mphc: Preserving privacy for workflow execution in hybrid clouds. In *2013 International Conference on Parallel and Distributed Computing, Applications and Technologies* (2013), IEEE, pp. 272–280.

[16] Singh, S., and Chana, I., A survey on resource scheduling in cloud computing: Issues and challenges. *Journal of grid computing 14*, 2 (2016), 217–264.

[17] Sooezi, N., Abrishami, S., and Lotfian, M., Scheduling data-driven workflows in multi-cloud environment. In *2015 IEEE 7th international conference on cloud computing technology and science (CloudCom)* (2015), IEEE, pp. 163–167.