# A Novel Multi-objective Particle Swarm Algorithm Based on a Neighborhood to Search Depth in Task Scheduling by Considering a New Security Model

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

Faculty of Computer Engineering, Yazd University, Yazd, Iran.
[1] m.mehravaran@stu.yazd.ac.ir
2 fadib@yazd.ac.ir
3 pajoohan@yazd.ac.ir

*Corresponding author
Received: 2019-11-24
Revised: 2020-06-28
Accepted: 2020-09-16

**Abstract**
Cloud computing is a novel technology that provides users with better opportunities to gain access to services on the Internet. Users should utilize organizational services to meet their needs. They can also benefit from non-organizational services with high capacity but limited security. This study aims to provide a new security model that addresses security requirements for tasks and data as well as security strength for resources and communication paths. The proposed security model is defined security distance concept. Minimizing security distance has to do with task scheduling so that the resources can be matched with the security level and the data will be fitted into the appropriate communication path. The proposed scheduling algorithm takes the server profit into account in addition to the minimum security distance. The increased server profits can lead to higher resource sharing by the servers. The proposed scenario is implemented based on a neighborhood to search depth in task scheduling. This algorithm utilizes a novel 'far and near neighborhood' approach to select the best particle position. The approach generates both diversity and convergence in the set of answers. Finally, the proposed algorithm is compared with three other similar scheduling algorithms obtained by VNPSO, MPSO and NSGAII, considering the security of the cloud computing environment. The computational results show the effectiveness of the proposed algorithm to obtain resources with similar security and higher server profits.

**Keywords**
Task scheduling, Security requirement, Security strength, Security distance, Multi-objective particle swarm optimization

## Introduction

Cloud computing includes resources that users can utilize on demand. In this way, task scheduling is a strategy to allocate the resources in such a way that the goals and performance criteria are met [1-3]. Cloud environment is the hybrid compromising of public and private clouds [4]. In addition to private cloud resources that have high security, public cloud resources are also needed. The problem of scheduling in a hybrid cloud faces challenges due to the use of public cloud resources such as security considerations and privacy [5, 6]. In IDC[1], security and confidentiality are the most important challenges that threaten the cloud environment [7]. In previous studies, confidentiality issues are considered as sensitive and insensitive [6-8], and security levels have been taken into consideration [8-10]. The security model presented in previous works [9, 11] presents three confidentiality, integrity and authentication services for tasks and data that generate an overlap to enforce security services. For example, the authentication service should be defined on a task, but using this service for data is an extra action. Furthermore, the security model provided in those studies does not cover all security attacks [12]. In addition to the confidentiality service, the data transmitted in the communication path also require a comprehensive service, which is not taken into account in previous works. So, we introduced a new security model in order to address the above-mentioned issues. In addition to defining the security needs for tasks and data, the new security model defines the security of resources for the

[1] International Data Corporation

link between them. In addition, by defining the security distance in the scheduling issue, the model schedules tasks upon resources and data on a communication path with similar security.

The server profit is another parameter that is set forth in the paper as a goal. Services and resources provided by the server are up-to-date and support the costs that the server seeks to earn more by specifying the execution cost for the user. As a result, another aim of this paper is to maximize the server profits.

Since there are no polynomial solutions to scheduling problems [12], different algorithms are typically utilized to solve them. Those algorithms are divided into three types [13], namely metaheuristic, heuristic and hybrid [14].

The algorithm proposed in this paper is to explore the scheduling of a metaheuristic algorithm for the discovery of multi-objective particle swarms based on a neighborhood to search depth. The proposed algorithm is N_MOPSO[2] , which, at the stage of selecting the best particle with a far and near neighborhood, seeks to create both diversity and convergence in the set of the best solutions.

The novelties of this study can be enumerated as follows:

1. A new security model is defined for task scheduling on resources with different security levels.

2. A multi-objective particle swarm algorithm is developed by considering the neighborhood in choosing the best overall particle position. In addition to convergence, a variety of solutions are taken into account by this new version of algorithm.

3. The proposed algorithm is implemented and evaluated in the MATLAB and WorkflowSim environments [15].

The rest of this paper is as follows. In the second section, the relevant works that have discussed security and multi-objective issues in task scheduling are reviewed. The proposed method is described in the third section. In the fourth section, the developed algorithm and similar ones are evaluated and compared with one another. The fifth section of the article presents the conclusion.

## 2. Review of literature

Abrishami et al. [16] suggested task scheduling in a hybrid environment, taking into account the sensitivity of the task. Indeed, instead of defining the level of confidentiality, their algorithm bore on sensitive and insensitive tasks. Sensitive tasks were scheduled in a private cloud while insensitive tasks were scheduled in a public or private cloud environment. The proposed model was implemented by a heuristic method. Then, in another study, Abrishami et al. [17] addressed task scheduling by considering time and cost constraints. They also examined the sensitivity of tasks in a hybrid cloud environment.

Sharif et al. [18] described the issue of privacy by defining three levels of security for tasks and resources. At the highest level of security, tasks were only capable of being run in a private cloud environment. At the second level of security, a task could be scheduled on private resources and some selected public ones. Finally, the lowest-level security tasks could be scheduled on both resources in hybrid cloud environments. The proposed scheduling algorithm was implemented using critical path algorithms [19].

Liu et al. [8] defined a scheduling algorithm with three security modes. The security mode was safe if tasks were only scheduled on a data center at a higher security level. If there were no constraints for scheduling, the security status was said to be risky. On the other hand, if the tasks were scheduled at data centers with the risk of Y ($0 < Y < 1$), the security state was a risk Y. This model was solved using the metaheuristic particle swarm optimization algorithm.

Following that work, Liu et al. [9] introduced authentication, confidentiality, and integrity security services to privacy preservation. Then, they utilized the ant colony optimization algorithm, a kind of metaheuristic algorithm, to define the security requirement of tasks for scheduling.

In 2016, Li et al. [11] proposed a model for security services. This system utilized three security services, including authentication, confidentiality and integrity. Figure 1 illustrates the various services provided in the process of task execution in that study.

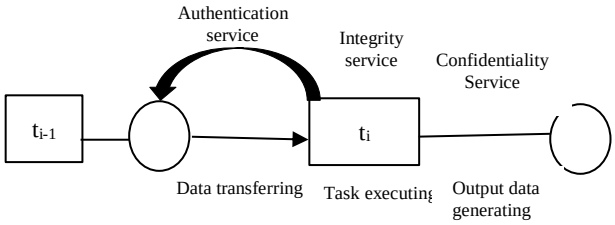


Fig.1. Security services [11].

As can be seen in the figure, each task requires three types of services, including authentication, integrity and confidentiality. The user with various security levels identifies these services. Tables I, II and III show the security service algorithms. Table I illustrates an example of the cryptographic algorithms used for security services. Table II indicates hash functions for authentication service, and Table III specifies the authentication algorithms for authentication services. The declared coefficients in these tables vary from zero to one, where zero shows the lowest power and one represents the highest power of the algorithm. The greater the power of the algorithm, the more complex its calculations, the more costly its execution, and the lower its running speed.

Based on the model introduced by Li, due to the implementation of security services, the overhead can reduce the execution speed and increase the costs. Chen et al. [15] proposed a method that avoided the transmission of sensitive data between resources by performing repetitive tasks on the resources. Thus, it did not require middle data cryptography when executing a

[2] Neighbor Multi Objective Particle Swarm Optimization

task on a single resource. In this regard, the time required for cryptography decreased from the total time of task execution. Metaheuristic algorithms were proposed by considering the security parameter in task scheduling. The scheduling algorithm proposed by Fernandez et al. [20] 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. In their study, the security was considered as a constraint of task scheduling on resources. Wen et al. [21] also tried to minimize the two important goals of scheduling algorithms, i.e. the time of task execution and the cost of task execution on resources. In order to achieve these goals, a multi-objective genetic algorithm was designed that took into account the maximum security constraint for tasks and resources.

**Table I**. Cryptographic algorithm [11].

| Encryption algorithms | Strength Security | Speed (Mb/s) | Overheads (KB/ms) |
|---|---|---|---|
| IDEA | 1.00 | 17.34 | 13.5 |
| DES | 0.9 | 18.21 | 15 |
| Rijndael | 0.64 | 39.88 | 21.09 |
| RC4 | 0.36 | 39.96 | 96.43 |

**Table II**. Hash functions [15].

| 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 III**. Authentication Algorithm [15].

| Authentication Algorithms | Strength Security | Speed (Mb/s) |
|---|---|---|
| CBC-MAC-AES | 0.9 | 163 |
| HMAC-SHA-1 | 0.6 | 148 |
| HMAC-MD5 | 0.3 | 90 |
| RC4 | 0.36 | 39.96 |

Naidu et al. [20] identified the probability of security benchmark according to Equation (1) as the problem constraints. They solved the problem with the proposed Particle Swarm Optimization algorithm. Equation (1) identifies $S_T$ by the level of task security, and $S_C$ shows the level of the security constraints of the resource.

$$Prob_{risk} = \begin{cases} 0 & if\ S_T - S_C \le 0 \\ 1 - e^{-0.5(S_T - S_C)} & if\ \ 0 < S_T - S_C \le 1 \\ 1 - e^{-0.5(S_T - S_C)} & if\ \ 1 < S_T - S_C \le 2 \\ 1 & if\ \ 2 < S_T - S_C \le 5. \end{cases} \quad (1)$$

As mentioned before, security levels are defined for tasks and data. In this paper, in addition to defining the security requirements for tasks and data, a security distance is introduced by a new security model that defines the security strength for resources and communication links between those resources. By reducing the security distance, the tasks are scheduled on the resources with similar security levels, data, and communication paths. The proposed method is presented in the next section.

## 3. The Proposed Method

To describe the proposed method, we first specify the problem space and then describe the multi-objective particle swarm optimization algorithm.

### *3.1 Problem space*

The environment of the task scheduling problem, as mentioned before, is expressed on a hybrid cloud. Another important concept in task scheduling is that of resources. We have some physical resources in the cloud environment, with virtual resources (VM) on them. Virtual resource properties include processing power (the number of instructions when it processes), hourly unit price for the user (Cost, $vm_j$), the final cost for the provider of $Cost_r(vm_j)$, and the security level $SS_{1\varepsilon\{c,i,p\}}(vm_j)$ which can be provided for authentication (a) and public (p) services. It should be noted that, due to their organization, private cloud resources have the highest level of security and do not have cost. Nevertheless, due to their limitations, it is inevitable to use public cloud resources.

In the scheduling problem, the workflow is represented by a directed graph. The tasks are characterized by the graph nodes $T = \{t_1, t_2 \dots t_n\}$, and the transmitted data are identified by the graph edges. If there is an edge between the tasks, one will be the precedence for the other, and the second task should wait for the completion of the precedence. In addition to the volume of data transmissions $d_{i,j}$, the security requirement for data $SR_{l\varepsilon\ \{c,i,p\}}\ (t_i,t_j)$ must be specified. $SR_{l\varepsilon\ \{c,i,p\}}(t_i,t_j)$, i.e. the security requirement, specifies the transitional data between $t_i$ and $t_j$ based on confidentiality service (c), integrity (i) and public (p) service, which is further explained in the next section.

#### *3.1.1 Security considerations*

As it can be seen, the security model shown in Figure 1, does not take all the necessary services into consideration, we propose a new security model indicated in Figure 2. In this figure, the authentication service on resources checks the input data that come from an authorized user. This service is applied to the output data. Therefore, the resource is responsible to verify the authentication service. The confidential and integrity services are checked in the communication paths for the data. The data in the communication path between the resources should be at the same level of security. As a result, we consider confidentiality and integrity services for data and communication paths.

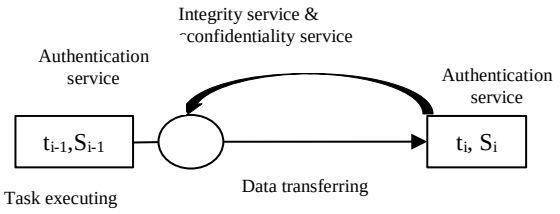


Fig. 2. A new security model for tasks and resources.

In Figure 2, task $t_{i-1}$ is the precedence of task $t_i$. So, after task $t_{i-1}$ is run and the output data are produced, $t_i$ can begin to work. If the task $t_i$ is scheduled on the resource $S_i$ and the $t_{i-1}$ is scheduled on the resource $S_{i-1}$, in order to increase the security, authentication services are deployed to authenticate users who intend to gain access to the input data to prevent spoofing attacks. The authentication algorithms depicted in Table III are utilized to check this service. Moreover, the data generated by the $t_{i-1}$ task, which must pass through the $S_{i-1}$ and $S_i$ resource paths, are also subject to change. To avoid it, the integrity and confidentiality algorithms presented in Tables I and II are used. These algorithms are applied in the resource communication path to prevent the data from being attacked. For the better understanding of certain concepts, they are defined in the following.

**Definition 1:** Security strength for resources

To schedule tasks, resources run the authentication service and ensure that the data come from an authorized user (task); otherwise, they do not accept the entered data. In addition to the authentication service, another security parameter that defines the general security level of the resource is defined for the resource security. Therefore, there are two parameters to represent the maximum security for the resources in reviewing the authentication service and the public security strength. It should be noted that the parameters take values varying from 0 to 1, where zero is the lowest level of security and one is the highest level. Prices are calculated on a clock basis. So, if the resource usage is one hour and a half, it is calculated as two hours. The server profits are calculated by the difference between the unit price of the hour and the expired amount.

**Definition 2:** Security strength for the communication paths between resources

To illustrate the security for the communication path between the resources, three parameters are used each to identify one of the security strengths of the paths for confidentiality, integrity and public security services. Public security services include arrangement, getting lost and repetition.

In previous studies, security for the communication path was not dealt with, while there are security attacks on the communication paths between resources. This paper, therefore, undertakes the investigation of the security strength issue. The factors affecting this security parameter can include the distance between resources, the number of switches between resources, and the position of the nodes between the paths. It should be noted that a communication path is not necessarily a direct communication channel; and it may cross several switches and nodes. Therefore, we define security strength according to the various algorithms used in communication paths. The security strength of the communication path between the two resources i and j is represented by $SS_{1\varepsilon\{c,\,i,\,p\}}(vm_j,vm_j)$. Security is defined by the confidentiality (c), integrity (i), and public security (p) for the communication path between resources.

**Definition 3**: Security requirement for tasks

In some studies, such as [9], the security requirement of tasks has been addressed as three services including confidentiality, integrity and authentication. In this paper, based on Figure 2 and the explanations before, task security services are considered as authentication and public security. Authentication services check the authorized user access to task's code, and public security services protect the modification of codes. So, security requirements can be defined as two coefficients, namely authentication and public security.

**Definition 4**: Security requirements for data

The security requirement services of data are confidentiality, integrity and public security. Confidentiality and integrity services have been utilized to prevent data loss and modification, while public security services prevent data loss and data disorders.

According to the definitions provided, security levels are defined for tasks and resources. The aim of scheduling tasks on resources is to make the levels of task security and resource security similar and to minimize the security risk. Li et al. [12] used the security risk[3] calculation parameter to schedule tasks. The security risk is calculated via Equation 2 and the Poisson function, which is a random function. In addition to involving complex calculations, the Poisson function fails to yield correct answers in some cases. This problem is expressed in Example 1. In Equation 2, $p$ is the security risk, $sr$ stands for the task security requirement, and $sl$ indicates the resource security strength. The set $\{a, q, c\}$ specifies the services of confidentiality, integrity, and authentication.

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

$$p(t_i) = 1 - \prod_{l\in\{a,g,c\}} p(t_i, sl_i^l) \quad (2)$$

**Example 1**: Violation of the Poisson function

Assume two resources with $SS_1$ = {0.14, 0.36} and $SS_2$ = {0.14, 0.14} security strengths. If there is a security requirement SR = {0.14, 0.36}, the Poisson function calculates the security risk and finds it 1 for both resources based on Equation 2. While it is quite clear that resource number 1 fully meets the data security requirement, resource 2 is far from the security requirement of the data. Regarding this problem, we utilize the Manhattan Distance in the proposed method to calculate the security distance, as described below.

**Definition 5:** Calculation of the security distance

Security distance specifies the degree of the compliance of security requirement with security

[3] Security risk is the security level of the task in task scheduling on sources

strength. To calculate the security distance, we use Manhattan distance which shows the difference between two dimensions. Equation 3 indicates the security distance of SR security requirement from SS security strength. In this equation, indices *a*, *i*, and *p* refer to authentication service, confidentiality service, and public service respectively. Also, β is specified as a coefficient in Equation 4. The lower the parameter $\partial$, the more consistent the services are in terms of security. To explain the parameter β, the following two explanations are needed:

$$\partial = \beta * |SS_a - SR_a| + \beta * |SS_c - SR_c| + \beta * |SS_i - SR_i| + \beta * |SS_p - SR_p| \quad (3)$$

**Definition 6**: Maximum security factor α

If the security strength of a resource is less than the security requirements of the data, the difference must be less than the maximum security factor, which is represented by α in Equation 4. In other words, to have a secure schedule, the security requirement should be lower than the security strength. However, this is not always possible. Therefore, in order to simplify the problem, resources are sometimes selected with a security strength value less than the security requirements. In this case, the difference should not be more than the Max security factor. For example, if the data security requirement is 0.5 for confidentiality and the maximum security factor is 0.1, resources with security services of higher than 0.4 can be selected.

**Definition 7**: Resource depletion factor (*W*)

Source depletion factor refers to the depletion of selecting a resource with security strength higher than security requirement in contrast to selecting a resource with security strength less than security requirement.

In Equation 3, when the security strength of the resource is greater than the data security requirement, the coefficient β is 1. On the other hand, when the security strength of the resource is less than the data security requirement, the value of β is equal to the resource depletion factor (*w*). This parameter causes Equation 3 to be multiplied by *w* in cases where security requirement is more than security strength to increase the security distance.

$$\beta = \begin{cases} w & if\ (SS_c(vm_j) - SR_c(t_p, t_i) < \alpha \\ 1 & else \end{cases} \quad (4)$$

Based on the security definitions, the objective function and the constraints involved in task scheduling can be defined as in the following sections.

*3.1.2. The Objective function*

In this study, for task scheduling, we examine two important objectives to satisfy both the user and the service provider. The first one is to maximize server profits. Server profit is derived from the difference between the cost of execution and the expenses. Example 3 illustrates a server profit.

**Example 3**: Suppose two resources are provided by the server. Both are at the same level of security, but the server expense for resource one is 0.2 dollar per hour and the one for resource two is 0.3 dollar per hour. If the server wants to have the same earnings from both resources, it should specify different costs for the resources. For example, the user's executing cost is 0.4 dollar per hour for resource one and 0.5 dollar per hour for resource two. Thus, the server profit for both resources is 0.2 dollar per hour. Obviously, the user chooses the resource that costs less because the security level is the same. Therefore, if the server wants to use both resources, it has to ignore the same profit and define the same execution cost for the user. In this case, the profit earned by the server will be greater when resource one is selected.

So, one of the goals of this study is to maximize server profits. Equation 5 indicates the profit obtained from scheduling $t_i$ on resource $vm_j$.

$$Profit(t_i, vm_j) = w(t_i, vm_j) * [cost(vm_i) - cost_r(vm_i)] \quad (5)$$

$w(t_i, vm_j)$: *The execution time of task $t_i$ on $vm_j$*

$cost(vm_i)$: *The cost of task scheduling on resource $vm_j$ for the user*

$cost_r(vm_i)$: *The resource cost for the server*

*Equation 6 shows the profit achieved by scheduling the entire task on the resources for the server.*

$$Profit = \sum_{t_i}^{allTask} \sum_{vm_j}^{allVms} Profit(t_i, vm_j) \quad (6)$$

The second goal is to minimize the security distance of tasks and resources so as to schedule tasks on similar security levels of resources and to schedule data on similar security paths. Equation 7 shows the security distance of each task as a security distance of task and resource (in terms of authentication and public services) and the security distance of data and communication path (in terms of confidentiality, integrity, and public services).

$$Security\ distance = (\sum_{t_i \in all\ Tasks} \partial_i) \quad (7)$$

$$\partial_i = \beta * |SS_a(vm_j) - SR_a(t_i)| + \beta * |SS_p(vm_j) - SR_p(t_i)| + Relations\_Security$$

$$Relation\_Security = \sum_{t_k \in\ all\ childs\ of\ t_i} \beta * |SS_l(vm_j, vm_i) - SR_l(t_i, t_k)| + \sum_{t_p \in\ all\ parrents\ of\ t_i} \beta * |SS_l(vm_i, vm_j) - SR_l(t_p, t_i)| \quad l \in \{c, i, p\}$$

$\partial_i$: *Security distance of the $i^{th}$ task from resource $vm_j$.*

*SS and SR indicate security strength and security requirements respectively.*

$SS_{a,p}$ *($vm_j$): This item specifies the security strength of resource j on the authentication and public service.*

$SP_{p,a}$ *($t_i$): This item specifies the security requirement of task i for the public security and authentication service.*

$SS_l$ *($vm_j$, $vm_i$): This item defines the security strength of the communication path of resource j and i on services l.*

$SR_l$ *($t_i$, $t_k$): This item specifies the security requirement of transition data from task i to task k.*

*Relation-Security: This item calculates the amount of the security required for the inputs (outputs) of a task with the security strength of the communication path between the scheduled resources. It is assumed that task i is scheduled to be on resource $vm_j$, and the parent and child of task i are scheduled on resource $vm_i$. It should be noted that the security distance has been calculated on confidentiality, integrity, and public services.*

Two methods can be used to address both objectives [22]. The first is the use of a weighted sum, which is not recommended due to the objectives being contradictory. The second method is the use of multi-objective solutions that address both goals. This study makes use of the latter method.

*3.1.3. Constraints*

The two important parameters of the scheduling algorithm, namely makespan and user's costs, are considered as the problem constraints. As such, the total running time (makespan) should not be longer than the maximum time specified by the user, and the cost of running the entire task on resources should not be greater than the maximum specified by the user. The problem constraints are delineated as follows:

*Subject to: Makespan< Deadline*

*TotalCost<MaxCost*

$TaskCost(t_i, vm_j) = W(t_i,vm_j)* Cost(vm_j)$

$TotalCost=\sum_{t_i \in workflow} TaskCost(t_i, vm_j)$ (8)

To calculate the total scheduling time (Makespan), the scheduling time of each task must be determined first. The completion time for each task is equal to the initiation of run time (EST) plus the time it takes to run on resource W ($t_i$, $vm_j$) as well as the time required for the security computations (SC). Equation 9 expresses the completion time of the $i^{th}$ task on resource j.

$EFT(t_i,vm_j)= EST(t_i)+ W(t_i,vm_j)+ +SC(t_i, vm_j)$ (9)

The initiation time of each task (EST) is equal to the maximum time for the completion of the parent's task plus the required data transfer time (DT), which is calculated according to Equation 10. In this expression, the time of data transmission is equal to the ratio of the data volume ($d_{p,i}$) to the bandwidth ($bw_{n,m}$) of the resources that schedule the tasks. If the tasks are scheduled on the same resources, the transition time is considered to be zero.

$EST(t_i)= max_{t_p \in parents\ of(t_i)}(\ EFT(t_p) + DT_{p,i}))$ (10)

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

The time required for the security computations of task $t_i$ on resource $vm_n$ -SC ($t_i$, $vm_n$) is calculated via Equation 11. In this expression, for each service, the time is added as a security overhead (SO) to the runtime. In fact, for the $i^{th}$ task and its input and output data, the total security overhead generated by each service is computed.

$SC(t_i, vm_n) = \sum_{ll \in \{i,c,p\}} SC_l(t_i, vm_n)$ (11)

$SC_l(t_i, vm_n) = SO_{a,p}(t_i, vm_n) + \sum_{t_p \in parents\ of(t_i)} d_{p,i} * \sum_{l \in \{i,c,p\}} SO_l(vm_k, vm_n) + \sum_{t_p \in childs\ of(t_i)} d_{i,p} * \sum_{l \in \{i,c,p\}} SO_l(vm_n, vm_m)$

*$d_{p,i}$: The volume of the data to be transmitted between the two tasks p and i*

*$SO_{a,p}$ ($t_i$, $vm_n$): A security overhead is generated by security algorithms on authentication and public security services.*

*$SO_l$ ($vm_n$, $vm_m$): The overhead of algorithms for the security services of data in the communication path between resources n and m. The subscript l can be any of the confidentiality, authentication, and public services. It should be noted that task i is scheduled on resource n, the children of task i on resource m, and the parents of task i on resource k.*

Based on the definitions given above, the next section presents the normal multi-objective particle swarm algorithm and the proposed multi-objective particle swarm algorithm.

*3.2. Multi-objective particle swarm algorithm*

Given that there is no polynomial solution for task scheduling in the cloud environment [12], we can use metaheuristic methods. One of the methods in this regard is the use of the particle swarm optimization (PSO) algorithm to solve the problem. The algorithm is appropriate owing to its low parameters and high speed [23]. In this algorithm, first, each response is identified by a particle. Secondly, the movement of the particles in space to find new responses leads to the convergence of swarms toward an optimal response. The particle movement involves three factors as follows:

A) The speed and the position previously shown by the particles

B) The best position that every single particle has experienced in the entire search space (pbest)

D) The best position experienced by all the particles (gbest)

The first step in implementing a metaheuristic algorithm is to show the answers. Accordingly, a number of particles are randomly created as the primary swarm. For each particle, a velocity vector is used whose initial value is zero. At this point, the best position experienced by each particle is the position in which it is located. Therefore, the current position of each particle is recorded as its best position. In each stage, the non-dominated particles of a swarm are determined and kept in an archive. Then, the best position is determined for all the particles, and, based on Equation 12, the new velocity of each particle is calculated.

*Velocity [i] = a*velocity [i] + c1*r1*(pbest[i]-population [i]) + c2*r2*(gbest-population [i])* (12)

In Equation 12, the positive coefficients c1 and c2 accelerate convergence. In fact, these coefficients control the best position of each particle (pbest) and the best position of the entire particles (gbest). To consider the impact of two positions equally, the coefficients must be assumed equal. In this case, r1 and r2 are the two random numbers between zero and one that create dispersion and diversity in the set of answers. For the particle to follow its previous behavior, its previous speed is used in the equation, which adds to the weight of the previous particle with inertia (a). Usually, variable numbers range from 0 to 0.9. Equation 13 calculates the new position of each particle based on the current population [i] and the previous speed velocity [i].

*Population [i] = velocity [i] + population [i]* (13)

In the next step, the particles are evaluated. In a single-objective algorithm, the objective function is already determined, based on which the particles can be evaluated and the best ones can be found. However, in multi-

objective problems with multiple-goal functions, the concept of domination is used to find the optimal set of answers.

**Definition 8:** The concept of domination

If we consider the two particles a and b, particle a will dominate b if and only if particle a is greater or equal to the values of particle b in all its objective functions.

*3.3. The particle swarm algorithm based on neighborhood*

Due to the high convergence rate in the particle swarm optimization algorithm and the probability of the solution's being stated in local optimum, we use the new neighborhood approach to select the best overall particle position from the archive members. Algorithm 1 describes the steps of the proposed algorithm.

**Algorithm 1: The pseudo-code of the proposed multi-objective particle swarm algorithm**

1. Set the primary values of the algorithm.
A) Generate an initial population of the particles (answers).
B) Determine the primary value of the parameters.
D) Determine the archive members of the initial population.
E) Determine the best position for each particle.
F) Determine the best position for all the particles.
2. Continue steps 3 through 7 until the number of repetitions is satisfied.
3. The new position and velocity of each particle is calculated according to Equations 13 and 14.
4. Assess the current population based on the objective functions and constraints.
5. The best position of each particle will be updated if the new particle dominates it.
6. The best position of all the particles is determined by the distance from the archive.
7. A new archive is available from the new population and the members of the previous archive.
8. The archive is considered as the optimal answer set.

There are certain parameters in the PSO algorithms. In our work, we set the number of population members to 20 and the number of repetitions to 100. Coefficients c1 and c2 are set to 2.05, and the inertia weight "a" is set to 0.9 for initialization. The size of the archive is the same as that of the population.

To model the solutions for each particle, we list the number of tasks. The corresponding number is considered by each member of the list, which specifies the resource number. The order of the scheduling tasks is prioritized according to Equation 14 [24].

$$rank(t_i) = \begin{cases} w_{i,r} & t_i = t_{exit} \\ w_{i,r} + max_{t_{p \in childs\ of(t_i)}}(rank(t_i)+c_{i,p}) & otherwise \end{cases} \quad (14)$$

In our algorithm, in addition to creating the initial population via the random method, we also generated HEFT [25] and DHEFT algorithms [26]. These algorithms help the problem converge sooner to yield optimal responses.

After the initial population is created, the archive has a collection of the non-dominated members of the population. If the archive is full, the crowding distance policy is used as shown in Figure 7 to remove the element. Crowding distance represents population density, and, wherever the population distance of a particle is lower (greater density), it can be eliminated in order to maintain dispersion and diversity in the responses.

At each stage, after the creation of the new particles in the population, the archive should choose the non-dominated members among the former values and the new members of the population.

As stated, we use the proposed algorithm to select the best position of particles via the neighboring method. We calculate the distance between the particles and the archive members. This new method, selects the nearest neighbor for a half of the population and the farthest neighbor for the other half with the archive members. It should be noted that the mentioned distance refers to Euclidean distance. In other words, Euclidean distance considers the two objective parameter (i.e. server profit and security distance). Conceding that there are 20 particles in the population, the best position for the first 10 particles are selected via the shortest Euclidean distance between the objective functions of the particles and those of the archive members. The next 10 particles will choose a member of the archive that has the greatest distance to the particle. The best position is, thus, obtained for each particle. By determination of the values in Equation 11 and calculation of the new velocity, new particle positions can be obtained. Also, the members of the new archive are selected based on the current population and the previous archive members. At the end, the archive members are presented as the best answers.

## 4. Evaluation of the proposed algorithm

We have evaluated the proposed scheduling problem with the MOPSO algorithm [27] based on the Zitzler-Deb Thiele (ZDT) criterion function [28]. The results show a great inconsistency in the set of optimal solutions. The next section presents the comparative results.

*4.1. Evaluation of the proposed algorithm based on the ZDT criterion function*

Our proposed algorithm is designed in the MATLAB environment. Based on the ZDT1-ZDT6 criterion functions, the mean results show better dispersion of responses in Pareto than in MOPSO. For each test function, all the algorithms are run for 20 times. The parameters are set as nPop=50, maxIt=200, c1=1, c2=2, and w=0.5. The Pareto graphs for the proposed algorithm and MOPSO are presented in Figures 3 and 4.

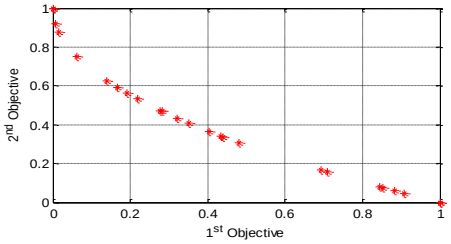


Fig. 3. Pareto graph of the MOPSO algorithm

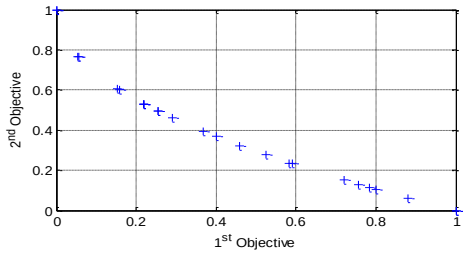


Fig.4. Pareto graph of the proposed N_MOPSO algorithm

As can be seen in the figures, the optimal response diagram in the proposed algorithm shows more dispersion than in the previous algorithm. In fact, it is clear that the selection of the nearby neighbors from the archive members improves the convergence while the selection of the farthest neighbors improves the diversity of the algorithm. The next section evaluates the proposed algorithm on the scheduling problem.

*4.2. Evaluation of the proposed algorithm on the task scheduling problem*

The proposed task scheduling algorithm is implemented in a hybrid cloud with security considerations and on server cost in the WorkflowSim simulation environment. In this environment, the codes written in Java language simulate the cloud environment and can be used to implement the algorithm [15]. The version of this simulator which is used in this paper is Workflowsim 1.0.

Suppose the number of task is N, the size of population is P and the repetition of algorithm is M. Therefore, the time complexity is as follows. As the size of each particle is equal to the number of tasks, we should review the list to calculate the goal functions (i.e. security distance and server profit), which emerges to be O(NP) for the population. The best position for all the particles is obtained in O(P ^ 2). This time order is achieved through calculating the distance between each particle and the archive members, assuming the size of the archive is equal to the population. If the archive is full, the crowding distance is used to locate the best particle, so the time order of this part is O(PlogP). Finally, the time order of the algorithm is O(M (NP + P ^ 2). As the number of the tasks is usually greater than that of the population, the time complexity of the algorithm is O(MNP).

*4.2.1 Simulation parameters*

In the simulation environment, the proposed algorithm N_MOPSO [4] and the algorithms for the evaluation of MPSO [29], Ranking-MOPSO [22] and NSGAII[30, 31] are implemented on real-world workflows, including Inspiral, HEFT, and Montage[5]. Each of these workflows has a specific structure, and its full details are expressed in Juve's paper [32]. Security requirements are added to the workflows to investigate the algorithm in the XML format.

In Jana's paper [29], an algorithm is developed without using an archive. At first, the median of the particles is calculated, and then the nearest population to this median is taken up as the best position for the particles. This algorithm also performs mutation and swaps in different parts of each particle in the population.

Another scheduling algorithm is Ranking-MOPSO provided by Alkayal et al. [22]. It uses the resource ranking method for each goal function. In fact, the resources are sorted based on individual goal functions, and the sum of their ranks is obtained for those goal functions. The rank values are used to evaluate each resource in the MOPSO.

Keshanchi at al. [30, 31] proposed genetic algorithms. They encoded chromosomes and implemented crossover and mutation functions. The present study benefits from the encoding of this algorithm.

The proposed algorithm intends to increase server profits and reduce the security distance of tasks and resources. It also defines some restrictions for the problem so that time and user's costs may not exceed the defined user constraints. To compare the proposed algorithm with some others, security risk [33] is utilized. This formula is shown in Equation 1.

*4.2.2 Algorithm evaluation*

The proposed algorithm and the others are compared in terms of three parameters including security distance, security risk, and server profits. If the size of the population is 20, the number of iterations will be 100, and c1 and c2 will equate 1.49. Using a HEFT graph, Figure 5 presents the evaluation results based on these parameters. As can be seen, the proposed algorithm provides more similar security for tasks, data, and resources; that is, the security distance and the security risk parameters in the proposed algorithm are less than those in the other two algorithms. The server profit in Ranking-PSO is higher than that in the proposed algorithm. However, Ranking-PSO is not comparable with the proposed algorithm in terms of security parameters.

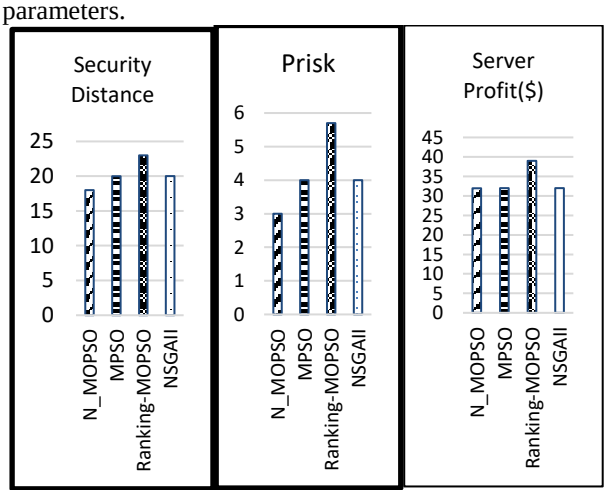


Fig. 5. The results of evaluation on a HEFT graph

The two parameters of scheduling in Figure 6 namely time and cost, are calculated for all the tasks via Equations 7 and 8. Makespan in the proposed algorithm is lower than MOPSO, and its costs are lower than that one. Nevertheless, the proposed algorithm defines these

[4] Neighbor Multi Objective Particle Swarm Optimization

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

parameters as two constraints so that the user-defined constraints can be obtained. In spite of considering time and cost as constraints, we let them increase so that the tasks can be scheduled on the resources with similar security.

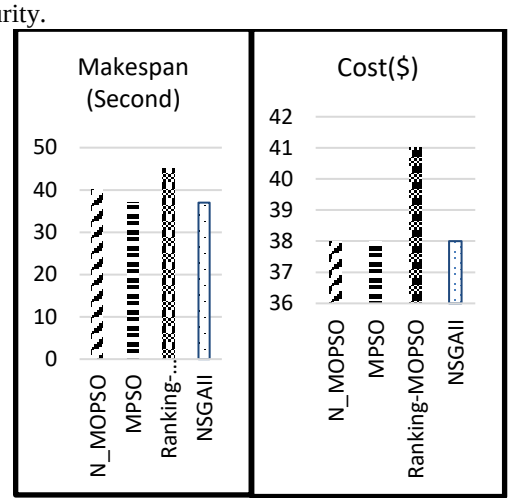


Fig.6. The constraints on a HEFT graph

Figure 7 shows the results of evaluating the proposed algorithm on a Montage graph. As it can be seen, the parameters of security distance and security risk in the proposed algorithm are lower than the other ones. In other words, as compared to the Ranking-PSO, there is a 57% improvement achieved. Moreover, an improvement of 25% can be observed in comparison with the NSGAII. The server profits have also increased.

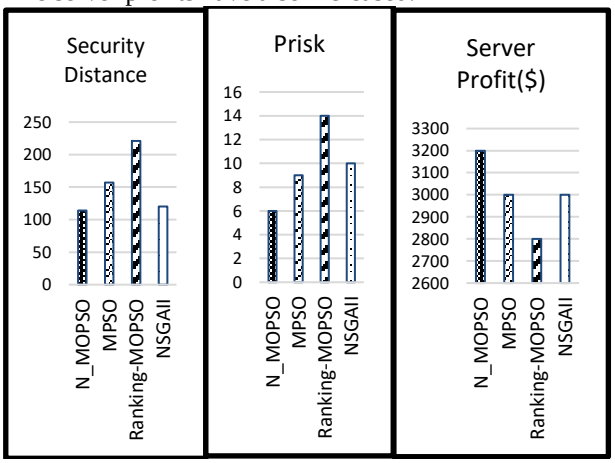


Fig. 7. The results on a Montage graph

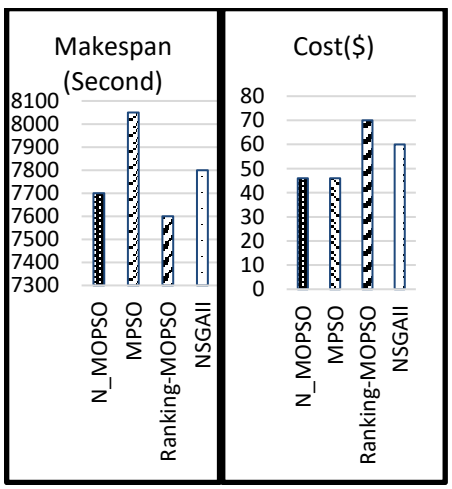


Fig. 8. The constraints on a Montage graph

The evaluation results of the proposed algorithm are presented on an Inspiral graph in Figure 8.

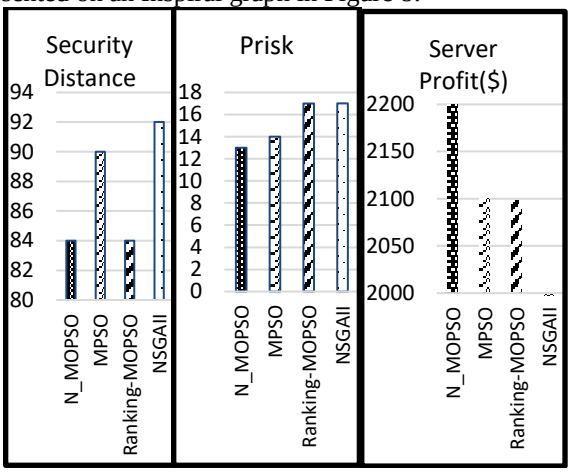


Fig.9. The evaluation results on an Inspiral graph

As it can be observed in Figure 9, the security distance is the same for the Ranking-PSO and the proposed algorithm, but the security risk and the server profits point to the improvement of the proposed algorithm. The security risk of the Ranking-PSO has dropped by 23%, but the server profits have increased dramatically in the proposed algorithm.

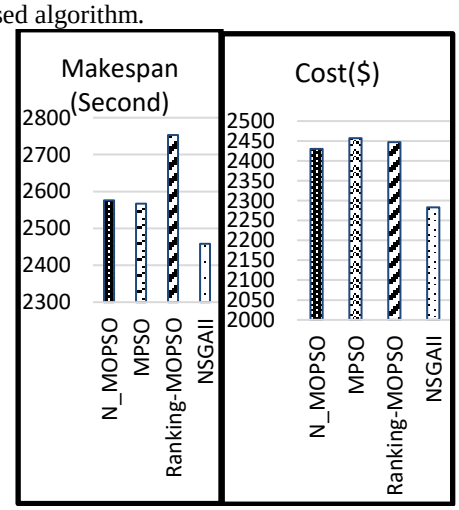


Fig.10 The constraints on an Inspiral graph

The problem constraints are shown on an Inspiral graph in Figure 10. Due to using the archive to obtain the best particles and the closest and the farthest neighbors, the proposed algorithm can help Pareto solutions converge and bring about variation to achieve an optimal solution. The computational results are also presented on real-world graphs and then compared with the results of similar algorithms that improve goal functions and take problem constraints into consideration.

## 5. Conclusion and recommendation for further research

In this paper, a novel particle swarm optimization algorithm was developed based on a neighborhood to search depth. The algorithm was used to solve the problem of scheduling tasks in a hybrid cloud environment with security considerations. The previous

security model was improved, and its deficiencies were eliminated. According to the new security model, security levels are defined for data, tasks, resources and the communication paths of resources. In addition, the concept of security distance, which is a new one, is used in task scheduling. Thus, resources can match the security level of tasks, and the data will fit into appropriate communication paths. Increasing the profit of servers and providing them with better satisfaction is another aim of the new model. The computational results showed that the proposed algorithm can satisfy the goals of corresponding problems acceptably.

For further research, secure scheduling is recommended to be done on other types of clouds, such as multi clouds [34]. It can also be done for multiple workflows (with multiple sets of task graphs), such as workflows with the possibility of data transmission [35]. Improvement of the multi-objective MOPSO algorithm and other objective functions [23] is another recommendation for further studies.

## 6. References


[1] C. Jianfang, C. Junjie, and Z. Qingshan, "An optimized scheduling algorithm on a cloud workflow using a discrete particle swarm," *Cybernetics and Information Technologies,* vol. 14, pp. 25-39, 2014.

[2] M. Naghibzadeh, "Modeling Workflow of Tasks and Task Interaction Graphs to Schedule on the Cloud," *CLOUD COMPUTING 2016,* p. 81, 2016.

[3] M. Yazdanbakhsh and R. Khorsand, "A Task Scheduling Strategy to Improve Qualitative Features in the Cloud Computing Environment," *Tabriz Journal of Electrical Engineering*, vol. 49, pp. 1427-1437, 2019 (in persian).

[4] L. Singh and S. Singh, "A survey of workflow scheduling algorithms and research issues," *International Journal of Computer Applications*, vol. 74, 2013.

[5] R. Gupta, "Above the Clouds: A View of Cloud Computing," *Asian Journal of Research in Social Sciences and Humanities*, vol. 2, pp. 84-110, 2012.

[6] T. Ghafari and S. Bakhtiari Chehelcheshmeh, "Secure Outsourcing Cloud Data using Lattice-based Secret Sharing," *Tabriz Journal of Electrical Engineering*, vol. 49, pp1211-1221,2019(in persian).

[7] H.Abrishami, A.Rezaeian, M.Naghibzadeh, "Scheduling in hybrid cloud to maintin data privacy", 20th National CSI Computer Conference, 2015. (In Persian)

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

[9] W. Liu, S. Peng, W. Du, W. Wang, and G. S. Zeng, "Security-aware intermediate data placement strategy in scientific cloud workflows," *Knowledge and information systems*, vol. 41, pp. 423-447, 2014.

[10] H. Chen, X. Zhu, D. Qiu, L. Liu, and Z. Du, "Scheduling for workflows with security-sensitive intermediate data by selective tasks duplication in clouds," *IEEE Transactions on Parallel and Distributed Systems*, 2017.

[11] Z. Li, J. Ge, H. Yang, L. Huang, H. Hu, H. Hu, *et al.*, "A security and cost aware scheduling algorithm for heterogeneous tasks of scientific workflow in clouds," *Future Generation Computer Systems,* 2016.

[12] M. L. Pinedo, "*Scheduling: theory, algorithms, and systems*", Springer International Publishing, 2016.

[13] F. Wu, Q. Wu, and Y. Tan, "Workflow scheduling in cloud: a survey," *The Journal of Supercomputing*, vol. 71, pp. 3373-3418, 2015.

[14] M. Masdari, S. ValiKardan, Z. Shahi, and S. I. Azar, "Towards workflow scheduling in cloud computing: a comprehensive analysis," *Journal of Network and Computer Applications*, vol. 66, pp. 64-82, 2016.

[15] W. Chen and E. Deelman, "Workflowsim: A toolkit for simulating scientific workflows in distributed environments," in *8th IEEE International Conference on E-science* , pp. 1-8, 2012.

[16] H. Abrishami, A. Rezaeian, and M. Naghibzadeh, "Workflow Scheduling on the Hybrid Cloud to Maintain Data Privacy under Deadline Constraint," *Journal of Intelligent Computing Volume*, vol. 6, p. 93, 2015.

[17] H. Abrishami, A. Rezaeian, and M. Naghibzadeh, "A novel deadline-constrained scheduling to preserve data privacy in hybrid Cloud," in *5th International Conference on Computer and Knowledge Engineering (ICCKE)*, pp. 234-23, 2015.

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

[19] S. Abrishami, M. Naghibzadeh, and D. H. Epema, "Deadline-constrained workflow scheduling algorithms for Infrastructure as a Service Clouds," *Future Generation Computer Systems*, vol. 29, pp. 158-169, 2013.

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

[21] Y. Wen, J. Liu, W. Dou, X. Xu, B. Cao, and J. Chen, "Scheduling workflows with privacy protection constraints for big data applications on cloud," *Future Generation Computer Systems*, 2018.

[22] E. S. Alkayal, N. R. Jennings, and M. F. Abulkhair, "Efficient task scheduling multi-objective particle swarm optimization in cloud computing," in *41st IEEE Conference on Local Computer Networks Workshops (LCN Workshops)*, pp. 17-24, 2016.

[23] K. Pradeep and T. P. Jacob ,"CGSA scheduler: A multi-objective-based hybrid approach for task scheduling in cloud environment," *Information Security Journal: A Global Perspective*, vol. 27, pp. 77-91, 2018.

[24] V. Arabnejad, K. Bubendorfer, and B. Ng, "Scheduling deadline constrained scientific workflows on dynamically provisioned cloud resources," *Future*

*Generation Computer Systems,* Vol.75, pp. 348-364, 2017.

[25] N. Chopra and S. Singh, "HEFT based workflow scheduling algorithm for cost optimization within deadline in hybrid clouds," in *Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, pp. 1-6, 2013.

[26] G. Kaur and M. Kalra, "Deadline constrained scheduling of scientific workflows on cloud using hybrid genetic algorithm," in *7th International Conference on Cloud Computing, Data Science & Engineering-Confluence,* pp. 276-280, 2017.

[27] Z. Fan, T. Wang, Z. Cheng, G. Li, and F. Gu, "An Improved Multiobjective Particle Swarm Optimization Algorithm Using Minimum Distance of Point to Line," *Shock and Vibration* ,vol. 2017

.

[28] R. Fan, L. Wei, X. Li, and Z. Hu, "A novel multi-objective PSO algorithm based on completion-checking," *Journal of Intelligent & Fuzzy Systems,* vol. 34, pp. 321-333, 2018.

[29] B. Jana, M. Chakraborty, and T. Mandal, "A Task Scheduling Technique Based on Particle Swarm Optimization Algorithm in Cloud Environment," in *Soft Computing: Theories and Applications*, ed: Springer, pp. 525-536. 2019.

[30] A. S. Kumar and M. Venkatesan, "Multi-Objective Task Scheduling Using Hybrid Genetic-Ant Colony Optimization Algorithm in Cloud Environment," *Wireless Personal Communications,* vol. 107, pp. 1835-1848, 2019.

[31] 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.

[32] G. Juve, A. Chervenak, E. Deelman, S. Bharathi, G. Mehta, and K. Vahi, "Characterizing and profiling scientific workflows," *Future Generation Computer Systems,* vol. 29, pp. 682-692, 2013.

[33] P. S. Naidu and B. Bhagat, "Secure workflow scheduling in cloud environment using modified particle swarm optimization with scout adaptation," *International Journal of Modeling, Simulation, and Scientific Computing,* vol. 9, p. 1750064, 2018.

[33] N. Sooezi, S. Abrishami, and M. Lotfian, "Scheduling Data-Driven Workflows in Multi-cloud Environment," in *2015 IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom)*, pp. 163-167, 2015

[35] M. Naghibzadeh, "Modeling and scheduling hybrid workflows of tasks and task interaction graphs on the cloud," *Future Generation Computer Systems,* vol. 65, pp. 33-45, 2016.