Meta-heuristic Innovative Algorithm of Multi Objectives in Tasks Timing at Cloud Computing System

Mohsen Sojoudi, Ahmad Tavakoli*, Mehdi Norouzi

Abstract--- *In this article a mathematical model with twin objectives is presented. The objectives are considered as: Minimization of the maximum tardiness of tasks completion time and the total early tasks penalties. Since tasks timing is a tardy and indefinite factor in cloud computing; therefore problem solving model is used as the combined Meta-heuristic innovative algorithm of multi objective swarm of particles based Parto archive has been used. The suggested algorithm with genetic operators as well as the directed and repeated counterpart structures in the format of multi operators are taken to assess the algorithm application. The results will be sorted based on quality, distraction, integrated, the number of non-defeated solutions and the gap from the ideal one is compared with the evolutionary algorithm results titled genetic algorithm. The final results of solved model indicate that firstly, this algorithm is stronger than NSGA-II algorithm but is weaker in timing, norms and scales. In other words, the suggested algorithm, is more capable to discover solutions, accordingly*.

Keywords--- *Multi objective particles swarm, Cloud computing system, tasks timing, NSGA II*.

I. INTRODUCTION

Cloud computing has become necessary in recent years as one of the most significant technologies to deliver advanced services via the public internet. The task scheduling is an important process in the infrastructure as a service in cloud computing which aims to implement those tasks imported to the system in an efficient way and considering the characteristics of the cloud environment. (Choudhary and Peddoju, 2012)

One of the main challenges in cloud computing is scheduling the tasks and resources sharing its purpose to implement the users' tasks to multiple computing resources efficiently by considering other features of the cloud environment.

Also task scheduling is considered as virtual machines for scheduling unit in order to allocate the heterogeneous physical resources to perform the tasks. (Braun, 2001)

By increasing the number of cloud users, the tasks to be scheduled will also increase. The scheduling in a cloud is a mechanism that allocates users' tasks to appropriate resources for implementation and directly affects the performance of the cloud.

Mohsen Sojoudi, Phd Student in Operations Research (OR)/ Management Sciences at Ferdowsi University of Mashhad, Mashad, Iran. Email: mohsened@yahoo.com
Ahmad Tavakoli, Associate Professor in Management, Faculty of Economic and Administrative Sciences at Ferdowsi University of Mashhad, Mashad, Iran. Email:* tavakoli-a@um.ac.ir
Mehdi Norouzi, Associate Professor in Molecular Genetics at Tehran University of Medical Sciences (TUMS), Tehranm Iran. Email: mnourouzi@tums.ac.ir

In the cloud computing system, the timing is used to reduce the expected time of requests and increase the productivity of resources. (Plestys & Colleagues 2007)

Scheduling algorithms plan to achieve high efficiency by minimizing the execution time of tasks. But the problem here is the balanced task distribution between resources and the sub-tasks between resources with regard to resources. (He, 2003)

Due to the importance of cloud computing system and task time scheduling in this system, this paper deals with the problem of scheduling tasks in cloud computing system with the aim of maximum completion of tasks minimization. On the other hand, met heuristic methods have been defined as optimization methods for problems for which finding solutions in a cloud computing systems are NP-HARD problems and their solutions with exact methods are impossible. (Lavanya, 2019)In this paper, a hybrid particle swarm optimization (PSO) algorithm based on pareto archive is proposed for solving these problems.

Research History

In each distributed computation system such as grid resources in cloud computing system, different scheduling algorithms have been suggested to resource allocation for user applications. (Koch, 2012)

In cloud computing environment, many scheduling algorithms have been proposed to provide the users requests. FIFO algorithms and rotational speed (RR) algorithm are based on the mode of task scheduling in cloud computing. (Salot, 2013)

Pandey et al (2010) have presented an optimization algorithm for particle swarm optimization (PSO) in order to schedule cloud resources applications and to calculate the cost of data transmission and complexity, accordingly. In 2009, Gu et al also proposed a new resource scheduling strategy based on genetic algorithm in cloud computing environment to consider load balancing and scheduling of the virtual machines. Gun et al (2010) suggested a “simulated annealing algorithm “ to schedule tasks in the cloud computing environment. Zhu et al. (2011) developed a multi-agent hybrid genetic algorithm for scheduling tasks in cloud system that applied processor productivity and load balance. Liu et al. (2005) suggested an ant colony optimization algorithm in scheduling service flow with respect service quality necessities.

Huang et al. (2013) discussed “exploring resource allocation and scheduling algorithms in cloud computing” and in 2014, Nirubah and John have investigated the impact of scheduling algorithms on energy efficiency in cloud computing. Leila Ismail and Abbas Fardoun also considered energy for scheduling tasks in a cloud computing system In 2016. In 2018, They have used a simulation tools. Juarez et al. have applied tasks scheduling in a cloud computing system, in parallel and they took two objectives of minimum completion the maximum time and energy flow into consideration to optimize the function of a two- objective heuristic algorithm. M. Lavanya et al. tried the issue of scheduling tasks in a cloud environment based on the SLA and proper time processing and presented an optimization algorithm to solve this problem in 2019.

In 2019, Sanaj & Joe Prathap have applied a squirrel search algorithm to schedule tasks in a cloud computing environment (IAAS). In this study, they have proposed a chaotic squirrel search algorithm for optimal scheduling of the multi-task problem in an infrastructure as a service (IaaS) in cloud computing environment.

Tong et al. (2007) developed a scheduling program in cloud computing environment using deep Q learning. Sharma and Garg (1998) developed an artificial neural network method to schedule tasks in cloud data centers with efficient energy.In this paper, they have used neural networks under supervision to provide optimal scheduling of tasks with the aim of efficient resource allocation and energy. Lee (2020) has done the same in hierarchical cloud service system in smart city to optimize the particle swarm optimization algorithm with the aim of maximizing the accountability and productivity of resources. Sharma and Garg (1998) presented a genetic algorithm inspired from harmony to schedule efficient scheduling tasks in cloud data centers and solve the scheduling problem. Therefore they have tested the tasks with the aim of increasing energy efficiency with various methods using the genetic algorithm and discovered a new combinational planning, accordingly.

Table 1 – Research Gap

Solution Algorithm	Others	Task Relevance (Classification Based)	Math Modeling	Tasks Classification	Objective Functions		Subject	Year	Researcher
					Multi-Purpose	Single Purpose			
Multi-objective Swarm Optimization Algorithm & Non-dominated Sorting Genetic Algorithm		*	*	*	*		Presenting a multi-objective met-heuristic algorithm to solve the tasks scheduling in cloud computing system	2020	This article
Genetic Algorithm						*	Genetic algorithm inspired from harmony to schedule efficient tasks scheduling in cloud data centers	2020	Sharma & Garg
Particle Swarm Optimization Algorithm				*		*	Optimal Task Scheduling & Allocation of Cloud System Resources In A Smart City	2020	Lee
Neural Network & Genetic Algorithm						*	Artificial Neural Network Based Method For Efficient Task Scheduling With Energy in Cloud Data Centers	2020	Sharma & Garg
Squirrel Search Algorithm				*		*	Squirrel Search Algorithm For Task Scheduling In Cloud Computing System - IAAS	2019	Sanaj & Prathap
Q- Learning						*	Cloud Computing Environment Using Deep Q- Learning	2019	Tong et al.
TBTS SLA-LB &				*	*		Task Scheduling In A Cloud Environment Based on SLA & Proper Time Processing	2019	Lavanya et al.
Multi-objectives Optimization Algorithm of Food Bacteria					*		Tasks Scheduling In Cloud System Using Multi-objective Optimization Algorithm of Food Bacteria	2018	Sobhanayak et al
Innovative Algorithm					*		Tasks Scheduling In Cloud System In Parallel	2018	Juarez Et al.
Simulation						*	Tasks Scheduling In Cloud System With Energy	2016	Ismail & Fardoun
Genetic Algorithm						*	Impact of Scheduling Algorithms On Energy Efficiency In Cloud Computing	2014	Niruban & John
Ants Colony						*	Review on The Allocation of Resources & Tasks Scheduling Algorithm in Cloud Computing	2013	Huang

As Shown in the above table, the difference of the present study with other researches in the proposed mathematical model is considering the simultaneous optimization of two objectives, tasks dependency on each other and finally the present two algorithms for solving the model.

The only reference for tasks categorization is Dashti research (2012) which is not a math model and the solution algorithm covers single objective optimization (minimizing maximum completion time). According to the literature review and research characteristics in the past, innovation can be mentioned as:

- To design the multi-objective mathematical model in order to solve scheduling problems in cloud computing systems considering the objectives of minimum completion time and minimizing the sum of delayed duties to servicing the cloudy system users.
- Grouping tasks to be carried out by each resource and the dependency among these tasks
- The application of iterative guided neighborhood search operators in genetic metheuristic algorithm of non-defeated ranking 2 is on the basis of Pareto archive to resolve the multi- objective model.

Mathematical Model

The tasks scheduling problem in cloud computing system involves “n” independent tasks of which requires these tasks to be processed and completed to the existing resources. The goal is to schedule and allocate tasks to the available resources to optimize the objectives for the problem. One of the assumptions considered in this paper is the dependence of the tasks to each other. In other words, the initiation of tasks requires their completion, accordingly since there are prerequisite relations among them. One of the other considerations in this paper, is the classification of associated tasks with a network of graph.

The purpose of this paper is to provide a mathematical model for double objectives for tasks scheduling in the cloud computing system. The goals in this model are as: minimization of maximum amount of tasks completion of minimization of the total number of tasks penalties. It is to be noted that the goals are not consistent with one another and the minimization of the maximum time completion will raise the total expedited penalties. Therefore it can be stated that the goals considered in the proposed model are paradoxical. For mathematical modeling, several default assumptions are considered:

The number of input tasks into system is much higher than that of the computational resources; each task is only performed and completed on one source; tasks are interdependent and the relationships between them are defined. Each resource has the ability to perform its functions, but each resource at a moment performs only one task. The resources are heterogeneous and have different hardware and software features.

Model Indexes

- l :The Number of Tasks Groups
- i,j : Tasks Index Groups
- l و l 'Group Index
- nl : Number of Tasks in Group I’
- K : Number of Resources و Resources Index K For

Parameters & Model Collections

- :$A^{l} = \{a_1^{l}, a_2^{l}, \dots, a_{nl}^{l}\}$I’ Set of tasks Group
- :$G_l = (V_l, E_l)$ I’The Network of Prerequisite Relations in tasks Group
- K : Set of Machines K={1,2,…, K }
- p_{ikl}:Task Processing time , i ‘By Machine k ‘

- D_{il} :Task Deadline i ' in l '
- :DE_{ik} Data Tasks i '
- :Mm_k Machine Memory k '
- :MC_k Machine Capacity time k '
- :B_k Machine's Bandwidth k '
- We_{il}:Delayed Task Penalties I' l'

Model Variables

- :$x^{\mathrm{l}}_{\mathrm{ik}}$Task Starting Time i ' in Group l ' by Machine k'
- :$y^{\mathrm{l}}_{\mathrm{ik}}$ If the resource k' the task i in group l ' Other wise the amount of variable is 1

$$f^{\mathrm{ll}'}_{\mathrm{ijkk}'} = \begin{cases} 0 \ if \ x^l_{ik} > x^{l'}_{jk} \\ 1 \ \text{if} \ x^l_{ik} \leq x^{l'}_{jk} \end{cases} \forall i, j, l, l', k$$

This variable is used for the limitation in allocation each resource to one task at the same time.

- Cil : Task Time Completion i ' in Group l '
- Cmax :Maximum Task Time Completion .
- Eil : Task Delayed I'in Group I'

In this study, the tasks processing time is calculated as below : (Ramezani et al, 2013)

$$p_{\mathrm{ikl}} = y^{\mathrm{l}}_{\mathrm{ik}}\left(\frac{DE_i}{Mm_k + Mc_k} + \frac{DE_i}{B_k}\right)$$

Main Structure of Mathematical Model

Suggested Double Objectives Mathematical Model is as per following:

1. $\min z1 = C_{max}$
2. $\min z2 = \sum_{l=1}^{L}\sum_{i=1}^{nl}(we_{il} \times E_{il})$
3. $c_{il} = \sum_{k=1}^{K} y^{\mathrm{l}}_{\mathrm{ik}}(x^l_{i,k} + p_{ikl}) \ \forall i, l$
4. $T_{\mathrm{il}} = \max\{0, \mathrm{D}_{\mathrm{il}} - \mathrm{C}_{\mathrm{il}}\} \ \forall i, l$
5. $\{ x^{\mathrm{l}}_{\mathrm{ik}} - \mathrm{x}^{\mathrm{l}'}_{\mathrm{jk}} \geq \left(y^l_{ik}\right)\left(y^{l'}_{jk}\right)(p_{jkl'}) - M\left(f^{ll'}_{ijk}\right) \ \forall i, k, l, l', i \neq j$

$$x^{\mathrm{l}'}_{\mathrm{ik}} - \mathrm{x}^{\mathrm{l}}_{\mathrm{jk}} \geq \left(y^l_{ik}\right)\left(y^{l'}_{jk}\right)(p_{ikl}) - M\left(1 - f^{ll'}_{ijk}\right) \ \forall i, k, l, l', i \ \neq j\}$$

6. $\mathrm{Min}\{\left(x^l_{jk'} \middle| \mathrm{y}^{\mathrm{l}}_{\mathrm{jk}'} = 1\right\} - \max\{\left(x^l_{ik} \middle| \mathrm{y}^{\mathrm{l}}_{\mathrm{ik}} = 1\right\} \geq p_{\mathrm{ikl}} \ \forall \mathrm{k}, \mathrm{k}', \mathrm{l}, \ \forall \ \mathrm{i} \geq \mathrm{j}$

(i must complete before j in l project)

7. $x^{\mathrm{l}}_{\mathrm{ik}} \leq M y^l_{ik} \ \forall i, k, l$
8. $C_{max} = \max(C_{il}) \qquad \forall i, l$)
9. $x^{\mathrm{l}}_{\mathrm{ik}} \geq 0 \ \forall i, k, l$
10. $E_{\mathrm{il}} \geq 0 \ \forall l$

$y_{ik}^{l} = \{0,1\} \; \forall i, k, l$

The equation 1 shows that the objective function is the minimization of the maximum completion time. Equation (2) represents the second objective function, which is the minimization of the sum of the early penalty. The third one refers to time required for processing. Equation (4) is used to calculate the delay in tasks completion.

The equation (5) guarantees that a resource such as K source cannot simultaneously be assigned to 2 tasks in a separate sets or two tasks belonging to one class. When a source K? is assigned to a task, it has to wait until the assigned task is completed so that the resource can be assigned to another task (i.e., m is a very large number).

The sixth equation is to ensure consistency between the tasks of each class I. If in a class such as I", the task i" is the prerequisite for j", the minimum starting time j will have to exceed the completion time of j will have to exceed the completion time of I" (The maximum task i starting time plus the task processing time I".

The equation7 shows that in the allocation of resource k for task I, this task can be initiated with this allocation (M is a very large number). The equation 8 calculates the maximum amount of task completion time which is equivalent to the maximum times of completion. And the rest of equations (9-11) are representing the range of decision variables.

II. METHODOLOGY

Designing particle optimization algorithm to solve the proposed model

The proposed structure of multi-objective integrated PSO is proposed to optimize the performance of the objective function considered in the model. The aim of the design of the above method is to achieve Pareto solutions. To evaluate this algorithm, its output is compared with NSGA-II algorithm.

In this paper, an optimization algorithm based on Pareto archive based on genetic algorithm operators as well as neighborhood guided structures is combined to solve the task scheduling problem in the proposed cloud computing system. The pseudo code below shows the overall structure designed for the swarm optimization algorithm of multi-objectives.

{Generate N feasible solutions as initial population .

Apply improvement procedure for generated particles.

Apply feasibility check procedure for improved particles .

Initialize pg and pi .

Initialize the Pareto archive set so that it is empty

While a given maximal number of iterations is not achieved

Update particle by eq. (12)

Improve population of particles .

Apply feasibility check procedure .

Evaluate the updated particles to get the new p_i *and* p_g

Update Pareto archive.

Select N solutions whit higher quality and higher diversity as population for next generation.

End while}

Return Pareto archive

In order to adapt the PSO and NSGA-II algorithms, it is to be examined how the solutions or particles are represented based on tasks scheduling in different categories and machines assigned to allocate machines to tasks.

In this paper, the solution representation includes two structures, the first one consists of a two dimensional matrix whose rows are equal to the number of vertices; that is each row contains tasks scheduling that are related to that line. In fact in this structure, tasks of each class are represented by observing constraints. The first matrix representation of the solution structure is described in figure 1. Suppose that a classifier has three categories as 6, 8 and 6 tasks.

Task/Category	T1	T2	T3	T4	T5	T6	T7	T8
1	1	2	4	3	5	6	0	0
2	1	2	3	6	4	5	7	8
3	1	2	3	4	5	7	0	0

Figure 1- Example for Matrix 1 of the solution monitoring structure

As shown in the above figure, the matrix has 3 lines in the first row, the sequencing of the first batch tasks in the second row, the sequencing of the second batch tasks and the third row, shows the third batch tasks.

The second structure is a two dimensional matrix in which it is determined that each task will be assigned to each task in each category. The dimensions of the matrix are like the first matrix. In each house, the number of machine is assigned to the activity of the house in the first matrix. Suppose that the above example is compared with the item 5 of resources

Task/Category	T1	T2	T3	T4	T5	T6	T7	T8
1	2	2	1	4	1	5	0	0
2	1	3	3	5	4	1	2	2
3	4	4	2	3	3	5	0	0

Figure 2- Example Matrix 2 of the Solution

To interpret the above matrix, the first matrix data should be provided. For example, the numbers1 & 2of the first one are carried out by the source number 2. Task 4 from the first class by resource 1 and task 3 from the same class by resource No. 4 and finally the task 5 by the first resource and task 6 by the source No. 5. (Check the first row of the 2nd Matrix). Likewise the allocation of resources to the second and third classes correspondent rows is determined.

In this paper, a series of two sided scheduling and series scheduling (Neuman[1] 18, 2003) is used to generate the possible sequence of tasks of each category and then due to the amount of resources required and constraints of the resource interference constraints, the second matrix is constructed for each of the matrixes related to tasks in the clusters (1st Matrix). Each of these methods will generate feasible N sequencing separately for each of the tasks cluster and then the second matrix is constructed for each of the tasks related to tasks in the class and the N solution is selected as

[1] Neuman

the initial population of the solutions. To select the answers 2N of the two methods in a set using Deb's Rules (K Deb[2] 19, 2002) have been classified and for any surface, Crowding Distance[3] 20 has been calculated as well as the N answers which have higher quality and dispersion will be selected. In this paper, improvements on answers have been designed as far as possible. Their procedure is based on the variable neighborhood search (VNS)21. Two neighborhood search structures (NSS)22 are designed as a combined VNS structure. The neighborhood search structures used are as follows.

The first category of neighborhood search:

This operator is taken from Shadrokh & Kianfar 23(2007). The index of a group will be formed randomly and integrated and later.The second matrix is then applied on the corresponding sequence and updated according to changes in the first matrix as well as the constraints of the model.

The second category of neighborhood search:

The second category of neighborhood search operates on the sequencing of the activities required in the interval [l,n-l] at consecutive home. As described in the previous section, two neighborhood search structures are coupled to the variables neighborhood search structure(VNS) which is as follows:

```
{For each input solution s
  K=1 //( k indicates the number of neighborhood search operator)
  While the stopping criterion is met do
        S1=Apply mutation type k
        S=Acceptance methode(S, S1)
         If s is improved then
            K=1
         Else
            K=k+1
            If k=3 then
               K=1
             Endif
End while}
```

As implied from the above structure, after applying the neighborhood structure on the solution, the acceptance procedure has been applied to the response and one of the two responses is selected as the next iteration (VNS) and it selects the dominant answer on the other solution.

Particles Updates

Here, we have used genetic algorithm for updating the particles. The method is as per following:

$$x_i^{t+1} = (x_i^t - p_i^t) + (x_i^t - p_g^t) + \overline{x_i^t} \quad (12)$$

[2] K Deb
[3] Crowding Distance

In the above equation:

- *Particle* x_i^{t+1} *:i ' iterative(generation) t+I'*
- *Particle* x_i^t *:I' (Iterative) t '*
- p_i^t *:I' The best answer ever been in this generation*
- p_g^t *t' The best answer in iteration*
- $\overline{x_i^t}$ *:A neighborhood produced by mutation operator* x_i^t
- *'-' : Crossover or intersection operator (crossover)*
- *'+' The symbol for "Choice"*

In fact, the answer "i" is generated in iteration t+1; two of the resulting intersection operators between x_i^t and p_i^t as well as two of the resulting intersection operator on x_i^t and p_g^t and one of the actions of the mutation operator on x_i^t Finally, the answers among the 5 answers and input answers which have higher quality and dispersion are selected as x_i^{t+1} . In fact, p_g^t and p_i^t are used as the guide to achieve the next iteration solutions.

Intersect Operator

In this part, a single cross-point operator is used.

Mutation Operator

The mutation operator updates the particles used by a local search of multi-objectives operators with iterative driven local research method. (Geiger 24 , 2007)

The two operators described in the improvement section are used in the operatory searching method for the generation of the neighbors. The multi-objectives search method algorithm is shown below.

{Multi objective multi operator search framework

Get solution x , set papprox={x}

Repeat (2-9)

Randomly select some x in papprox for

Which nh(x) has not been investigated yet

For all NR={nh1,nh2,…,nhr}

Generate neighborhood nhi(x)

Update papprox with all elements xnh in nhi(x)

if x in papprox then

Mark x as 'investigated'

Endif

Untile no element as x in papprox with x still to be investigated

Return papprox}

In the search procedure used in this research and the algorithm, a set of solutions (Papprox) is constructed which basically contains only the initial solution (input answer). In each iteration of this method, a member of the set that has not been investigated randomly selected and the neighboring solutions will be generated using two neighboring search functions. Then by using the dominant relations, the set (Papprox) is updated and whole members of the set are reviewed. At the end, the algorithm which consists of optimal local set (Papprox) of solutions that are in the vicinity or neighboring of the initial solution, is returned as the result of the algorithm.

In this research, the search operators of the neighborhood in the form of iterative neighborhood search with guided motion and mutation are used of which the structure is represented (Geiger 2007) as follows; this search function for optimization of NP-HARD has been designed.

{Iterated local search

Get solution x

xn=local search(x)

2.x=perturbation(xn,history)

3.xl=local search(x)

4.xn=acceptance criterion(xn,xl,history)

5. if stopping condition is not met, goto 2}

In this research, two neighborhood search functions described in the improvement procedure section are used as two iterative functions. In fact, in the multi-objectives search structure NR={IT1,IT2} in each neighborhood search function, an answer is sent to the corresponding function as the step 1 to the corresponding function and in the next step, the operator is applied on the solution. After generating a neighborhood, according to the past information, the solution after generating a neighborhood, the disturbance operator described below is applied to the neighborhood, and the corresponding neighborhood search operator is applied to the new solution. Finally, using a reception criterion which is based on dominant relations, one of the two neighboring solutions will be accepted. This will continue until the stop criterion in fact a new solution will be generated per iteration.

Updating p_i^t **&** p_g^t

Among the discovered neighborhoods for this response, for each particle if there is a better neighbor than the existing pI , pi will be replaced, instead; otherwise it will remain unchanged. Of all the answers give, if the best one is better than pg, then it will be replaced or will remain unchanged.

Updating Pareto Archive

In the proposed algorithm, a set called” Pareto Archive” is considered to hold non-dominated solutions generated by the algorithm. This set is updated at each algorithm iteration. Updating is so that the solutions produced in that iteration and the solutions in the Pareto archive are answered, classified and generated together into a mating pool, then among these solutions, the solutions in the first level or the same non-dominated solutions in the first level or the same non-dominated solutions are selected and considered as the new “Pareto Archive”.

Choosing the Next Generation Crowding Distance Answers

In each iteration, the algorithm needs a population of the solutions, in order to select the population of next iteration and after classification and calculation of crowding distance criteria for any answer considering its level in accordance with Deb Rules(Deb, 2002) N refers to the solutions with the highest quality and dispersion selected as the iteration population.

NSGA-II Algorithm

NSGA-II is the genetic algorithm for solving multi-objectives problems and has been presented through NSGA-II protocol By Deb in 2002. The general structure of the algorithm is as below:

{Generate N feasible solutions as initial population at random.

Initialize algorithm parameters.

While algorithm terminated

Apply selection operator for select parents.

Apply crossover operator.

Apply mutation operator.

Combine current population and new solutions generated in current iteration as q.

Apply fast-sort-nondominated method.

Calculate crowding distance for each solution.

Select N solutions with higher quality and higher diversity for next iteration.}

NSGA-II algorithm performance is based on the quality of the solutions according to their quality and the first criterion will be the quality and dispersion. In this algorithm framework, after generating the initial population, the mutation and intersection operators are applied to the solutions. Later the new solutions will be generated, all solutions are classified at the same level. The dispersion criterion using the Deb's formula (2002) is calculated. At the end of each iteration, n as (population size) which has a higher quality and dispersion, is selected as generation and iteration solutions.

It is possible to produce basic solutions in this algorithm and meet the intersection operator such as the production of primary answers in the PSO algorithm. The dual tournament procedure has been used to select the parents. In each selection, two non- recessive responses will be selected as the parents. To perform the mutation operator, the second neighborhood search operator, described in the improvement procedure section, is also applied.

Data Analysis

After designation of the model parameters and algorithm, the problem are solved by the proposed algorithms, accordingly. In the followings, we describe the parameters and the sample characteristics and finally the solutions results will be investigated.

General assumptions of algorithms or characteristics of the model samples or parameters:

A) **Model Parameters Settings**

- Data for each task in integration interval (400-800 Mbps) is produced.
- The memory of each machine is 512 Mbps
- Integrated interval bandwidth is 100-500 Mbps

- The delivery time of each task is 0.7-1 seconds
- Each machine capacity in integration interval is 100-500 millions of commands per seconds
- Early penalties in integration intervals are considered **1-20**

B) Assumptions For Solutions Algorithms

- Population rate for NSGA-II and PSO algorithm is equal to 200
- The number of iteration in variable neighborhood search algorithm is equal to 10
- The number of iteration of both algorithms is equal to 300
- To perform NSGA-II, the mutation operator rate is 0.1 and intersection operator 0.8 as well

The Results of Sample Solutions

Since there is no specific library for task scheduling problems and also there is no possibility of using standard ones, it has been tried to solve the problems as per designed by means of algorithm in the research by Batool Dashti (2012). The Sample problems table designed is shown as below:

Table 2 – Sample Problems

No.	Number of Tasks	Number of Resources
1	50	5
2	100	5
3	150	5
4	200	5
5	300	5
6	400	5
7	500	5

As discussed, the proposed model in this research paper has two objectives, minimizing the maximum completion time as well as minimizing the early fines and penalties tardiness. While in Dashti (2012) research project, the goal is to minimize the maximum completing; therefore in this paper after solving the model, the first objective function of the present research model is compared with the objective function of Dashti's (2012) results of these comparisons are presented in table 3.

Table 3- The Comparison between the PSO & Dashti (2012) results

No.	The Results of Dashti's Research	The Research Result (This Paper)
1	358.39	364.07
2	651.07	669.99
3	956.67	964.51
4	1249.9	1262.81
5	2304.13	2313.45
6	2409.28	2499.79
7	2961.59	3006.31

As the results shown in table 3, the initial objective function values of the model (the maximum time of completion) resulted from the proposed PSO algorithm are included in the present study with a negligible difference of the corresponding values in Dashti research (2012). The main reason for this difference in this thesis is to consider the minimization of two opposing objectives at the same time. The difference between the maximum amounts of time completion in the research by Dashti (2012) is justified. According to the slight difference between the values of the two researches, it is suggested that if the proposed PSO algorithm is implemented in this study only to optimize the maximum amount of time or to implement into single objective, it will perform better than the proposed PSO in the field of research by Dashti (2012) and this indicates the validity of the proposed algorithm in this paper.

We have evaluated the performance of the two objectives in this study through PSO algorithm based on Pareto Archive, the results are compared with those of NSGA-II due to the quality, and dispersion and integration (refer to the research by Tavakoli Moghadam et al, 2011). It should be noted that the proposed NSGA-II algorithm has been used for solving multi-objective problems; however, some researchers have used the algorithm to solve the scheduling ones (Zhu 25 et al, 2014[4]). The results[5] comparison of two algorithms is shown in Table 4.

Table 4- Comparison between PSO & NSGA-II Algorithms

Prob.	PSO					NSGA-II				
	Quality metric	Spacing metric	Diversity metric	NOS	MID	Quality metric	Spacing metric	Diversity metric	NOS	MID
1	85.4	0.8694	633.2	65	1085.5	14.6	0.6601	333.01	71	1265.4
2	99.01	1.003	790.6	49	1183.5	0.989	0.8649	415.5	33	1842.54
3	100	0.7634	919.5	91	863.93	0	0.99	777.1	69	1536.65
4	100	0.9911	1092.3	78	689.4	0	0.4562	879.3	84	1003.67
5	79.5	1.3482	1213.7	53	1153.5	20.5	0.7941	906.6	63	1807.62
6	89.8	0.889	1609.4	86	794.09	10.2	0.7054	992.4	46	1659.43
7	79.3	1.11	3451.6	96	1031.7	20.7	0.63	1834.9	86	1741.62

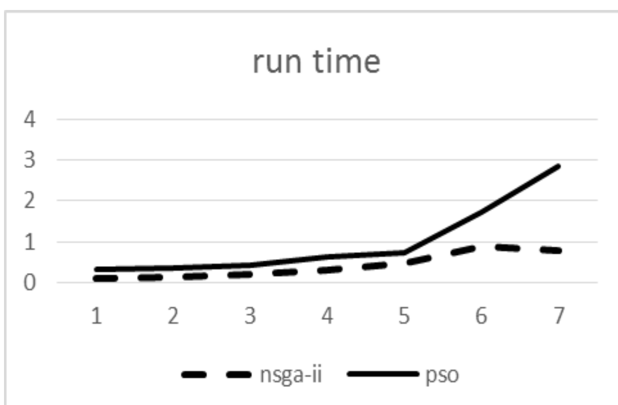

Figure 3- The Time Comparison between PSO & NSGA-II Algorithm

The results of the comparison in table (4) show that the PSO algorithm in all cases, has higher potentiality to produce more and high quality answers than the NSGA-II algorithm. The results also the distance index together with ideal answer, indicate that the answers in PSO algorithm have less difference with the ideal answer rather than NSGA-II algorithm. The

[4] Zhu
[5] dfd

PSO algorithm is capable of producing higher frequencies than the NSGA-II algorithm or PSO algorithm has more capability to explore and extract the feasible region than NSGA-II algorithm. As shown in the above table, the NSGA-II algorithm produces answers with higher uniformity and integration than PSO algorithm. Furthermore the comparison of performance time of the two algorithms suggests that the particle swarm optimization algorithm takes more time to solve the tasks scheduling problems than that of NSGA-II algorithm.

III. Conclusion

In this paper, an integrated proposed algorithm for tasks scheduling in cloud computing system has been presented to optimize the scheduling and allocation of the tasks in the cloud computing environment so that the objectives for the problems are considered optimal. The dependencies are assigned and considered to each other and classify them. The goals are as: minimization of the maximum amount of tasks' time completion, minimization of the total fines on tasks. The problem is based on its complex nature among the hard, uncertain and non-deterministic polynomials (NP-hard). Therefore the integration algorithm for optimization of swarm particles optimization on Pareto Archive basis has been proposed for solving the desired problem.

The results of the proposed PSO algorithm were compared with the NSGA-II algorithm in five indicators with the responses available in the Pareto Archive as well as comparing the distance from the ideal answer, quality, integration and dispersion of the multi-objectives issues and problem solving time criterion. The results show that firstly, this algorithm is more robust in all cases to produce high- quality answers with more dispersion than the NSGA-II algorithm, are stronger but weaker in integration and time criterion. In the other hand, the proposed algorithm is more capable of extracting regions of responding. In terms of other objective functions, considering penalties in phase or using other met heuristic approaches have been presented to resolve the desired issue as a future proposal for interested researchers.

References

[1] G. N. Gan, T. L. Huang, and S. Gao, "Genetic simulated annealing algorithm for task scheduling based on cloud computing environment", in Proc. Int. Conf. Intell. Comput. Integr. Syst.,pp. 60–63,2010.
[2] H. Liu, D. Xu, and H. Miao, "Ant colony optimization based service flow scheduling with various QoS requirements in cloud computing", in Proc. 1st ACIS Int. Symp. Softw. Netw. Eng., pp. 53–58, 2011.
[3] Huang L, Chen H, Hu T, "Survey on Resource Allocation Policy and Job Scheduling Algorithms of Cloud Computing", Journal of Software, pp. 480-487, 2013.
[4] Ismaila, L., Fardoun, A. (2016). EATS: Energy-Aware Tasks Scheduling in Cloud Computing. Procedia Computer Science 83 (2016) 870 – 877.
[5] M. Choudhary and S. K. Peddoju, "A dynamic optimization algorithm for task scheduling in cloud environment," Int. J. Eng. Res. Appl., vol. 2, no. 3, pp. 2564–2568, 2012.
[6] Koch, Fernando & Assuncao, Marcos & Netto, Marco. (2012). A Cost Analysis of Cloud Computing for Education. 182-196. 10.1007/978-3-642-35194-5_14.
[7] Salot, p. A Survey of Various Scheduling Algorithm in Cloud Computing Environment. Ijret: International Journal of Research in Engineering and Technology, Volume: 02 Issue: 02, 2013.
[8] J. GU, J. Hu, T. Zhao, and G. Sun, "A new resource scheduling strategy based on genetic algorithm in cloud computing environment," J. Computer, vol. 7, no. 1, pp. 42–52, 2012.
[9] Juarez, F., Ejarque, J., Badia, R.M. (2018). Dynamic energy-aware scheduling for parallel task-based application in cloud computing. Future Generation Computer Systems 78 (2018) 257–271.
[10] K. Zhu, H. Song, L. Liu, J. Gao, and G. Cheng, "Hybrid genetic algorithm for cloud computing applications", inProc. IEEE Asia-Pacific Serv. Comput. Conf., pp.182–187, 2011.
[11] Lavanya, M., Shanthi, B., Saravanan, S. (2019). Multi objective task scheduling algorithm based on SLA and processing time suitable for cloud environment. Computer Communications, Volume 151, 1 February 2020, Pages 183-195
[12] Li, J. (2020). Resource optimization scheduling and allocation for hierarchical distributed cloud service system in smart city. In Press, Journal Pre-proof.

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

[14] Sanaj, M.S., Joe Prathap, P.M. (2019). Nature inspired chaotic squirrel search algorithm (CSSA) for multi objective task scheduling in an IAAS cloud computing atmosphere. Engineering Science and Technology, an International Journal, Available online 22 November 2019.

[15] Sharma, M., Garg, R. (2020). An artificial neural network based approach for energy efficient task scheduling in cloud data centers. Sustainable Computing: Informatics and Systems, Available online 16 January 2020, 100373

[16] T. D. Braun, H. J. Siegel, N. Beck, L. L. Boloni, M. Maheswaran, A. I. Reuther, J.P. Robertson, M. D. Theys, B.Yao, D. Hensgen, and R. F. Freund, "A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems", Journal of Parallel and Distributed Computing, vol. 61, issue 6, pp. 810-837, 2001.

[17] T. Jenifer Nirubah. R. Rani John , "A Survey of the Impact of Task Scheduling Algorithms on Energy-Efficiency in Cloud Computing", International Journal of Engineering Research & Technology (IJERT)Vol. 3 Issue 1, pp.1284-1291,2014.

[18] Plestys, R., Vilutis, G., Sandonavicius, D., "The Measurement of Grid QoS Parameters"; Proceedings of the ITI 2007; 29th Int. Conf. on Information Technology Interfaces, Cavtat, Croatia, June 25-28 2007.

[19] He, X., Sun, X-He, Laszewski, G.V., "QoS Guided Min- Min Heuristic for Grid Task Scheduling", Journal of Computer Science and Technology 18(4), pp. 442-451, 2003

[20] Zhao Tong, Hongjian Chen, Xiaomei Deng, Kenli Li, Keqin Li, A Scheduling Scheme in the Cloud Computing Environment Using Deep Q-learning, Information Sciences (2019), doi: https://doi.org/10.1016/j.ins.2019.10.035

[21] Sharma, M., Garg, R. (2020). HIGA: Harmony-inspired genetic algorithm for rack-aware energy-efficient task scheduling in cloud data centers. Engineering Science and Technology, an International Journal, Volume 23, Issue 1, February 2020, Pages 211-224.