Current Trends in Technology and Sciences
ISSN : 2275-0535
9th Symposium on Advances in Science & Technology (9thSASTech 2014), Mashad, Iran

The New Approach for Solving Task Scheduling in Cloud Computing Environment Using Combination of Genetic Algorithm and Tabu Search

Fereshte Habibi
Department of computer Engineering , Sari Branch, Islamic Azad University, Sari, Iran, Fereshtehabibi42@yahoo.com

Dr. Homayon Motameni
Department of computer Engineering, Sari Branch, Islamic Azad University, Sari, Iran ,motameni@iausari.ac.ir

Farhad Ramezani
Department of computer Engineering ,Sari Branch, Islamic Azad University, Sari, Iranramezani.farhad@gmail.com

***Abstract* — Taskscheduling problem in cloud computing environment is an important challenge that has a direct effect on quality of service provided therein. Task scheduling includes the process of tasks mapping to available resources based on task requirements and features. Task scheduling problem in cloud computing is a very important problem and included in Np problems that specifies an optimal schedule for implementation of tasks and optimal allocation of resources so that within shorter time, more task can be processed. Various algorithms have been offered so far for solving task scheduling problem in cloud computing. In this paper, a combined algorithm was presented for improvement of task scheduling problem in infrastructure layer of cloud computing, based on evolutionary genetic algorithm as well as tabu search algorithm. Empirical results indicated that the offered algorithm has a high efficiency and created a balance between users' criteria.**

***Keyword* — Cloud Computing, Genetic Algorithm, Tabu Search Algorithm,Task Scheduling.**

1. INTRODUCTION

Upon increasing development of information technology and increase of various applications, doubtless need to integrated calculations is necessary for more users. Considering various applied needs of users, varied processing processes are required. In case these processes are performed based on more completely developed method, then need frequent and varied processors, but it is not possible for plenty of users. Therefore, use of technologies such as cloud computing that upon accepting the users' need to do computational processes of users and recover the obtained results, seems to be necessary. Hence, based on such perspective of software, users get needless of hardware and computational environments.

National Society of Standards and Technologies defines cloud computing as below:

"Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." [1].

In fact, cloud computing technology in infrastructure layer provides usability of computational resources and saving as a service in terms of need type and through making an abstract layer on all physical resources aiding virtualization, provides resources management dynamically.

Task scheduling is a key process in the cloud infrastructure layer that process of tasks mapping to available resources is formed based on tasks requirements and features and its goal is implementing inputted demands to system on the resources in an efficient procedure and assuming other cloud environment characteristics. In cloud computing environment, any user may face hundreds of virtual resources for performing any task. In this case, assigning tasks to virtual resources by the own user is impossible. Due to dynamic features of cloud computing and different needs of various applications of resources, tasks scheduling discussion is assumed as Np problem [2]. For this purpose, different algorithms have been presented so far including as below: FCFS [3], Round Robin [4], Min-min [5], Max-min [6], genetic [7], ant colony [8], simulated annealing [9], tabu search algorithm [10].

Among these methods, genetic algorithm was raised as the most well-known algorithm and used in different academic contexts such as engineering sciences as a powerful mean for solving optimization problems. Therefore, in this paper, the offered algorithm was compared to genetic algorithm.

In this paper, to solve task scheduling problem in cloud environment, combination of two genetic algorithm and tabu search algorithm was used. Genetic algorithm despite of high capabilities in problem Space search in stability has a weak performance in local search. Purpose of presenting the offered algorithm is combination of

general search capabilities of genetic algorithm to local search of tabu algorithm.
In this paper, in second part, offered architecture was explained. In third part, the offered algorithm was described and later in fourth part, summary of implementation was provided and fifth part includes conclusion.

2. Offered Architecture

There are N Data Centers In the presented cloud environment that each one of these Data Centers may be located at each point of geographical environment.
Each Data Center is comprised of a few virtual machines that are connected to their respective Data Center using communication canals.
Every virtual machine is comprised of a few processors for processing information with varied processing power and performance of virtual machines is also varied. In addition, virtual machines may have processors with varied performance.
Furthermore, each processor has different costs for data processing. Processors are connected to their respective virtual machine using communication canals. Ultimately, all of these Data Centers in cloud environment are connected to each other via communication canals.
In this environment, users that need different processors for their data processing, transmit their demands in a file for Scheduler system of cloud environment with varied transmission rates. In continue, schedulersystem of cloud environment receives demands of users and using scheduling algorithm deals with allocation of resources for users' tasks processing. Moreover, each user for processing all of his tasks determines a proposed end time and proposed cost. In addition, user can suppose a priority for the said two parameters. Purpose of this scheduling algorithm is processing and implementing users' tasks within minimum time and lower cost. After tasks scheduling by scheduling algorithm and their implementation in cloud system, processed data is transmitted to users.

2.1. Parameters assumed for offered architecture

2.1.1. Features of a Data Center

In the offered architecture, each one of Data Centers has following features:

- Band broadness usable in Data Center for receiving and transmitting data in cloud environment system;
- List of virtual machines provided in cloud environment by respective Data Center
- Geographical zone of Data Center

2.1.2. Features of a virtual machine

Each virtual machine in the offered architecture has following features:

- Operating system type of respective virtual machine
- Architecture of respective virtual machine such as Intel, IBM, Mac etc.
- List of available processors

2.1.3. Features of a processor

In the offered architecture, each one of processors has following features:

- Processing ability of respective processor based on MIPS
- Cost of users' tasks performed on respective processor for each second
- Rate of available processing ability by processor in Working days
- Rate of available processing ability by processor in holidays
- Rate of available processing ability by processor in other times such as midnight.

2.1.4. Users

In the offered architecture, users transmit their requests with following features for cloud environment system.

- List of tasks: Request of each user may include several tasks that each one may need a specific processor.
- Offered time for processing total tasks
- Offered price for processing total tasks
- Geographical zone of user
- Band broadness usable by users for receiving and transmitting data in cloud environment system
- Priority of "offered time" and "offered price" parameters that its value is within [0,1] range:
  Price priority + time priority = 1

2.1.5. Requested tasks

- Number of requested task Instructions of user based on MIPS
- Size of file that data related to user's respective request is saved therein.
- Operating system type required for user's requested task.
- Architecture type required for user's requested task.

2.2. Constraints assumed in this architecture

- Users' tasks are space-shared or offline and transmitted to scheduler system.
- There is no pause for processors, it means when a processor is allocated to a task, will be at the disposal of processor until end of its respective task processing.
- No constraint has been supposed for order of users' tasks processing.
- Any task must be processed on a processor with its related operating system and architecture.

3. Offered Algorithm

The proposed method is combination of strengths of two genetic algorithms and local tabu search for accessing to a more optimal algorithm. The said methods are combined to each other so that cover each other's weaknesses and result to a completely integrated and optimal algorithm. Considering items mentioned in previous sections, for better understanding of offered algorithm, its stages were implemented on cloud

environment of offered model as described in table (1). In this sample cloud environment, two Data Centers so that each Data Center comprised of two virtual machines with different capabilities, were used. Each machine uses two processors for processing users' tasks.

Table (1): Basic information of Data Center in offered cloud environment

Data Center information			Machine information			Processor information					
Data center name	Transmission rate	Geographical zone	Virtual machine name	Operating system type	Architecture type	Processor name	Throughput	Cost	Max.available throughput		
									Working day	Holiday day	Other times
D1	58	+03:30	M1	Windows	Intel	P1	10	1000	50%	90%	30%
						P2	25	2500	45%	50%	25%
			M2	Unix	IBM	P1	30	3000	56%	80%	60%
						P2	10	1500	70%	89%	50%
D2	70	+03:00	M1	Windows	IBM	P1	20	2000	30%	70%	40%
						P2	35	3000	60%	95%	15%
			M2	Unix	Intel	P1	15	1500	30%	90%	25%
						P2	40	4000	40%	90%	35%

In addition, basic information related to users' tasks is provided in table (2). In this table, there are 3 users that each one has transmitted specifications of his tasks for scheduler system of cloud environemnt as offline to be processed by processors availble in sample cloud environment.

Table (2): Basic information of users' task in offered cloud environemnt

Users information						Tasks information				
User name	Transmission rate	Offered price	Offered time	Time priority	Cost priority	Task No.	Task size	Instructions number	Operating system type	Architecture type
U1	30	1000	200	0.7	0.3	1	50	150	Windows	Intel
						2	35	80	Unix	IBM
						3	8	20	Windows	Intel
U2	45	9000	180	0.4	0.6	1	18	50	Unix	Intel
						2	25	60	Unix	IBM
						3	43	100	Unix	IBM
U3	60	1500	150	0.5	0.5	1	5	15	Windows	Intel
						2	20	55	Windows	IBM
						3	30	70	Unix	IBM

In offered algorithm, data coding was used for better identification of resources and processors related to table (1) and processors' indexing was used for coding the available data, so that in new coding, a unique index is assumed for each one of processors. Table (3) shows coding procedure of table (1). Each row of this table refers to respective processor in table (1).

Table (3): Coded processors in offered cloud environment

Row No.	Data center name	Machine name	Processor name
0	D1	M1	P1
1	D1	M1	P2
2	D1	M2	P1
3	D1	M2	P2
4	D2	M1	P1
5	D2	M1	P2
6	D2	M2	P1
7	D2	M2	P2

In addition in the offered algorithm, tasks coding table was used for better identification of tasks. Each row of

Current Trends in Technology and Sciences
ISSN : 2275-0535
9th Symposium on Advances in Science & Technology (9thSASTech 2014), Mashad, Iran

this table refers to one of tasks of respective user in table (2). Table (4) shows coding procedure of table (2) tasks.

Table (4): Coded tasks in offered cloud environment

Index No.	User name	Task No.
0	U1	1
1	U1	2
2	U1	3
3	U2	1
4	U2	2
5	U2	3
6	U3	1
7	U3	2
8	U3	3

Figure (1) shows general stages of offered algorithm.

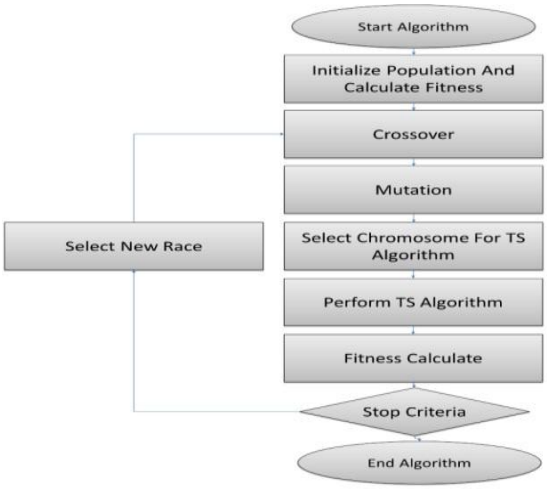


Fig.1. General stages of offered algorithm

3.1. Chromosome structure

In the offered algorithm, to exhibit chromosome, a one-dimensional array in the length of total available tasks for processing was used. Each cell of a chromosome that is so called a gene consisted of two fields that are shown as UT_i and DP_i. First field refers to i^{th} task of coded tasks table and second field DP_i denotes processor No. i in coded processors table. Figure (2) shows a sample chromosome assumed in offered algorithm.

2,0	6,1	8,7	0,0	1,3	7,4	5,2	3,6	4,3

Fig.2. Chromosome structure in offered algorithm

As it is observed in chromosome structure figure (2), each array number shows the number of task to be processed by respective processor. For instance content of second gene is 1,6 that in fact refers to task 6 of coded tasks table to be processed by processor 1 provided in coded processors table. Each chromosome shows a schedule for performance of all tasks of users.

3.2. Fitness

Fitness function in very important in evolutionary algorithms, because these algorithms work correctly only when this function has been defined correctly. This function is a benchmark to diagnose if a response is better or competent than other responses. Chromosome fitness in offered algorithm was assumed as reducing completion time of total available tasks for processing in scheduler system of cloud environment as well as fitness of all users for improving users' tasks processing so that their needs are met completely. Considering definitions and hypotheses explained in previous parts, to compute the fitness of a chromosome, at first fitness of total cost used for processing tasks of user j is calculated based on equation (1).

$$P_j = \frac{CT_j}{OP_j} * PC_j (1)$$

In above equation, OP_j denotes offered cost of user j for processing total respective tasks and CT_j denotes calculated cost in schedulersystem of cloud environment for processing tasks of user j and PC_j denotes priority assumed by user j for the cost and ultimately P_j denotes fitness of cost supposed for user j.

Later, based on equation (2), fitness of total time needed for processing total tasks of user j is calculated.

$$T_i = \frac{TT_j}{OT_j} * PT_j (2)$$

In above equation, OT_j denotes offered time of user j for processing his total respective tasks and TT_j denotes calculated time in scheduler system of cloud environment for processing tasks of user j and PT_j denotes assumed priority of user j for time and finally T_j indicates fitness of time assumed for user j.

Considerable equations (1) and (2), fitness assumed for user j is calculated based on equation (3).

$$\mathbf{F_j = P_j + T_j} \quad \mathbf{(3)}$$

In this equation, P_j denotes fitness of cost assumed for user j and T_j denotes fitness of time assumed for user j and F_j denotes fitness of user j.

Ultimately, fitness of chromosome in offered algorithm was assumed as total fitness of all users and completion time for processing all tasks in scheduler system of cloud environment that whatever chromosome fitness is lower, that chromosome will have better fitness. Equation (4) predicts chromosome fitness in offered algorithm.

$$\mathbf{Fitness = Makespan + \sum_{j=0}^{j} Fj \ (4)}$$

In equation (4), Makespan means completion time for processing total tasks of users in scheduler system of cloud environment.

3.3. Selection of parents for crossover and mutation

There are various strategies for selection of best chromosomes. In this algorithm, ranking method was used to select parents. Selection based on ranking is so that all chromosomes are ranked based on fitness as per equation (5). The best chromosome receives the rank of greatest fitness and the worst chromosome obtains rank 1.

$$\mathbf{Cr_{1<=i} = (Mux\text{-}Fit_i) + 1, i <= n \ (5)}$$

So, in this method of selection of parents, all chromosomes will have the chance of selection.

3.4. Crossover operator

In this algorithm, single point crossover operator was used. In this crossover method, after selecting two parents for crossover operation, one random number is selected within range 0 to n-1 (n denotes number of tasks) and genes located within range 0 to selected random number are transferred from first parent to the child and later its similar genes are deleted in second parent. Genes remained in second parent are inserted respectively in empty cells. Purpose of using this crossover method is making similar children of parents to can make better children of parents with better fitness. Experimental results indicated that upon using this crossover, chromosomes are dispersed in better space of problem that consequently results in better answer in better time. Figure (3) shows applying crossover operator for two selected chromosomes.

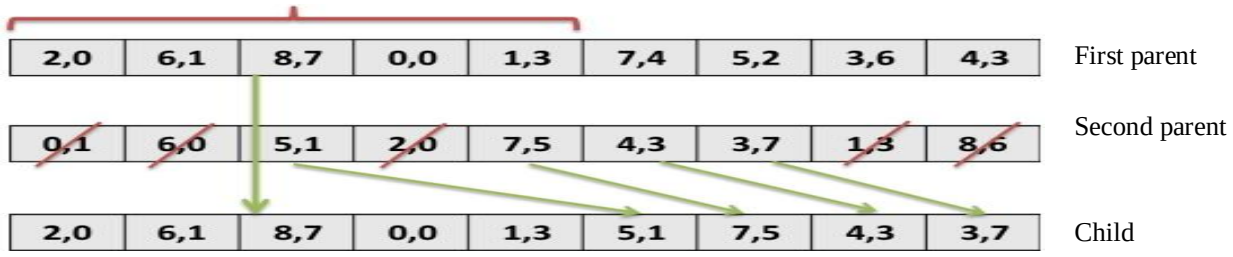


Fig.3. Applying crossover operator for to chromosomes in offered algorithm

3.5. Mutation operator

Mutation causes search in intact spaces of problem. In this algorithm, exchange method was used for mutation so that after selection of a parent chromosome, 1 gene was selected randomly and its processor field is changed. Main purpose of applying this type of mutation is exchanging processor parameter of a task in order that a task can be processed in a better processor. Consequently, each chromosome obtains a better answer in problem space. Figure (4) shows applying mutation operator.

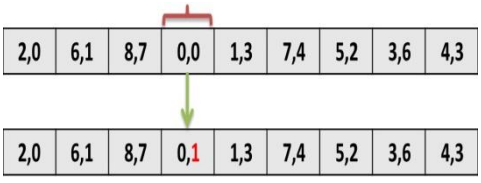


Fig.4. Applying mutation operator in offered algorithm

3.6. Population optimization

At this stage, before selecting chromosomes for transferring to the next generation using tabu search algorithm, a rate of available population chromosomes are optimized and added as new children to the population.

In the offered algorithm, in an iteration ring, a few chromosomes are selected based on ranking method and local search algorithm is calling for them. Local search algorithm in each iteration receives one of genetic chromosomes as current solution and upon applying its operators, optimizes chromosomes during several stages.

Neighborhood in tabu search of offered algorithm has been defined based on displacement of 2 genes and one of processors selection modes, so that total probable displacements of a chromosome's genes and one of processor selection modes are recorded in a list called tabu list. Selection of 2 genes and one of processor selection modes for displacement was made randomly according to tabu list. Later, this movement is deleted from tabu list to avoid selections of repeated neighborhoods. Number of neighborhoods selection is in accordance with defined number in order to optimize chromosome as possible.

3.7. Selection of chromosome

Chromosomes selection action for next generation in offered algorithm is a combined selection. In this method, at first chromosomes are sorted based on fitness and repeated chromosomes are deleted. Later, a rate of chromosomes with better fitness and remaining chromosomes are selected randomly for population of next generation. In this selection method, chromosomes dispersal will be kept always and chromosomes will be dispersed in problem space, and also causes premature prevention of local optimum points.

3.8. Termination condition

Termination condition in offered algorithm is limiting generation number, it means if algorithm reached to the maximum respective generation, best chromosome is exhibited and algorithm is completed, otherwise stage 2 is repeated.

4. Assessment of Algorithm

Programming language C#.Net 2012 was used for implementation of offered algorithm. Algorithm was implemented on a computer with processor Intel(R) Pentium(R) 4 CPU 3.00GHz and ram 2GB. In addition, results obtained from offered algorithm simulation were compared to genetic algorithm as the most widely used algorithm for solving scheduling problems. To test the

offered algorithm efficiency, 8 data Collection of test were designed to cover small, middle and big cloud environment systems. Each test was named test N_Day. In this naming method, N denotes test number and Day shows working day that this test Collection will be processed in cloud environment. Table (5) shows designed test data.

Table (5): Collection of designed test data

Designed test No.	Designed tests specifications											
	Users specifications							Specification range of each task				
	User number	Total task	Offered cost	Transmission rate	Offered time	Time priority Cost priority	Geographical zone	Task number of each user	Task size	Instructions number of Each task	Operating system type	Architecture type
Test 01-H	68	296	[1000,2000]	[425,550]	[100,300]	[0,1]	ALL GMT	[2,7]	[600,750]	[2500,3000]	[1,6]	[1,4]
Test 02-W	48	262	[1000,2000]	[425,550]	[100,300]	[0,1]	ALL GMT	[2,7]	[600,750]	[2500,3000]	[1,6]	[1,4]
Test 03-W	69	319	[1000,2000]	[425,550]	[100,300]	[0,1]	ALL GMT	[2,7]	[600,750]	[2500,3000]	[1,6]	[1,4]
Test 04-H	34	155	[1000,2000]	[425,550]	[100,300]	[0,1]	ALL GMT	[2,7]	[600,750]	[2500,3000]	[1,6]	[1,4]
Test 05-H	52	244	[1000,2000]	[425,550]	[100,300]	[0,1]	ALL GMT	[2,7]	[600,750]	[2500,3000]	[1,6]	[1,4]
Test 06-W	59	279	[1000,2000]	[425,550]	[100,300]	[0,1]	ALL GMT	[2,7]	[600,750]	[2500,3000]	[1,6]	[1,4]
Test 07-H	31	131	[1000,2000]	[425,550]	[100,300]	[0,1]	ALL GMT	[2,7]	[600,750]	[2500,3000]	[1,6]	[1,4]
Test 08-H	46	209	[1000,2000]	[425,550]	[100,300]	[0,1]	ALL GMT	[2,7]	[600,750]	[2500,3000]	[1,6]	[1,4]

Whereas each test of table (5) needs a model of cloud environment to process total tasks assumed in each test by processors situated in cloud environment, collection of a model of cloud environment was designed using parameters expressed in previous parts. This collection of model was called Model N_GMT that N denotes number of a model that designed for test N of table (5) and GMT denotes geographical zone assumed for scheduler center of cloud environment. Table (6) shows collection of model designed for test data of table (5).

Table (6): Collection of designed model

Designed model No.	Specifications of designed model for each test data											
	Specifications rangeof each Resource			Specifications rangeof each Machine			Specifications rangeof each Processor					
	Resource number	Transmission rate	Geographical zone	Machine number	Operating system type	Architecture type	Processor number	Throughput	Cost	Max range of available throughput		
										Working day	Holiday day	Other time
Model 01_+06:30	2	[600,800]	All GMT	3	[1,6]	[1,4]	6	[60,75]	[120,150]	[55,65]	[55,65]	[60,75]
Model 02_-12:00	2	[600,800]	All GMT	6	[1,6]	[1,4]	15	[60,75]	[120,150]	[55,65]	[55,65]	[60,75]
Model 03_+02:00	2	[600,800]	All GMT	3	[1,6]	[1,4]	9	[60,75]	[120,150]	[55,65]	[55,65]	[60,75]
Model 04_+12:00	1	[600,800]	All GMT	3	[1,6]	[1,4]	5	[60,75]	[120,150]	[55,65]	[55,65]	[60,75]
Model 05_+03:30	2	[600,800]	All GMT	6	[1,6]	[1,4]	7	[60,75]	[120,150]	[55,65]	[55,65]	[60,75]
Model 06_+03:30	1	[600,800]	All GMT	3	[1,6]	[1,4]	5	[60,75]	[120,150]	[55,65]	[55,65]	[60,75]
Model 07_+05:00	1	[600,800]	All GMT	2	[1,6]	[1,4]	4	[60,75]	[120,150]	[55,65]	[55,65]	[60,75]
Model 08_+12:00	1	[600,800]	All GMT	3	[1,6]	[1,4]	5	[60,75]	[120,150]	[55,65]	[55,65]	[60,75]

Figure (5)shows a comparison between implementation time of results obtained from offered algorithm and genetic algorithm implementation on test data of table (5). As it is observed, offered algorithm reaches to the result within longer time and its cause may be local search on the problem space.

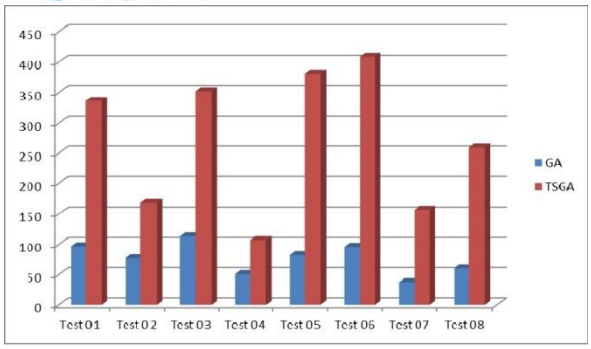


Fig.5. Implementation time of results obtained from offered algorithm and genetic algorithm implementation

Figure (6) shows comparison between fitness diagram resulted from offered algorithm and genetic algorithm implementation for data of Test 02_w. As shown in figure (6), offered algorithm always in most cases could perform better than genetic algorithm.

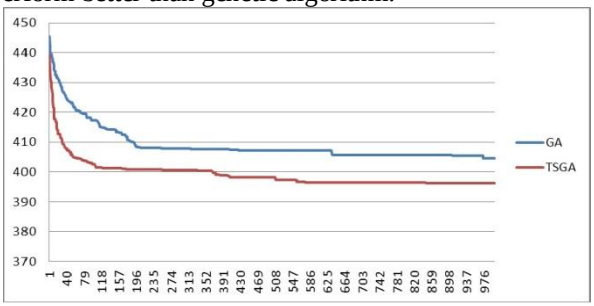
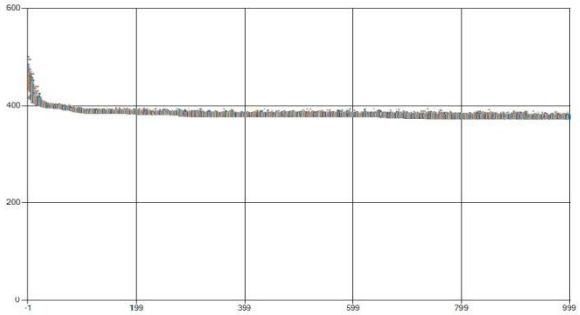


Fig.6. Comparison of fitness chart of offered algorithm to genetic algorithm

Figure (7) shows dispersal diagram resulted from offered algorithm and genetic algorithm implementation for data of Test 07_w. According to this figure, in offered algorithm, population dispersal is always kept and more optimal space of problem was searched and use of suitable genetic operators resulted in lack of premature convergence of responses.



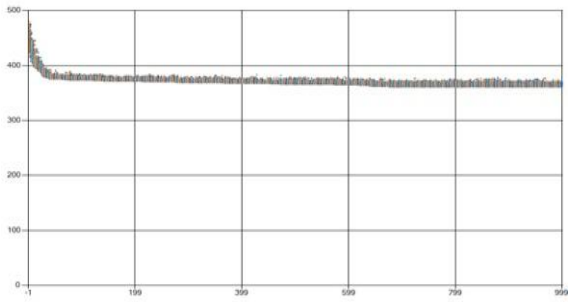


Fig.7. Dispersal diagram resulted from implementation of offered algorithm and genetic algorithm

Figure (8) Comparison of Gantt diagram shows sample scheduling for several samples of test data of table (5).

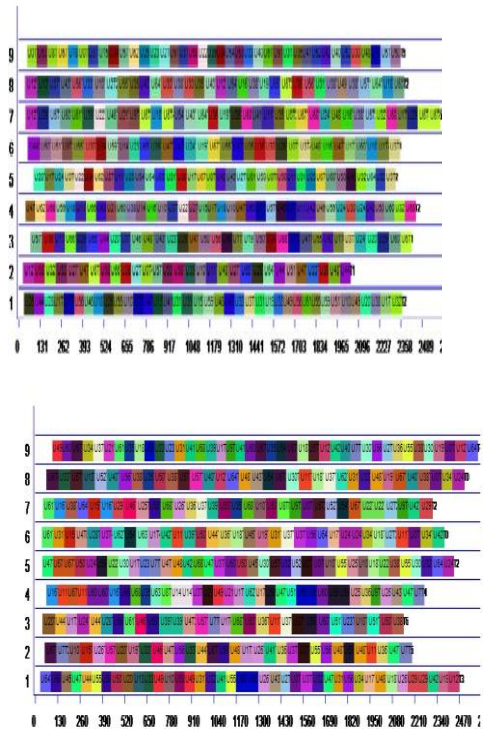


Fig.8. Gantt chart diagram, resulted from offered algorithm implementation for several test samples

In consideration of obtained results, a comparison may be provided between genetic algorithm and offered algorithm as below:

Current Trends in Technology and Sciences
ISSN : 2275-0535
9th Symposium on Advances in Science & Technology (9thSASTech 2014), Mashad, Iran

Table (7): Summary of offered algorithm and genetic algorithm implementation

Test No.	Best generation		Implementation time		Best implementation time		Best fitness		Mean fitness		Make span	
	GA	TSGA	GA	TSGA	GA	TSGA	GA	TSGA	GA	TSGA	GA	TSGA
Test 01-H	946	995	97	336	91	334	1088.60	1020.60	1119.92	1054.65	3715	3802
Test 02-W	974	847	77	169	79	199	404.60	396.20	409.78	406.31	1384	1426
Test 03-W	998	983	113	352	113	346	856.60	836.80	871.92	851.22	2602	2578
Test 04-H	494	537	51	107	48	58	449.00	453.80	461.75	464.62	3825	3802
Test 05-H	995	900	82	381	81	345	899.20	876.00	919.28	908.41	5352	5316
Test 06-W	342	761	96	409	34	311	1502.80	1482.80	1551.92	1533.98	8977	8957
Test 07-H	901	673	38	157	35	105	374.20	362.20	385.81	374.58	2511	2599
Test 08-H	690	603	60	259	41	155	915.40	920.40	942.20	944.05	5284	5303

5. CONCLUSION

Upon developing cloud environment and increasing diversity of requests, a system based on cloud processing is needed so that to respond the different demands more suitable and rapider. In this paper, a new method was presented using combination of genetic algorithm and tabu search algorithm for solving task scheduling problem in cloud environment. Empirical results indicated that offered algorithm has higher efficiency for solving tasks scheduling problem in cloud environment and could create a balance between users' criteria. Purpose of balance is that in the offered algorithm, satisfaction of all users to be attracted and to process the tasks on suitable processors in order to process more tasks within shorter time in cloud environment.

REFERENCE

[1] Peter Mell and Timothy Grance, "The NIST Definition of Cloud Computing", Recommendations of The National Institute of Standards and Technology, September 2011.

[2] Hong Sun, Shi-ping Chen, Chen Jin, Kai Guo,"Research and Simulation of Task Scheduling Algorithm in Cloud Computing", TELKOMNIKA Indonesian Journal of Electrical Engineering, Vol.11, No.11, pp. 6664-6672, 2013.

[3] R. P. BRENT, "Efficient implementation of the First-Fit Strategy for Dynamic Storage Allocation", ACM Transaction on programming Languages and Systems, Vol. 11, No. 3, pp.388-403, 1989.

[4] Daniel Nurmi, Rich Wolski, Chris Grzegorczyk, Graziano Obertelli, Sunil Soman, Lamia Youseff, Dmitrii Zagorodnov, "The Eucalyptus Open-source Cloud-computing System", presented at the IEEE International Symposium on Cluster Computing and the Grid(CCGrid'09), 2009.

[5] Tracy D. Braun, Howard Jay Siegel, Noah Beck, Ladislau L. Boloni, Muthucumaru Maheswaran, Albert I. Reuther, James P. Robertson, Mitchell D. Theys, Bin Yao, Debra Hensgen, Richard F. Freund, "A Comparison of Eleven Static Heuristic for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems", Journal of Parallel and Distributed Computing, Vol.61, pp.810-837, 2001.

[6] Upendra Bhoi, Purvi N. Ramanuj, "Enhanced Max-min Task Scheduling Algorithm in Cloud Computing", International Journal of Application or Innovation in Engineering &Management (IJAIEM), Vol.2, Issue.4, pp.259-264, April 2013.

[7] Pardeep Kumar and Amandeep Verma, "Scheduling Using Improved Genetic Algorithm in Cloud Computing for Independent Tasks", International Conference on Advances in Computing Communications and Informatics (ICACCI), 2012.

[8] Xiangqian Song, Lin Gao, Jieping Wang, "Job scheduling based on ant colony optimization in cloud computing", International Conference on Computer Science and Service System(CSSS), pp.3309-3312,2011.

[9] Monir Abdullah and Mohamed Othman, "Simulated Annealing approach to cost-based multi- quality of service job scheduling in Cloud Computing enviroment", American Journal of Applied Sciences, Vol.11, No.6, pp.872-877, 2014.

[10] I. De Falco, R. Del Balio, E. Tarantino and R. Vaccaro,"Improving search by incorporating evolution principles in parallel tabu search", IEEE Conference on Evolutionary Computation,Vol.2, pp.823-828,1994.

AUTHOR'S PROFILE

Fereshte Habibi, M.Sc. of Computer Engineering.