International Journal of Research in Advent Technology, Vol.6, No.5, May 2018
E-ISSN: 2321-9637
Available online at www.ijrat.org

A Tabu Stochastic Diffusion Search Based Fuzzy Scheduling in Cloud

Ranga swamy Sirisati[1], Dr.Sridhar Mandapati[2]
Research Scholar, Department of Computer Science & Engineering, Acharya Nagarjuna University, Guntur, Andhra Pradesh, India[1]
Associate Professor, Department of Computer Applications, RVR & JC College of Engineering, Guntur, Andhra Pradesh, India [2]
Email: sirisatiranga@gmail.com[1], *mandapati12@gmail.com*[2]

Abstract-A novel paradigm of Information Technology (IT) which can deliver computer resources including applications and data as a service over the internet to its users is referred to as cloud computing. Cloud computing's objective is to provide trustworthy, dynamic and virtual services, by providing resources for knowledge sharing, storage and computation. Scheduling is an essential requirement in cloud computing wherein the jobs are executed with certain constraints or metrics. Scheduling in the cloud environment is a Non-deterministic Polynomial (NP)-hard problem. These types of scheduling problems are solved using various optimization algorithms. This work presents a Tabu Search (TS) algorithm for presenting an approach to fuzzy rule base design for cloud scheduling. The parameter and the structure of the fuzzy rule base are solved using the TS. The performance of the proposed method is enhanced as the TS along with a symmetric neighbourhood structure is used for determining the fuzzy rule based parameters. For decreasing the effect of the TS problems, this work presents TS based on Stochastic Diffusion Search (SDS) called (Tabu-SDS). Greater diversity is provided to the candidate solutions of the TS using the suggested algorithm. Tabu SDS fuzzy scheduling is an extension of the fuzzy concept in the form of metaheuristic. It has been experimentally proven that compared to the existing techniques, the suggested technique achieves better performance.

Index Terms- Cloud Computing, Scheduling, Cloud Services, Fuzzy Rule, Tabu Search (TS) and Stochastic Diffusion Search (SDS).

1. INTRODUCTION

There has been a lot of attention given to cloud computing as a computing model for different types of application domains. Users can lease computing resources in the form of Virtual Machines (VMs) using cloud computing services. They can avail these from large scale data centres that are driven by service providers. A wide variety of applications can be dynamically deployed using cloud services and also this can be service on-demand. There are basic aspects of cloud servicing: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). Machine virtualization is used by most Cloud Service Providers (CSP) to provide flexible as well as cost effective sharing of resources. Managing the cloud computing resources efficiently is the onus of the CSPs, so that the required resources are available to the cloud users on demand. Thus, whenever needed, consumers across the world can access applications and data from the cloud anywhere [1].

There may be several thousand virtual resources that may be accessed by users in cloud computing; thus, allocation of tasks manually is impossible for everyone. The VM layer has the onus of task scheduling using the resources virtually, due to commercialization and virtualization .Hence, scheduling plays a significant role in assigning resources to each task effectively and efficiently. Different parameters that are used in cloud computing are as follows: **Makespan**: The time between the start and the completion of a sequence of a schedule .**Resource Cost**: Determined by the time occupied by a resource and the capacity of the resource. The more powerful a resource is, the greater is its cost. **Scalability**: It is the ability to cope up with the increased workload and to add resources correspondingly. **Reliability**: The capability with which, despite failure, the system continues to function. **Resource Utilization**: This parameter is used for defining how a system can use its resources effectively [2].

The process by which resource is assigned on the basis of budget constraints for the required tasks in the cloud is referred to as scheduling. This assists the files to be uploaded and downloaded during the transmission process. The purpose of scheduling is to enhance the utilization of resources. For avoiding the application to be hanged and to allow effective load distribution, scheduling is a must; without scheduling many errors can take place [3]. A critical component of cloud resource management is job scheduling. This process determines the way in which jobs have to be executed in terms of resource mapping and the time of execution. Scheduling takes care of multiplexing and resource sharing at various levels. For instance, a

server can be shared among many VMs, each VM may support many applications, and each application may consist of many threads. The objective of scheduling in the cloud is to enable the resources to be allocated to jobs and scheduled in such a manner that the jobs are completed in minimal time and cost with maximum user satisfaction and throughput. Broadly speaking, a user is expected to finish the job at hand with minimal time as well as cost [4].

Cloud computing is on-demand service; here, as per the demand of the client, resources, data, package and alternative devices are provided at the precise time. An NP-hard optimization problem is cloud task scheduling; there are several metaheuristic algorithms that have been suggested for solving the same. When virtualized clusters are utilized for executing several tasks on the cloud computing platform, the problem becomes even more challenging and complex. This is the reason why from several low level execution tasks in multiple processors to high level tasks in grid and cloud environments there are several heuristics that have been suggested. The recently published works make use of evolutionary algorithms such as the ant colony, bee colony and Particle Swarm Optimization (PSO) for optimization problems [5].

For satisfying the requirements of the users, some of the traits of cloud computing can be used for scheduling resources. It is possible to predict the needs of the user as the user's demand for cloud computing resources normally shows a long and continuous trait. The demand for the resources can be predicted, on the basis of the feature of the moving average technique. However, in case of resource scheduling, the fuzzy logic theory can be used for solving the extra difficulty that is caused by the imprecise user needs. By means of intense logical reasoning, fuzzy logic can model the human mind and construct a "human friendly system" between the cloud computing system and its users. Feedback control can be used for optimizing the dynamic scheduling of resources. It can also dynamically adjust the configuration on the basis of the performance of the present resource and the needs of the user as well [6].

The use of Annealing (SA) and thermodynamic SA has been evaluated by Moschakis&Karatza [7]. This can serve Bag-of-Tasks applications by scheduling a multi-cloud system that is dynamic and has VMs of varied performance. A Moving Average-based Fuzzy Resource Scheduling (MV-FRS) method for virtualized cloud environment has been suggested by Priya&Babu [8]. This can be utilized in the virtual cloud environment for optimizing the resource scheduling by means of VMs. Using the new type of architecture with dynamic queues that are based on a hybrid algorithm that uses Fuzzy Logic and Particle Swarm Optimization algorithm (TSDQ-FLPSO), a task scheduling was proposed by Alla et al., [9]. This can optimize the waiting time as well as makespan. A Cloud Theory-based Particle Swarm Optimization (CTPSO) algorithm has been proposed by Ma &Xu [10]. It can solve a variant of the vehicle routing problem that is referred to as Multiple Decision maker Vehicle Routing Problem with Fuzzy Random Time Windows (MDVRPFRTW). For dealing with cloud service composition issue, Xu& Sun [11] presented a Fuzzy operator based Bat algorithm (FBAT). In this, the original real number coding and decoding operations in the bat algorithm are replaced using the fuzzy encoding and decoding operations.

Alkhashai&Omara [12] introduced and implemented an enhanced task scheduling algorithm to assign the users' tasks to multiple computing resources. The aim of the proposed algorithm is to reduce the execution time, and cost, as well as, increase resource utilization. The proposed algorithm is considered an amalgamation of the PSO, the Best–Fit (BF), and TS algorithms (called BFPSOTS). According to the proposed BFPSOTS algorithm, the BF algorithm has been used to generate the initial population of the standard PSO algorithm instead of to be random. The trap of the local optimality is evaded by the TS algorithm. The proposed hybrid algorithm (i.e., BFPSOTS) has been implemented using Cloudsim. The implementation results prove that the proposed hybrid algorithm (i.e., BFPSOTS) outperforms the standard PSO algorithm. Section 2 explains various proposed methods. Section 3 discusses the results obtained in detail, and section 4 concludes the work.

2. METHODOLOGY

For relatively known nonlinear systems, the rule based fuzzy modelling is used as an authoritative mechanism. Some of the complicated mechanism leads to bad result such that fuzzy mechanism can be applied to merge information from different sources for identifying the system uncertainty. After the credentials of fuzzy rule based methods, the fuzzy controllers addresses the genuine control engineering problem like stability, robustness etc. [13]. Documenting an optimized fuzzy rule base is the difficult process while designing fuzzy models. Though the traditional fuzzy rule design does not find optimal fuzzy rules and membership functions, it is built based on human operator's experienced knowledge. In this section, TS algorithm, Tabu fuzzy and Tabu SDS fuzzy algorithm are discussed.

2.1. *Tabu Search (TS) Algorithm*

In order to overcome the local optimality issue in the process for solving combinatorial optimization problems, TS algorithm was suggested by Glover as

an intelligent optimization technique. Neighbourhood mechanism has been employed for search. The set of all the formations that can be obtained by a move is the neighbourhood of a solution. The process which changes the search from the present solution to its neighbouring solution is referred to as move. The reverses of the last move are not allowed in order to avoid a move from coming back to a solution that has been recently visited. These moves are regarded as "tabu" and are recorded in a list referred to as Tabu list. Hence, explicit memory is used for recording the search history. Initially, the Tabu list is empty. Then, it is built in subsequent search iterations and in later iterations it is circularly updated [14].

An admissible move is a move that is not present in the Tabu list. After a move is made, if the solution through this move is better than the solutions obtained by the prior iterations, then this solution is the new best solution. Based on Tabu condition and valuation principles, the next solution is identified. By making use of the Tabu conditions or constraints on the possible moves, it avoids the solutions previously obtained to be regenerated. Frequency and recent memory are the two factors on which the Tabu conditions are based. Aspiration mechanism is another important element in the TS. In case a Tabu list move results in a better solution than before then, this move gets out of the Tabu list. This property is used for preventing the removal of goo moves from consideration. It has a crucial role to play in the process of search. Below shows the pseudo code of TS [15]:

Begin
Set the parameters; $i = 1$;
Initialize a path at random
clear up the tabu list
while condition is not satisfied Do
Calculate the so_far_bestpath's neighborhood;
Bestpath; *so_far_bestpath*;
update tabu list;
$i = i + 1$;
End

2.2. *Tabu Fuzzy*

Combinatorial optimization problems can be easily solved using TS. Nonetheless, the quality of solutions that are obtained by TS depends heavily on the initial parameters such as the selection of neighbourhood (trial solutions) and the Tabu list size (or Tabu Period (TP)). The search in the TS procedure is guided by two critical parameters. The constants for the entire search procedure are held by these two parameters. Yet, in many of the cases, this approach is not as effective. For utilizing TS in case of practical problems, the Tabu list size is tough to determine. As there was no general method for selecting it, it was experimentally selected in most literature. The balance between growth and variation is to be maintained by a well-organized TS method [16].

There should be different TP associated with a "good" and a "bad" solution, and this has been known from experiments. The "distance" controls the diversification as well as the intensification. The TS may lose the balance between intensification and diversification if there is a fixed setting of distance between current as well as trial solutions through the run. This may lead to the production of sub optimal solution. Hence a search procedure must be employed for adapting these TS parameter settings. It is tough to find algorithms for obtaining optimal adaptive parameter setting as the interaction between the TS parameter setting and the TS performance is complicated and unexplored.

The result obtained for the assignment problem is stated as a vector x of dimension N signifies the N positions whose ith element x_i is a number denotes the entity given to the ith position. The objective function is represented as in equation (1):

$$f(x) = \sum_{i=1,\, j=x_i}^{N} c_{ji} \qquad (1)$$

The Potential Value (PV) is used for measuring the "goodness" degree of a solution. The possible solutions are measured by the membership values of f (x), to the best solution f (x^*).

Subsequently, for n iteration, the degree of membership of the present solution x to the best solution x^*, is evaluated by a membership function $\mu(x)$. This will map a solution x to the unit interval ranges as [0, 1].

$$\mu(x) = 1 - [f(x) - f(x^*)] / R(n) \qquad (2)$$

Where f (x^*) is the optimal (minimum) function value and R(n) meant the range of all function values created for iteration n. Equation (2) is used to evaluate the location of f (x) on the range scale. So the solutions with functional values become closer to the best value that takes higher membership value.

The equation (2) is altered as equation (3) when the minimum function value is indefinite:

$$\mu(x)=\begin{cases}1-[f(x)-f(x^{'}(n))]/R(n), & f(x)\geq f(x^{'}(n))\\ 1 & , f(x)<f(x^{'}(n))\end{cases} \quad (3)$$

Where $f(x^{'}(n))$ meant the best functional value of iteration n. Equation (3) shows the membership as f (x) is gained better than the best functional value. After measuring the membership function value, it let PV (x) = $\mu(x)$.

It is not clear how the parameters should be changed during the run as the process for varying the "distance" and the TP between the current and trial solutions for obtaining superior performance is not known. It has been observed from experiments that both lower TP and "distance" between the current and the trial solutions are preferred when the potential value of a solution is higher. It is imperative for both the TP and "distance" between the current and trial solutions to be enhanced; this is because when the best value function is stuck for a long period at the same value, the TS is frequently searched in the local vicinity of a local minimum for a long period and in this case, the TS should focus on exploiting than exploring. As per this, a fuzzy system is developed that adjusts the "distance" and the TP between the current and the trial solutions (DS) with TP and DS as outputs and PV and the Number of Non-improved Iterations (NNI) as the input variables .Simplifying, there are three fuzzy sets: low, medium and high that are associated with each variable. The problem to be solved determines the definition of fuzzy sets.

For adjusting the TP and DS there are six fuzzy rules that are used. The linguistic description for the six rules has been given below for clarity:

1. If PV is low, then TP is high and DS is high.
2. If NNI is high, then TP is high and DS is high.
3. If PV is medium and NNI is low, then TP is low and DS is low.
4. If PV is medium and NNI is medium, then TP is medium and DS is medium.
5. If PV is high and NNI is low, then TP is low and DS is low.
6. If PV is high and NNI is medium, then TP is medium and DS is medium

2.3. *Proposed Tabu SDS Fuzzy*

Based on a simple interaction of the agents, the SDS is a multi-agent global search and optimization algorithm. The process through which SDS allocates resources has been demonstrated by presenting a high level description of SDS in the form of a social metaphor. To solve best-fit pattern recognition and matching problems, SDS has introduced a novel probabilistic approach. SDS is a distributed mode of computation.SDS has a strong mathematical framework unlike many of the nature inspired search algorithms; the framework is used for delineating the behaviour of this algorithm by looking into the allocation of resources, convergence to global optimum, linear time complexity, minimal convergence criteria and robustness [17].

Firstly, the population is initialized when the SDS algorithm begins a search or optimization. Every agent, in an SDS search, maintains a hypothesis h; this is used for defining a likely solution to the problem. Test phase and diffusion phase are the two phases that follow the initialization. In the former, a partial hypothesis is evaluated by the agent that returns a Boolean value. Thus, SDS checks whether the agent hypothesis is successful or not based on the accurate recruitment approach used later in the iteration, across the population successful hypotheses diffusion takes place; In this manner, the information regarding solutions that are potentially good spreads across the entire population of agents. Every agent in the test phase performs partial function evaluation, pFE. This represents some function of the hypothesis of the agent pFE = f (h). Every agent in the diffusion phase recruits another agent that interacts and has potential to communicate the hypothesis.

Usually, in many of the heuristic search algorithms, it is a challenge to assure the existence of optimal solutions. The next biggest challenge is the local minimum or maximum. This is why the heuristic search algorithms are continuing to be developed. This work strives to improvise the TS' performance by employing SDS; SDS, in turn, provides greater diversity to TS' candidate solutions. When no more good solutions are there in TS, SDS will bring in the TS. SDS works with best solutions to provide a good range of result for the candidate solutions of Tabu search. This is done by interchanging the old solutions of Tabu list with the new solution of SDS. All iterative examination method should be in some instance to accept the unaltered moves from i to j in V^* (i.e. f (j) > f (i)). SDS aids in avoiding the local minimum. This version that has been suggested is more heuristic and stronger; it helps reduce the local minimum problem and obtain optimal solution [18].The suggested Tabu SDS is given in the following steps:

Step 1: Choose an initial solution i in S. Set i^* =i and k=0.

Step 2: Set k=k+1 and generate a subset V^* of solution in N (i, k).

Step 3: Choose the best j in V^* and set i = j.

Step 4: Select best subset from N (i, k) add in B.

Step 5: If the best solution is not obtained then call the SDS with best subset from Tabu list.

Step 6: Select the best solutions from SDS output and include it to Tabu list.

Step 7: If a stopping state is met stop the process else go to Step 2.

Where, the existing best solutions list is represented by B and this contains the best neighbours of V^*; Hence, when the algorithm's behaviour lags behind its expected value, the algorithm can recover to its best previous states. The meaning of the update step B is deleting the neighbours that have been used and then rearranging the others.

Tabu SDS fuzzy scheduling is an extension of the fuzzy tabu evaluation concept in the metaheuristic form. An iterative procedure which is population based is the technique that has been suggested. The beginning of the process is the initial population. For this generation, the construction phase of this SDS can be used. There are sets of individuals referred to as population on which the algorithm works. The algorithm itself is divided into two various sub-sets which are spawned in distinct mannerisms. One set is generated using recommendation operator and a selection scheme; this is a conventional evolutionary method. The construction phase is employed by the other set. It is similar to the construction phase of the SDS. However, the difference is that the traditional tabu evaluation function has been replaced by the general version of the fuzzy tabu evaluation function.

3. RESULTS AND DISCUSSION

In this section, the fuzzy, SDS fuzzy, tabu fuzzy and tabu SDS fuzzy methods are used. The makespan and resource utilization as shown in tables 1 & 2 and figures 1 & 2.

Table 1. Makespan for Tabu SDS Fuzzy

Number of Jobs	Fuzzy	SDS Fuzzy	Tabu Fuzzy	Tabu SDS Fuzzy
200	44	40	41	40
400	92	86	87	85
600	147	138	140	137
800	193	182	185	182
1000	247	226	231	226

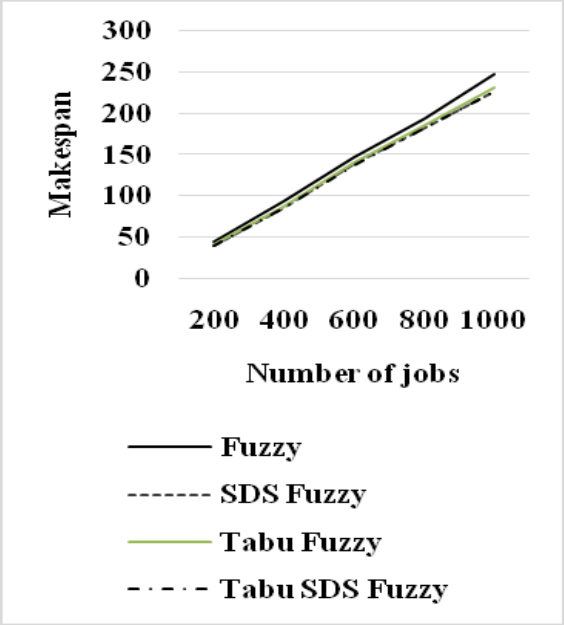

Figure 1. Makespan for Tabu SDS Fuzzy

From the figure 1, it can be observed that the tabu SDS fuzzy has lower makespan by 9.52%, same value & 2.46% for 200 number of jobs, by 7.9%, 1.16% & 2.32% for 400 number of jobs, by 7.04%, 0.72% & 2.16% for 600 number of jobs, by 5.86%, same value & 1.63% for 800 number of jobs and by 8.87%, same value & 2.18% for 1000 number of jobs when compared with fuzzy, SDS fuzzy and tabu fuzzy.

Table 2. Resource Utilization for Tabu SDS Fuzzy

Number of Jobs	Fuzzy	SDS Fuzzy	Tabu Fuzzy	Tabu SDS Fuzzy
200	79	82	81	82
400	78	81	81	81
600	81	84	84	85
800	82	85	84	86
1000	79	81	81	82

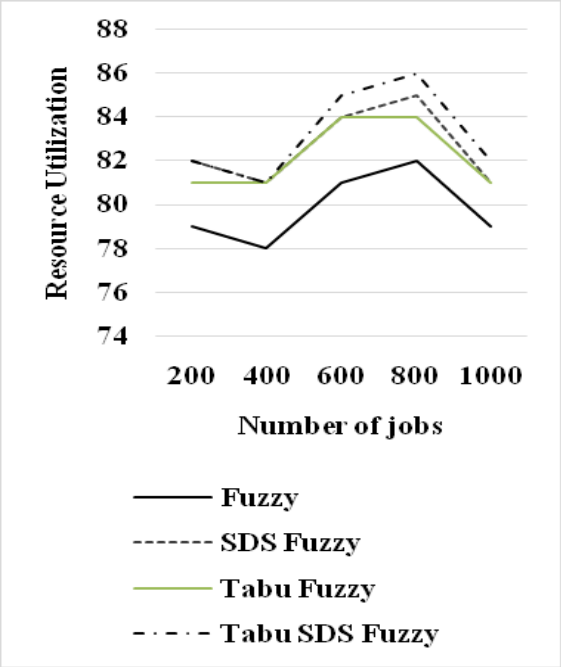

Figure 2. Resource Utilization for Tabu SDS Fuzzy

From the figure 2, it can be observed that the tabu SDS fuzzy has higher resource utilization by 3.72%, same value & 1.22% for 200 number of jobs, by 3.77%, same value & same value for 400 number of jobs, by 4.81%, 1.18% & 1.18% for 600 number of jobs, by 4.76%, 1.16% & 2.35% for 800 number of jobs and by 3.72%, 1.22% & 1.22% for 1000 number of jobs when compared with fuzzy, SDS fuzzy and tabu fuzzy.

4. CONCLUSION

One of the major issues in managing the resources for cloud environment is scheduling. This work suggests a tabu fuzzy technique to solve the issue of scheduling. For determining the TP and selecting the neighbourhood, a fuzzy approach is used by the suggested tabu-fuzzy technique. An important version of the TS is the presented approach, tabu SDS which can decrease the issue of local minima and the non-optimal solution and enhance the finding of the optimal solutions. This strategy makes use of the best neighbours existing in the current list of best solutions; these solutions are used in the SDS to improvise in case the algorithm is unable to find a new neighbour or is stuck in the local minimum. In the construction phase of the SDS, the traditional TS evaluation function has been replaced by a general version of the tabu fuzzy evaluation function. Results show that the tabu SDS fuzzy has higher resource utilization by 3.72%, same value & 1.22% for 200 number of jobs, by 3.77%, same value & same value for 400 number of jobs, by 4.81%, 1.18% & 1.18% for 600 number of jobs, by 4.76%, 1.16% & 2.35% for 800 number of jobs and by 3.72%, 1.22% & 1.22% for 1000 number of jobs when compared with fuzzy, SDS fuzzy and tabu fuzzy.

REFERENCES

[1] Keshk, A. E., El-Sisi, A. B., &Tawfeek, M. A. (2014). Cloud task scheduling for load balancing based on intelligent strategy. International Journal of Intelligent Systems and Applications, 6(5), 25.

[2] Ahluwalia, A. (2016). A Survey on Task Scheduling Techniques in Cloud Computing. International Journal of Engineering Trends and Technology (IJETT), 33 (9).

[3] Nallakumar, R., Sengottaiyan, N., &Nithya, S. (2014). A Survey of Task Scheduling Methods in Cloud Computing. International Journal of Computer Sciences and Engineering, 2(10), 9-13.

[4] Goel, H., &Chamoli, N. (2014). Job Scheduling Algorithms in Cloud Computing: A Survey. International Journal of Computer Applications, 95(23).

[5] Navimipour, N. J., &Milani, F. S. (2015). Task scheduling in the cloud computing based on the cuckoo search algorithm. International Journal of Modeling and Optimization, 5(1), 44.

[6] Chen, Z., Zhu, Y., Di, Y., & Feng, S. (2015). A dynamic resource scheduling method based on fuzzy control theory in cloud environment. Journal of Control Science and Engineering, 2015, 34.

[7] Moschakis, I. A., &Karatza, H. D. (2015). Multi-criteria scheduling of Bag-of-Tasks applications on heterogeneous interlinked clouds with simulated annealing. Journal of Systems and Software, 101, 1-14.

[8] Priya, V., &Babu, C. N. K. (2017). Moving average fuzzy resource scheduling for virtualized cloud data services. Computer Standards & Interfaces, 50, 251-257.

[9] Alla, H. B., Alla, S. B., Ezzati, A., &Mouhsen, A. (2017). A novel architecture with dynamic queues based on fuzzy logic and particle swarm optimization algorithm for task scheduling in cloud computing. In Advances in Ubiquitous Networking 2 (pp. 205-217). Springer, Singapore.

[10] Ma, Y., &Xu, J. (2015). A cloud theory-based particle swarm optimization for multiple decision maker vehicle routing problems with fuzzy random time windows. Engineering Optimization, 47(6), 825-842.

[11] Xu, B., & Sun, Z. (2016). A fuzzy operator based bat algorithm for cloud service composition. International Journal of Wireless and Mobile Computing, 11(1), 42-46.

[12] Alkhashai, H. M., &Omara, F. A. (2016). An Enhanced Task Scheduling Algorithm on Cloud

Computing Environment. International Journal of Grid and Distributed Computing, 9(7), 91-100.
[13] Bagis, A. (2008). Fuzzy rule base design using tabu search algorithm for nonlinear system modeling. ISA transactions, 47(1), 32-44.
[14] Bagis, A. (2003). Determining fuzzy membership functions with tabu search—an application to control. Fuzzy sets and systems, 139(1), 209-225.
[15] Wang, D., Xiong, H., & Fang, D. (2016). A Neighborhood Expansion Tabu Search Algorithm Based On Genetic Factors. Open Journal of Social Sciences, 4(03), 303.
[16] Li, C., Yu, J., & Liao, X. (2002). Fuzzy tabu search for solving the assignment problem. In Communications, Circuits and Systems and West Sino Expositions, IEEE 2002 International Conference on (Vol. 2, pp. 1151-1155). IEEE.
[17] El-henawy, I. M., & Ismail, M. M. (2014). A hybrid swarm intelligence technique for solving integer multi-objective problems. International Journal of Computer Applications, 87(3).
[18] Al-Obaidi, A. T. S., &Majeed, A. B. A. D. (2014). Proposal of tabu search algorithm based on cuckoo search. International Journal of Advanced Research in Artificial Intelligence (IJARAI), 3(3).