



Contents list available at JMCS

Journal of Mathematics and Computer Science

Journal Homepage: www.tjmcs.com



JSWA: An Improved Algorithm for Grid Workflow Scheduling using Ant Colony Optimization

¹Emetis Niazmand, ²Arash Ghorbannia Delavar,

³Javad Bayrampoor, ⁴Ali Reza Khalili Boroujeni

Department of Computer, Payame Noor University,

PO BOX 19395-3697, Tehran, IRAN

¹*emetis.niazmand@yahoo.com, ²a_ghorbannia@pnu.ac.ir, ³javadbayrampoor@yahoo.com,
⁴khalili_boroujeni@yahoo.com*

Article history:

Received February 2013

Accepted April 2013

Available online April 2013

Abstract

In this paper we propose an improved algorithm for scheduling grid workflow by using ant colony optimization method. Ant colony optimization (ACO) is a meta-heuristic for combinatorial optimization problems. JSWA algorithm is measured by using parameters such as reliability, cost, request and acknowledgement time and bandwidth. Regarding the proposed algorithm and its comparison with scheduling algorithm, we have established a new competency through which the tasks are carried out by considering preference criterion parameters. To do so, there should be less time complexities in accessing tasks for the present algorithms compared with the proposed one. By implementing a technical method we could consider a system in which the efficiency and optimization are increased and finally the time needed for program performance is decreased by using the target function. Also we could estimate the real time of tasks' commute by calculating the commute time compared with the previous algorithms. The result is that JSWA is more efficient than the algorithms such as ACS and MOACO.

Keywords: Grid Workflow Scheduling, Ant Colony Optimization, Meta-heuristic, JSWA.

1. Introduction

In recent years science scholars have succeeded in presenting new resolutions through which they can create better capabilities by different simulations of the environmental conditions. Ant colony method is one of the technical meta-heuristic tools using the environment in its present mood.

Grid computing environment includes different interconnected machines by interface networks to execute different tasks that have diverse computational requirements. In distributed systems such as grids, in which there are various sources, deciding about regularity of task and selecting the computing machines is an important issue in grid environments. The main purpose of grid systems is to optimize using sources and maximize the efficiency of the system. Managing various resources and task scheduling in grid environment is a challenging and indispensable task [2, 4]. In recent years, grid computing has attracted the attention of engineers and scientists for executing high performance parallel and distributed applications due to major advancements in wide-area network technologies and low cost of powerful computing and high-speed network resources [10]. Grid environments have tried to study various scheduled algorithms in reaching the above purposes. Scheduling is an important tool for this purpose. Tasks scheduling is an NP-complete problem and finding the absolute optimal solution is so hard [2, 7, 10, 12, 13]. So many heuristics have been developed to solve this problem. The heuristic scheduling can be classified into two categories: on-line mode and batch-mode heuristics. In the on-line mode heuristics, a task is mapped on to a machine as soon as it arrives at the scheduler. In the batch-mode heuristics, tasks are not mapped on to machines as they arrive; instead, they are collected into the buffer and then it is scheduled at prescheduled time [2, 5, and 6].

Computational models for Grid scheduling problems and their resolution use heuristic and meta-heuristic approaches. Scheduling problems are at the heart of any grid-like computational system. Different types of scheduling based on different criteria, such as static versus dynamic environment, multi-objectivity, adaptability, etc., are identified [9].

In the second section of the paper, we will discuss the previous works and in the third section the main part of our research and the formulas utilized will be studied and analyzed. In section 4, the related simulations by comparing the different simulations of ACS and MAOCO with our proposed method will be presented; and in part 5, we will reach an overall conclusion on the discussion.

2. Related Works

By studying the previous algorithms, it was found that the real processing times have not been investigated. In other words, the selected matrix has only produced a method which is useful only during the administration time and it does not request and acknowledge time.

2.1. Ant colony optimization method with ACS algorithm

Most calculating functions concentrated on business and sciences can be shown as work flows. We can design and implement each work flow in some layers. The higher layer can explain workflow and the second layer is used to present a subjective view of workflow. In this case workflow is shown as tasks and for performing each workflow a series of services is used as follows: $s_i = \{s_i^1, s_i^2, \dots, s_i^{m_i}\}$. In article [1], Authors focus on scheduling module to propose a suitable scheduling algorithm. The abstract model for workflow can show a graph of Directed Acyclic Graphs (DAG) [1, 8, 10, 11] in which $G = (V, A)$ and n is the number of tasks and $V = \{T_1, T_2, \dots, T_n\}$.

2.1.1. ACS algorithm for the scheduling problem: The elementary idea of ACO is to simulate the foraging behaviour of ant colonies. When a group of ants sets out from the nest to search for the food source, they use a special kind of chemical to communicate with each other. The chemical is referred to as pheromone. Once the ants discover a path to food, they deposit pheromone on the path. By sensing pheromone on the ground, an ant can follow the trails of the other ants to the food source. As this process continues, most of the ants tend to choose the shortest path as there have been a huge amount of pheromones accumulated on this path. This collective pheromone-depositing and pheromone-following behaviour of ants becomes the inspiring source of ACO [1, 7, 13, 14, and 15].

In reference [1] the adaptive ACS algorithm to tackle the workflow scheduling problem in grid applications has been applied. Informally, the algorithm can be viewed as the interplay of the following procedures.

Initialization of algorithm: All pheromone values and parameters are initialized at the beginning of the algorithm.

Initialization of ants: A group of M artificial ants are used in the algorithm. In each iteration, each ant randomly selects a constructive direction and builds a sequence of tasks.

Solution construction: M ants set out to build M solutions to the problem based on pheromone and heuristic values using the selection rule of the ACS algorithm.

Local pheromone updating: Soon after an ant maps a service instance s_{ij} to task T_i , the corresponding pheromone value is updated by a local pheromone updating rule.

Global pheromone updating: After all ants have completed their solutions at the end of each iteration, pheromone values corresponding to the best-so-far solution are updated by a global pheromone updating rule.

Terminal test: If the test is passed, the algorithm will be ended. Otherwise, go to step 2) to begin a new iteration.

Heuristic information is an important factor in ACO algorithm. By adding a heuristic factor to the current ACO, the ants were able not only to use previous findings, but also decrease the effect of pheromone misdirection in unrelated routing and overcome low-speed

convergence [16]. Some presuppositions and formulas were presented for calculating 7 heuristic methods as cost, reliability, time, proposed budget, proposed finishing time, cost and time simultaneously and the results were shown with a series of simulations.

2.2. Multiple-Objective Ant Colony Optimization

The first extensions of ACO for MCOPs were proposed by the end of the 90s. While various early approaches targeted problems where the objectives can be ordered lexicographically or preferences of a decision maker are known, most proposals of MOACO algorithms were targeting towards multi-objective problems that are tackled in terms of Pareto optimization. For an overview of most of the available Multiple-Objective Ant Colony Optimization (MOACO) algorithms we refer to [3].

2.2.1. Available MOACO algorithms: All proposed MOACO algorithms somehow try to modify underlying ACO components so as to allow the algorithm to direct the search towards the different regions of the Pareto front simultaneously. A main question is how to represent the solution components of different areas of the front in the form of pheromones. Essentially, the possibilities available here are to use one pheromone matrix or several pheromone matrices. In the latter case, typically each objective has associated one pheromone matrix. Related to this decision is the choice of the ants that deposit pheromones. Typically, if one pheromone matrix is used, some or all non-dominated solutions are selected for update while in the use of several pheromone matrices some elitist choices w.r.t. the objectives represented in the pheromone matrices are done. During the solution construction, the question arises how to use the pheromones and also the heuristic information. If several pheromone matrices or several forms of heuristic information are used, a common choice is to use some weighted aggregation of these values [3].

3. Problem Definition

Regarding the works previously done, the parameters considered for service quality include 3 parameters of reliability, cost and time, and in this paper we have added bandwidth to them. Also we will modify the formulas and devise some amounts for the parameters in commute time which have not been utilized in the main research paper. We appropriate some amounts in tasks to administrate the task itself and some amount for retrieval and also some amount for the response. Also the formulas used in the previous paper involved time and calculation complexities which will be improved greatly regarding the formulas we will present here.

The related equations will appear as follows:

Service instances will be $s_i = \{s_i^1, s_i^2, \dots, s_i^{m_i}\}$ in which s is the number of service samples.

Grid Service Provider (GSP) is $s_i^j.r, s_i^j.t, s_i^j.c, s_i^j.b$ respectively for reliability, time, cost and bandwidth parameters.

The mapping heuristic information in the graph will be as follows: using max amounts is to devise $[0, 1]$ for η .

1- Reliability Heuristic: The amount of heuristic for mapping s_i^j to T_i as η_{ij} for Reliability is:

$$\eta_{ij} = RG_{ij} = \frac{s_{-rp_j} + s_{-rr_{ij}} + s_{-ra_{ij}}}{3 * \max_r} \quad (1)$$

In which s_{-rp_j} is related to Task performance reliability and $s_{-rr_{ij}}$ related to demand reliability, $s_{-ra_{ij}}$ related to respond to demand reliability and \max_r related to the highest reliability.

2- Cost Heuristic: The amount of heuristic for mapping s_i^j to T_i as η_{ij} for cost is:

$$\eta_{ij} = CG_{ij} = \frac{s_{-cp_j} + s_{-cr_{ij}} + s_{-ca_{ij}}}{\max_c} \quad (2)$$

In which s_{-cp_j} is related to Task performance cost and $s_{-cr_{ij}}$ related to demand cost, $s_{-ra_{ij}}$ related to respond to return cost and \max_c related to the highest cost.

3- Time Heuristic: The amount of heuristic for mapping s_i^j to T_i as η_{ij} for time is:

$$\eta_{ij} = TG_{ij} = \frac{s_{-tp_j} + s_{-tr_{ij}} + s_{-ta_{ij}}}{\max_t} \quad (3)$$

In which s_{-tp_j} is related to Task performance time and $s_{-tr_{ij}}$ related to demand time, $s_{-ta_{ij}}$ related to respond to demand time and \max_t related to the longest time.

4- Bandwidth Heuristic: The amount of heuristic for mapping s_i^j to T_i as η_{ij} for bandwidth is:

$$\eta_{ij} = BG_{ij} = \frac{s_{-bp_j} + s_{-br_{ij}} + s_{-ba_{ij}}}{3 * \max_b} \quad (4)$$

In which s_{-bp_j} is related to Task performance bandwidth and $s_{-br_{ij}}$ related to demand bandwidth, $s_{-ba_{ij}}$ related to respond to demand bandwidth and \max_b related to the broadest bandwidth.

5- Suggested Budget Heuristic: The amount of heuristic for mapping s_i^j to T_i as η_{ij} for suggested budget is:

$$SB_i = \frac{budget}{\cos t_i} \tag{5}$$

$$\eta_{ij} = \frac{1}{SB_i} \tag{6}$$

6- Suggested Deadline: The amount of heuristic for mapping s_i^j to T_i as η_{ij} for suggested deadline is:

$$avg_time_i = \frac{EST_i + BEST_i}{2} \tag{7}$$

$$SD_i = avg_time_i \times \frac{Deadline}{max_time} \tag{8}$$

$$\eta_{ij} = \frac{1}{SD_i} \tag{9}$$

In which *ETS* is the earliest performance time in downward status (first to end) and *BEST* is the earliest performance time in upward status (end to first).

7- Time/Cost heuristic: The amount of heuristic for mapping s_i^j to T_i as η_{ij} for time/cost is:

$$\eta_{ij} = \frac{1}{2}(CG_{ij} + TG_{ij}) \tag{10}$$

8- Overall Heuristic: The amount of heuristic for mapping s_i^j to T_i as η_{ij} for overall qualities is:

$$\eta_{ij} = \frac{1}{4}(TG_{ij} + CG_{ij} + RG_{ij} + BG_{ij}) \tag{11}$$

The limitations used in this suggested method are as follows:

Reliability limitation: the needed reliability in this situation should not be less than the threshold level one.

$$K.Re\ liability = \min_{1 \leq i \leq n} S_i^K . R \geq Re\ liability\ C\ onstra\ int \tag{12}$$

Time limitation: it is not acceptable to consider a time period more than the deadline for the whole process.

$$\sum_{i=1}^n s_i^{k_i} .t \leq \text{Deadline} \tag{13}$$

Cost limitation: the expended cost should be less than the overall cost.

$$\sum_{i=1}^n s_i^{k_i} .c \leq \text{Budget} \tag{14}$$

Bandwidth limitation: bandwidth should not be less than the bandwidth needed.

$$K .\text{Bandwidth} = \min_{1 \leq i \leq n} S_i^{K_i} .b \geq \text{Bandwidthc onstra int} \tag{15}$$

Reliability optimization: in this situation, we consider cost, time and bandwidth limitations to reach a condition where the most reliability is achieved.

Time optimization: in this situation, we consider cost, reliability and bandwidth limitations to reach a condition where the least time is spent to do the job.

Cost optimization: in this situation, we consider reliability, time and bandwidth limitations to reach a condition where the least cost is incurred to do the job.

Bandwidth optimization: in this situation, we consider cost, time and reliability limitations to reach a condition where the broadest bandwidth is used to do the job.

Local update pheromone: the primary amount of τ_0 is calculated as follows:

In the first step, we will regard τ_0 to be equal with η_{ij} . In the future steps of pheromone calculation and making the local equation update, the following will be done:

$$\begin{cases} \tau_{ij}^{new} = (1 - \rho)\tau_{ij}^{old} + \rho\tau_0 & \text{if Task Selected} \\ \tau_{ij}^{new} = (1 - \rho)\tau_{ij}^{old} & \text{else} \end{cases} \tag{16}$$

For example for a heuristic method we have the following explanation.

$$\begin{cases} \tau_A^{new} = (1 - \rho)\tau_A^{old} + \rho\tau_0 & \text{if Task Selected} \\ \tau_A^{new} = (1 - \rho)\tau_A^{old} & \text{else} \end{cases} \tag{17}$$

To make the common parameters update, which is going to be achieved by local up-to-dating, the best statuses for the tasks will be considered.

$$\tau_{ij}^{new} = (1 - \rho)\tau_{ij}^{old} + \rho f \tag{18}$$

And f of Reliability, Cost, Time and Bandwidth optimization will be considered as follows:

Reliability optimization: in this case we will consider 3 statuses and the amount of f is calculated by the following:

$$f = \begin{cases} 0.5 + \frac{\text{avg_Reliability}}{\text{max_Reliability}} & \text{if } k_cost > \text{budget} \ \& \ \text{makspan} > \text{Deadline} \\ 1 + \frac{\text{avg_Reliability}}{\text{max_Reliability}} & \text{if } k_cost < \text{budget} \ \& \ \text{makspan} < \text{Deadline} \\ 0.75 + \frac{\text{avg_Reliability}}{\text{max_Reliability}} & \text{otherwise} \end{cases} \quad (19)$$

In which avg_Reliability is related to average reliability, max_Reliability is the maximum amount of reliability, K_cost is related to the whole cost for the best status, K_makspan is related to the time of doing all the task in the best status, Budget is related to the suggested budget and Deadline is related to the suggested deadline.

Time optimization: in this case we will consider 2 statuses and the amount of f is calculated by the following:

$$f = \begin{cases} \frac{\text{Budget}}{\text{K_cost}} + \frac{\text{min_time}}{\text{avg_time}} & \text{if } k_cost > \text{Budget} \\ 1 + \frac{\text{min_time}}{\text{avg_time}} & \text{if } k_cost \leq \text{Budget} \end{cases} \quad (20)$$

In which avg_time is related to average time, and min_time is the least time needed for performance and K_cost is related to the whole cost for the best status and Budget is related to the suggested budget.

Cost optimization: in this case we will consider 2 statuses and the amount of f is calculated by the following:

$$f = \begin{cases} \frac{\text{Deadline}}{\text{K_makspan}} + \frac{\text{min_cost}}{\text{avg_cost}} & \text{if } k_makspan > \text{Deadline} \\ 1 + \frac{\text{min_cost}}{\text{avg_cost}} & \text{if } k_makspan \leq \text{Deadline} \end{cases} \quad (21)$$

In which avg_cost is related to average performance cost, and min_cost is the least cost needed for performance and K_makspan is related to the whole time needed for the best status and Deadline is related to the suggested deadline.

Bandwidth optimization: in this case we will consider 3 statuses and the amount of f is calculated by the following:

$$f = \begin{cases} 0.5 + \frac{\text{avg_Bandwidth}}{\text{max_Bandwidth}} & \text{if } k_cost > \text{budget} \& \ k_makspan > \text{Deadline} \\ 1 + \frac{\text{avg_Bandwidth}}{\text{max_Bandwidth}} & \text{if } k_cost < \text{budget} \& \ k_makspan < \text{Deadline} \\ 0.75 + \frac{\text{avg_Bandwidth}}{\text{max_Bandwidth}} & \text{otherwise} \end{cases} \quad (22)$$

In which avg_Bandwidth is related to average bandwidth, max_Bandwidth is the broadest bandwidth needed for performance and K_cost is related to the whole cost for the best status, K_makspan is related to the whole time needed for the best status, Budget is related to the suggested budget and Deadline is related to the suggested deadline.

The main equation for the next Task selection will be as follows:

$$\left\{ \begin{array}{l} s_i^j = \arg \max_{1 \leq j \leq m_i} \tau_{ij}^\alpha \eta_{ij}^\beta \quad \text{if } q < q_0 \\ \text{Select Random else} \end{array} \right. \quad (23)$$

The graph used for the data is as follows (Figure 1):

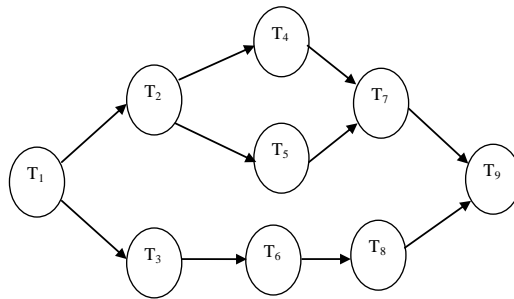


Fig1. An e-Economic workflow

We will consider algorithm administration as follows:

Adjusting the presuppositions and primary amount allocation, selecting ants with higher amounts of capabilities. Ants with capabilities less than the threshold level expected are not utilized.

Calculating η values to calculate the mapping function.

Administering mapping function to select the next status and selection of q_0 .

If $q < q_0$, the best Task is selected regarding the mapping function. Otherwise, task selection will be carried out randomly.

Local update for all ants.

Returning to status 3, until all tasks are administered.

General update by using ants' local update (choosing the best status up to now)

Administering the cycle from status 3 until all cycles finish.

The flowchart of the JSWA is shown on Figure 2.

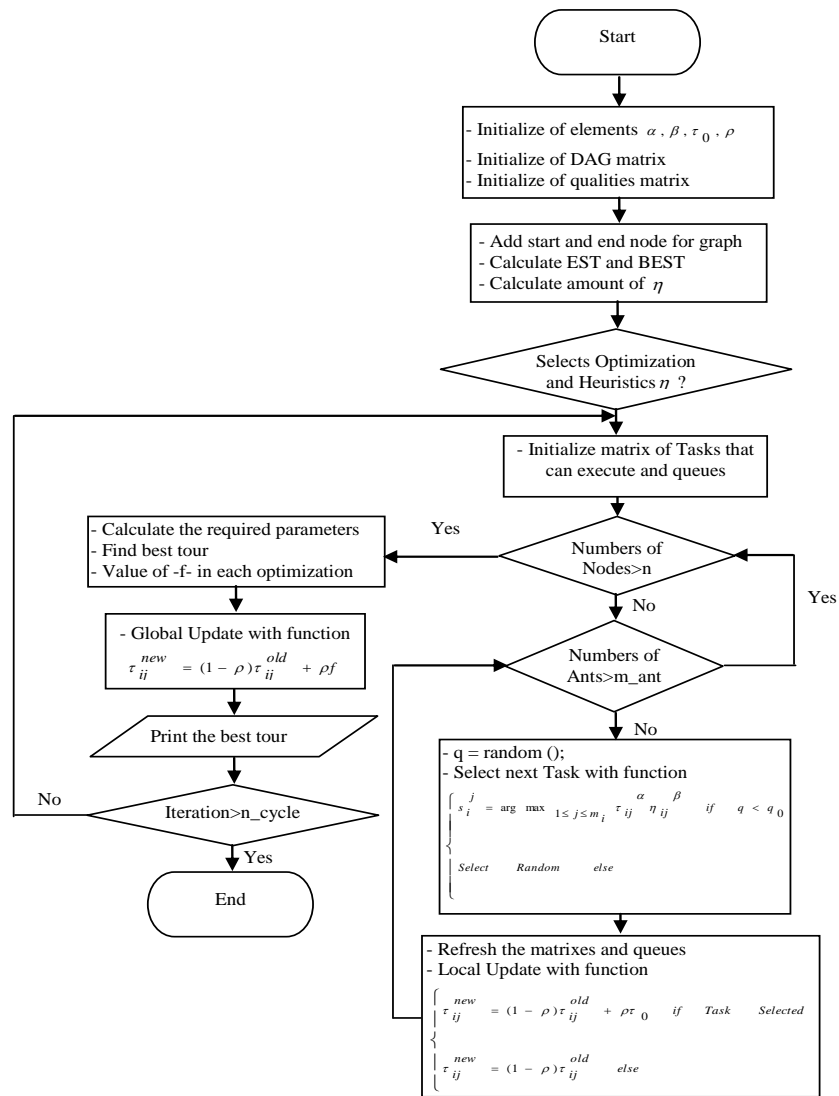


Fig2. Flowchart of the JSWA algorithm

The Algorithm of the JSWA is shown as follows:

JSWA Algorithm

This algorithm is used for the scheduling problem with several QOS parameter with ant colony system approach.

- Initialize of elements m_ant , $n=9$, Row, $taw0$, alpha, beta, Budget, Reliability Constraint, Deadline, taw , n_cycle , $q0$.

- Initialize of DAG (Directed Acyclic Graph) matrix

- Add start node for forward graph

- Add start node for backward graph

- Initialize of Reliability, Cost, Time, Bandwidth matrix

%-----

- Heuristic Information (Calculate the ETA= RG (Reliability Greedy), TG (Time Greedy), CG (Cost Greedy), BG (Bandwidth Greedy), SD (Suggested Deadline), SB (Suggested Budget), TC (Time/Cost), OP (Overall Performance))

- Selects which Optimization and then which Heuristics you want for Task map

%-----

The scheduling rule of mapping a service instances S to T is: mapping formula

for $x=1:n_cycle$

Initialize matrix of Tasks that can execute

Initialize of all queues

for $i=1:n$

for $j=1:m_ant$

$q = \text{random} ();$

if($q \leq q0$)

$S = taw.^{\alpha} + ETA.^{\beta};$

else

Select randomly one of the Tasks that can be executed and not mapped

end

Refreshes the matrix of tasks that can execute

if(select the Task)

$taw = (1 - Row) * taw + Row * taw0; \% \text{Local Update pheromone}$

```

else
taw =(1-Row)*taw;
end
    Refresh the queues
end
end
%-----
Calculate the required parameters
Find best tour
Value of -f- in each optimization
taw (i,j)=(1-Row)*taw (i,j)+Row*f; %Global Update pheromone
Print the best tour
end

```

4. Simulation and Analysis

Regarding the formulation in the previous methods and the changes implemented and added to them and coding the text and simulations for all the statuses to select the next status, considering the amount of η and pheromone evaporation and 8 heuristic statuses considered for Reliability, Cost, Time, Bandwidth, Suggested Budget, Time and Cost together, Suggested Deadline and the Overall heuristic statuses altogether, the following methods were achieved by heuristic methods. Comparing the simulations regarding the mapping resulted and presupposed values and selection of each type of heuristic methods for each of the optimizations mentioned above for the suggested method, shows a kind of improvement. Although the results are gained from only one graph, we can achieve the same results for the other graphs too. Anyway, the proposed algorithm has less complexity of time and a better efficiency in comparison to the previous algorithms. Simulation results are as follows:

Simulation for the status in which Reliability optimization is considered, is shown in Figure3. Simulation for the status in which Cost optimization is considered, is shown in Figure4. Simulation for the status in which Time optimization is considered, is shown in Figure5. Simulation for the status in which Bandwidth optimization is considered, is shown in Figure6.

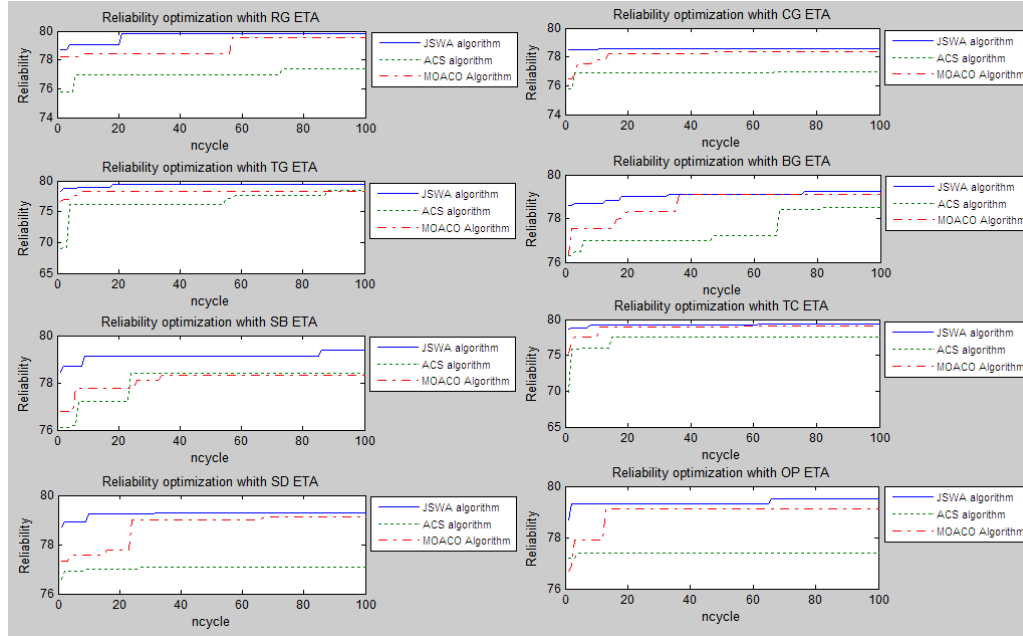


Fig3. Reliability Optimization: Reliability Heuristic (RG), Cost Heuristic (CG), Time Heuristic (TG), Bandwidth Heuristic (BG), Suggested Budget Heuristic (SD), Time / Cost Heuristic (TC), Over All Heuristic (OP)

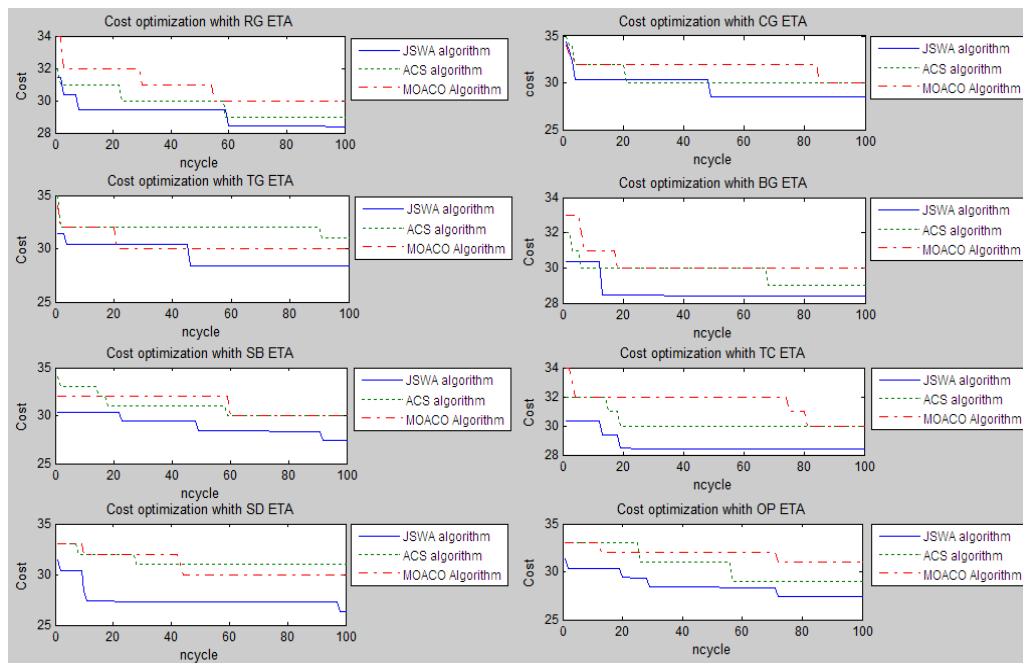


Fig4. Cost Optimization: Reliability Heuristic (RG), Cost Heuristic (CG), Time Heuristic (TG), Bandwidth Heuristic (BG), Suggested Budget Heuristic (SD), Time / Cost Heuristic (TC), Over All Heuristic (OP)

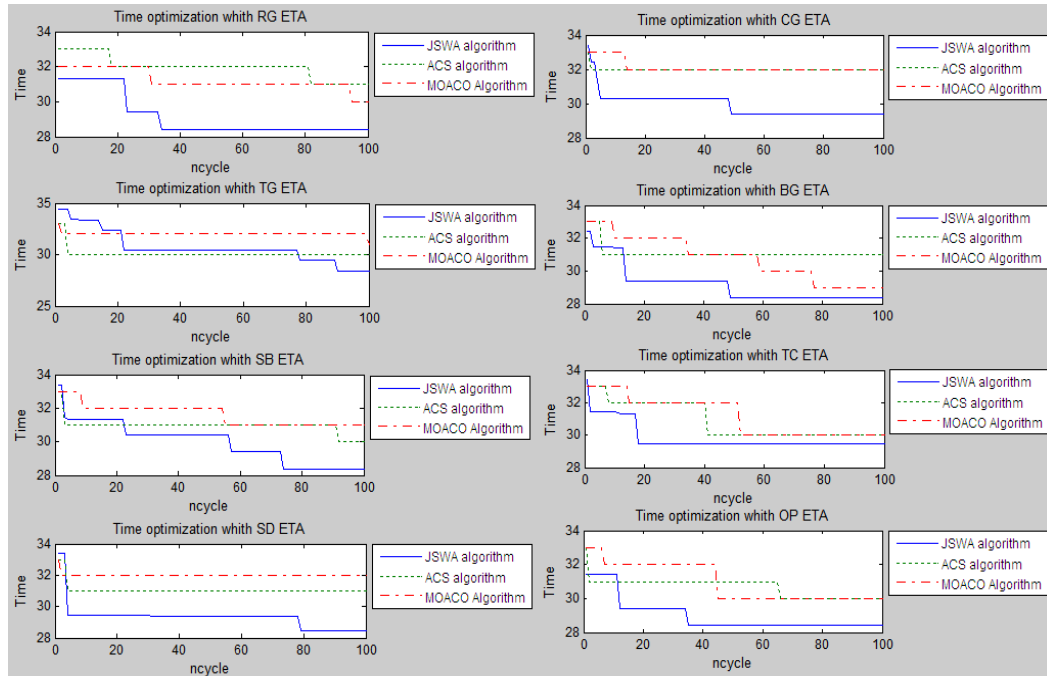


Fig5. Time Optimization: Reliability Heuristic (RG), Cost Heuristic (CG), Time Heuristic (TG), Bandwidth Heuristic (BG), Suggested Budget Heuristic (SD), Time / Cost Heuristic (TC), Over All Heuristic (OP)

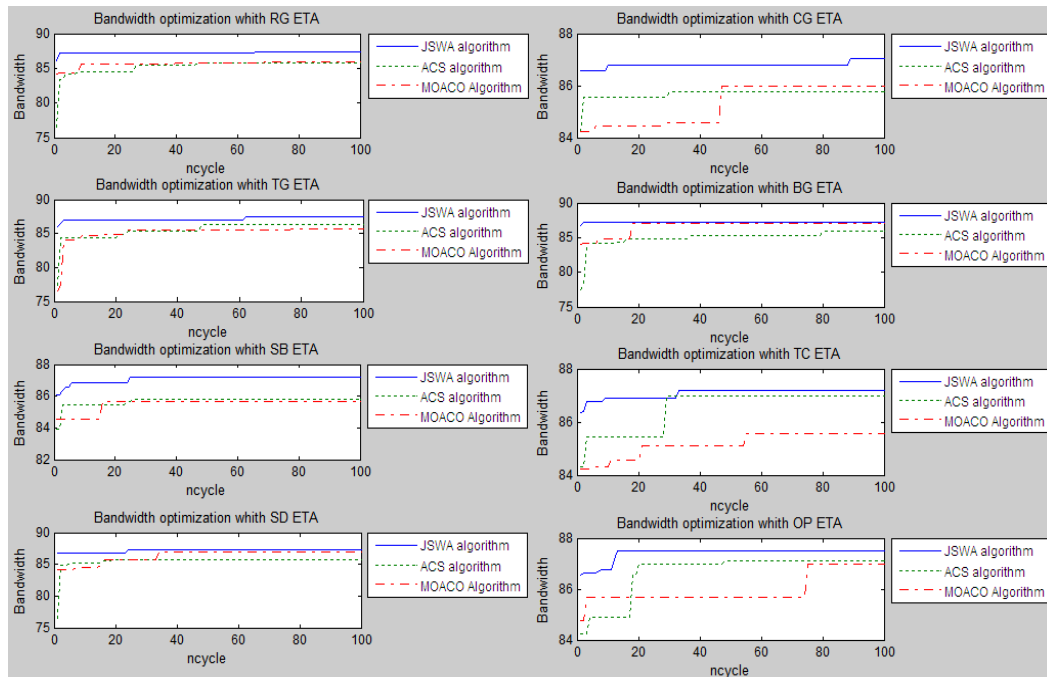


Fig6. Bandwidth Optimization: Reliability Heuristic (RG), Cost Heuristic (CG), Time Heuristic (TG), Bandwidth Heuristic (BG), Suggested Budget Heuristic (SD), Time / Cost Heuristic (TC), Over All Heuristic (OP)

Simulation and analysis with parameters $\alpha=0.7$, $\beta=0.4$, $\rho=0.6$, $q_0=0.3$, $number_of_ants=100$ and $iteration=100$ for Reliability optimization, Bandwidth optimization, Cost optimization and Time optimization regarding heuristics RG, CG, TG,

BG, SB, TC, SD and OP respectively for Reliability, Cost, Time, Bandwidth, suggested Budget, Time/Cost, suggested Deadline, Overall heuristics are shown in Table 1.

Table1. Result of Experiment

Reliability Optimization	RG	CG	TG	BG	SB	TC	SD	OP
max amount for ACS Algorithm	77.40	77.00	78.50	78.50	78.40	77.60	77.10	77.40
max amount for JSWA Algorithm	79.80	78.60	79.50	79.23	79.40	79.30	79.30	79.50
max amount for MOACO Algorithm	79.56	78.33	78.22	79.11	78.33	79.11	79.11	79.11
improved JSWA compared to ACS	3.01	2.04	1.26	0.93	1.26	2.14	2.77	2.64
improved JSWA compared to MOACO	0.31	0.34	1.61	0.15	1.34	0.24	0.24	0.49
Bandwidth Optimization	RG	CG	TG	BG	SB	TC	SD	OP
max amount for ACSAlgorithm	85.78	85.78	86.33	86.00	85.78	87.00	85.67	87.11
max amount for JSWAAAlgorithm	87.37	87.04	87.44	87.19	87.19	87.19	87.26	87.52
max amount for MOACOAlgorithm	85.89	86.00	85.67	87.00	85.67	85.56	87.00	87.00
improved JSWA compared to ACS	1.82	1.45	1.27	1.36	1.61	0.21	1.83	0.47
improved JSWA compared to MOACO	1.70	1.19	2.03	0.21	1.74	1.87	0.30	0.59
Cost Optimization	RG	CG	TG	BG	SB	TC	SD	OP
min amount for ACSAlgorithm	29.00	30.00	31.00	29.00	30.00	30.00	31.00	29.00
min amount for JSWAAAlgorithm	28.38	28.56	28.37	28.38	27.38	28.44	26.38	27.44
min amount for MOACOAlgorithm	30.00	30.00	30.00	30.00	30.00	30.00	30.00	31.00
improved JSWA compared to ACS	2.14	4.80	8.48	2.14	8.73	5.20	14.90	5.38
improved JSWA compared to MOACO	5.40	4.80	5.43	5.40	8.73	5.20	12.07	11.48
Time Optimization	RG	CG	TG	BG	SB	TC	SD	OP
min amount for ACSAlgorithm	31.00	32.00	30.00	31.00	30.00	30.00	31.00	30.00
min amount for JSWAAAlgorithm	28.42	29.40	28.39	28.40	28.39	29.44	28.43	28.42
min amount for MOACOAlgorithm	30.00	32.00	31.00	29.00	31.00	30.00	32.00	30.00
improved JSWA compared to ACS	8.32	8.12	5.37	8.39	5.37	1.87	8.29	5.27
improved JSWA compared to MOACO	5.27	8.12	8.42	2.07	8.42	1.87	11.16	5.27

5. Conclusion

The results of experiments show that JSWA algorithm is improved and it has less complexity and a better efficiency compared with the two other algorithms, namely ACS and MOACO. Average improvement of JSWA as compared to ACS is 2.01, 1.25, 6.47, 6.37 and average improvement of JSWA as compared to MOACO is 0.59, 1.20, 7.31, and 6.32 respectively for Reliability optimization, Bandwidth optimization, Cost optimization and Time optimization. Also the proposed algorithm reduces the time and cost and changes reliability and bandwidth to the best possible amount. It also has resulted in a better mapping function for the Tasks and the next status selection. The next work can focus on the delay created in ants distribution and the delay present for responding and concentrated processing. Also we can consider the nodes and parameters to be dynamic regarding the environmental conditions and achieve the best scheduling in this new situation.

References

- [1] Wei-Neng Chen and Jun Zhang, "An Ant Colony Optimization Approach to a Grid Workflow Scheduling Problem With Various QoS Requirements", *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS*, VOL. 39, NO. 1, JANUARY (2009).
- [2] Arash Ghorbannia Delavar, Ali Reza Kalili Boroujeni and Javad Bayrampoor, "BPISG: A Batching Heuristic Scheduling Algorithm With Taking Index Parameters for Mapping Independent Tasks on Heterogeneous Computing Environment", *IJCSI International Journal of Computer Science Issues*, Vol. 8, Issue 6, No 1, November (2011).
- [3] Manuel lopez-ibanez, Thomas stuetzle, "An Analysis of Algorithmic Components for Multi objective Ant Colony Optimization: A Case Study on the Biobjective TSP", June (2009).
- [4] A.GhorbanniaDelavar, M.Nejadkheirallah and M.Motalleb, "A New Scheduling Algorithm for Dynamic Task and Fault Tolerant in Heterogeneous Grid Systems Using Genetic Algorithm", *IEEE* (2010).
- [5] Kamalam.G.K and Muralibhaskaran.V, "A New Heuristic Approach: Mean Algorithm for Scheduling Meta-Tasks on Heterogeneous Computing Systems", *Journal of Computer Science and Network Security*, January (2010).
- [6] G. K. Kamalam and V. MuraliBhaskaran, "An Improved Min-Mean Heuristic Scheduling Algorithm for Mapping Independent Tasks onHeterogeneous Computing Environment", *Journal of Computational cognition*, December (2010).
- [7] Jun Mao, "Task Scheduling of parallel programming systems using Ant Colony Optimization", *Proceedings of the Third International Symposium on Computer Science and Computational Technology(ISC SCT '10)* Jiaozuo, P. R. China, 14-15, pp. 179-182, ,August (2010).
- [8] Fangpeng Dong; Selim G. Akl, "Distributed Double-Level Workflow Scheduling Algorithms for Grid Computing" *Journal of Information Technology and Applications* Vol. 1 No. 4, pp. 261-273, March (2007).
- [9] FatosXhafa, Ajith Abraham, "Computational models and heuristic methods for Grid scheduling problems", *Future Generation Computer Systems* 26, 608_621,(2010).
- [10] AmitAgarwal, Padam Kumar, "Economical Task Scheduling Algorithm for Grid Computing Systems" *Global Journal of Computer Science and Technology*, Vol. 10 Issue 11 (Ver. 1.0), October (2010).
- [11] Yi He, Zili Shao, Bin Xiao, QingfengZhuge, Edwin Sha, "Reliability Driven Task Scheduling for Heterogeneous Systems", This work is partially supported by TI University Program, NSF EIA-0103709 and Texas ARP 009741-0028, (2001).
- [12] LianghongWu,Yaonan Wang, and Shaowu Zhou, "Improved differential evolution algorithm for resource-constrained project scheduling problem", *Journal of Systems Engineering and Electronics* Vol. 21, No. 5, pp.798–805 Available online at www.jseepub.com,October (2010).

- [13] Claudia Ruth Gatica, Susana Cecilia Esquivel, Guillermo Mario Leguizamon, "An ACO approach for the Parallel Machines Scheduling Problem", *Inteligencia Artificial* 46, 84-95 doi: 10.4114/ia.v14i46.1550, **(2010)**.
- [14] P. S. GEORGILAKIS, P. G. VERNADOS, C. KARYTSAS, "An ant colony optimization solution to the integrated generation and transmission maintenance scheduling problem", *JOURNAL OF OPTOELECTRONICS AND ADVANCED MATERIALS* Vol. 10, No. 5, p. 1246 – 1250, May **(2008)**.
- [15] A.E. Rizzoli, R. Montemanni, E. Lucibello, L.M. Gambardella, "Ant colony optimization for real-world vehicle routing problems From theory to applications", 13 September **(2007)**.
- [16] A. Ghorbannia Delavar, S. Hoseyny, R. Maghsoudi, "BCO-Based Optimized Heuristic Strategies for QoS Routing", *The Journal of Mathematics and Computer Science* Vol .5 No.2, 105-114 **(2012)**.