TASK SCHEDULING IN GRID ENVIRONMENT WITH ANT COLONY METHOD FOR COST AND TIME

Arash Ghorbannia Delavar[a], Javad Bayrampoor[a],
Ali Reza Khalili Boroujeni[a], Ali Broumandnia[b]

a: Department of Computer, Payame Noor Universtiy, PO BOX 19395-3697, Tehran, IRAN
b: Department of Computer, Islamic Azad Universtiy, Souch Tehran Branch, IRAN
arash.ghorbannia@gmail.com, javadbayrampoor@yahoo.com,
khalili_boroujeni@yahoo.com, broumandnia@azad.ac.ir

ABSTRACT

In this paper, we would like to propose an improved algorithm for task scheduling in grid environment with metaheuristic ant colony optimization method considering the cost and time parameters from quality of service. The proposed algorithm is evaluated by using the required cost and time parameters to carry out the task. With implementation these parameters in the simulate environment, we can create situation that scheduling task will be done with better position and achieve high performance on computational grids. Finally the experiment and simulated results will show that the proposed heuristic scheduling algorithm performs significantly to ensure high throughput, reduced time and cost. Also proposed algorithm is more efficient in the grid environment. This proposed algorithm is more efficient than the adaptive ACS and MOACO algorithms.

KEYWORDS

Grid; Scheduling; Ant Colony Optimization; Metaheuristic Algorithm

1. INTRODUCTION

For many years, researchers and scientists, taking into account and inspired by the environment, have managed to invent methods in order to address the existing issues, particularly issues that are NP-complete. Ant colony optimization method is one of the technical and metaheuristic approaches to solving NP-complete problems using the current state of the environment.

Grid environments have tried to study various scheduling algorithms for 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 too hard [2, 7, 10, 11, 12]. So many heuristics have been developed to solve this hard 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, 6].

Ant colony optimization (ACO) is a metaheuristic approach in which a colony of artificial ants cooperates in finding good solutions to difficult optimization problems. Ant colony optimization (ACO) is a nature inspired method for the solution of hard combinatorial optimization problems [3, 12, 13, 14].

DOI : 10.5121/ijcsea.2012.2501

Task duplication is an effective scheduling technique for reducing the response time of workflow applications in dynamic grid computing systems. In literature, task scheduling heuristics for DAG applications have been classified as list scheduling, cluster-based scheduling and duplication-based scheduling [10].

For the majority of Grid systems, scheduling is a very important mechanism. In the simplest of cases, scheduling of jobs can be done in a blind way by simply assigning the incoming tasks to the available compatible resources. Nevertheless, it is a lot more profitable to use more advanced and sophisticated schedulers. Moreover, the schedulers would generally be expected to react to the dynamics of the Grid system, typically by evaluating the present load of the resources, and notifying when new resources join or drop from the system. Additionally, schedulers can be organized in a hierarchical form or can be distributed in order to deal with the large scale of the Grid [8].

In this paper and in the second part of this paper, the attempts that have already been made in this respect will be discussed. In the third part of the paper, the main task and the formulas used in this paper will be discussed which include presenting the new algorithm and competence function. In the fourth part, related simulations are conducted along with a comparison made between simulations associated with the proposed method and adaptive ACS and MOACO methods. The fifth part includes the conclusion.

2. RELATED WORKS

Heuristic scheduling algorithms require mapping and scheduling tasks on machines in a grid processing system in order that it may construct an acceptable optimization. For example, mapping a task on to a machine with a minimum access time and a minimum cost; mapping a task on to a machine with a faster execution time than the time of the availability of the machine; or mapping a task on to a machine with less accomplishment time. Therefore, in order to carry out a better evaluation of the behaviour of the heuristic scheduling mappings, we need to make changes in the existing parameters and the simulation of various processing environments, whether homogeneous or heterogeneous.

2-1 Ant Colony System (ACS) algorithm

Most centralized computational applications in business and science can be demonstrated as a workflow. Each workflow can be designed and implemented on a multilayered basis. The upper layer involves the description of the workflow and the second layer gives an abstract view of the workflow. In this case, the workflow is shown as a set of services: $S_i = \{s_i^1, s_i^2, ..., s_i^{m_i}\}$. In paper [1] a scheduling module has been focused to propose an appropriate scheduling algorithm. The abstract model of the workflow will be as DAG(Directed Acyclic Graphs) where G=(V,A) , ‘n’ is the number of tasks and V={T1,T2,…,Tn} [1, 10].

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 set 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, 12, 13, 14, 15].

In reference [1], authors apply the ACS algorithm, which is one of the best ACO algorithms so far, to tackle the workflow scheduling problem in grid applications. Informally, the algorithm can be viewed as the interplay of the following procedures.

1) Initialization of algorithm, 2) Initialization of ants, 3) Solution construction, 4) Local pheromone updating, 5) Global pheromone updating, 6) Terminal test

Pheromone and heuristic information are the most important factors of an ACO algorithm. In general, pheromone is used to record the historical searching experiences and bias the ants' searching behaviour in future. On the other hand, heuristic information is some problem-based values to guide the search direction of ants. As the scheduling problem is mainly to map all tasks in the abstract workflow to service instances to form a concrete workflow, we denote the pheromone value of mapping service instance s_{ji} to task Ti as τ_{ij} , and the heuristic information value of mapping s_{ji} to Ti as η_{ij}.

2-2 Multiple-Objective Ant Colony Optimization (MOACO)

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 MOACO algorithms we refer to [3,9].

2-2-1 Available MOACO algorithms

In many practical problems, several conflicting criteria exist for evaluating solutions. In recent years, strong research efforts have been made to develop efficient algorithmic techniques for tackling such multi-objective optimization problems. Many of these algorithms are extensions of well-known metaheuristics. In particular, over the last few years, several extensions of ant colony optimization (ACO) algorithms have been proposed for solving multi-objective problems. These extensions often propose multiple answers to algorithmic design questions arising in a multi-objective ACO approach. However, the benefits of each one of these answers are rarely examined against the alternative approaches [3].

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].

In ACO problems with multiple objectives, there is more than one objective, therefore, their optimization all is of great importance. In such problems, objectives contradict each other. In other words, an improvement in one objective may result in worsening the others. To find optimum solutions to these problems is important. Therefore, some of the solutions might not be the best one as far as all the objectives are concerned. These solutions comprise a set of desired solutions [15, 17].

For MOACO algorithm, different suggestions have been offered and each one has its own design and competence functions and specific updates. Some of these algorithms have been presented in [17].

3. DEFINITION OF PROPOSED ALGORITHM

The problem of task scheduling will be studied in homogeneous computing environment. In this model, there is number of dependent tasks to allocation and number of machines to execute these tasks.

Given the importance and complexity of the decentralized and distributed environments, a fast and optimum heuristic scheduling algorithm may play a key role in improving the productivity of a system so that, taking the outstanding parameters in the grid environment into consideration, the scheduling of the tasks may be carried out optimally, the tolerance of the error of the system may increase, and eventually the output of the system may improve.

In this paper, we want to consider the parameters of cost and service quality time and to conduct the grid workflow scheduling using the ant colony method. Here we will define two kinds of optimizations for cost and time and three kinds of heuristic methods. Using these optimization and heuristic methods, we will define functions for local and general pheromone updating as well as competence functions to select the next task and calculate η. Moreover, since the formulas used in the previous paper were complicated as far as timing and calculations, new formulations have been presented here.

Fitness function for proposed algorithm will be mentioned.

Heuristic information of mapping is as follows:

1. The value of the heuristic information of mapping s_i^j on T_i for Cost is calculated with the following formula:

$$\eta_{ij} = CG_{ij} = \frac{\min\_cost}{Cost_{ij}} \tag{1}$$

Where $Cost_{ij}$ the cost of task execution and the cost comprise incurred to go from node i to j and min_cost involves the minimum cost to execute a task in grid workflow scheduling.

2. The value of the heuristic information of mapping s_i^j on T_i as η_{ij} for Time is calculated using the following formula:

$$\eta_{ij} = TG_{ij} = \frac{\min\_time}{Time_{ij}} \tag{2}$$

Where $Time_{ij}$ includes the execution time of task and the time spent to go from node i to j and min-time involves the minimum time to execute a task in grid workflow scheduling.

3. The value of the heuristic information of mapping s_i^j on T_i as η_{ij} for Time / Cost is:

$$\eta_{ij} = \frac{1}{2}(CG_{ij} + TG_{ij}) \tag{3}$$

In the proposed method, two kinds of constraints and two kind of optimization have been used that are defined as follows:

1. Time constraint: total time spent to select all the tasks which cannot be accepted if it exceeds the deadline. That is,

$$\sum_{i=1}^{n} s_i .t \leq Deadline \tag{4}$$

2. Cost constraint: cost incurred to select all the tasks which must be less than or within the budget. That is,

$$\sum_{i=1}^{n} s_i .c \leq Budget \tag{5}$$

3. Time optimization: to consider the cost constraint and be in a situation where the minimum possible time is spent.
4. Cost optimization: to consider the time constraint and be in a situation where the minimum possible cost is incurred.

Local updating of pheromone: Once an ant constructs its solution, it must update the existing pheromone on the trail. In doing so, the following function is used.

In the first step, τ_0 is considered a very small value. In the following steps, calculation of pheromone and the local updating of the formula will be as follows:

$$\begin{cases} \tau_{ij}^{new} = (1-\rho)\tau_{ij}^{old} + \rho\tau_0 & if \quad Task \quad Selecte \\ \tau_{ij}^{new} = (1-\rho)\tau_{ij}^{old} & else \end{cases} \tag{6}$$

Once all the ants have conducted the concerned timing and can select the possible states, general updating must be done. General updating is achieved through local updating and is based on the best state of the tasks selected.

$$\tau_{ij}^{new} = (1-\rho)\tau_{ij}^{old} + \rho\Delta\tau_{ij} \tag{7}$$

The value of $\Delta\tau_{ij}$ is equal with heuristic information (η_{ij}) in the states of cost and time optimization.

The main formula to select the next task is as follows:

$$\begin{cases} s_{ij} = \arg\max_{ij \in N} \tau_{ij}^{\alpha} \eta_{ij}^{\beta} & if \quad q \leq q_0 \\ Select \quad Random & else \end{cases} \tag{8}$$

In this selection rule, a random number q is generated and then compared the q_0; N is the collective of tasks that can be selected. Amount of η_{ij} is the specific problem heuristic information, and τ_{ij} the pheromone trails. α, weight of pheromone factor; β, weight of heuristic visibility factor; ρ, the evaporation or decay rate of pheromone; τ_0, threshold for pheromone at the beginning of the algorithm. The impact of these four factors has been studied thoroughly in the algorithm.

Used Data sets in this paper are data set 1 is as a complete graph with 51 nodes and the data set 2 is as a complete graph with 29 nodes.

We will consider algorithm administration as follows:

1. 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.
2. Calculating η values to calculate the mapping function.
3. Administering mapping function to select the next status and selection of q0.
4. If q<=q0, the best Task is selected regarding the mapping function. Otherwise, task selection will be carried out randomly.
5. Local pheromone update for all ants.
6. Returning to status 3, until all tasks are administered.
7. choosing the best status up to now
8. General pheromone update by using ants' local pheromone update
9. Administering the cycle from status 3 until all cycles finish.
10. print the best solution

Pseudo-code of the components of the proposed algorithm is as follows:

```
Proposed Algorithm
%this Algorithm for the scheduling problem with Time and Cost parameter with colony
system approach
% Initialization of elements
% Initialize of DAG matrix
% Initialize of Cost  and time matrix
%Heuristic Information (calculate the ETA)
%-----------------------------------------------------
Select the Heuristic and optimization 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 of Tasks scheduled
 for i=1:n
    for j=1:m_ant
              q = random ();
       if(q<=q0)
         S=taw.^alpha+ETA.^beta;
       else
         Select randomly one of the Tasks
       end
         %Local_Update pheromone
         if (Task selected)
           taw=(1-Row)*taw+Row*taw;
         else
            taw=(1-Row)*taw;
         end
       Refresh matrixes
   end
end
%***************************
calculated the min time and cost
Find best tour
value of -f- in
%Global_Update pheromone
taw_C(i,j)=(1-Row)*taw_C(i,j)+Row*ETA(i,j)
Print the best tour
end
```

4. SIMULATION AND ANALYSIS

To better evaluate the behaviour of mapping heuristics, a model of the execution times of the tasks on the machines is needed; so that the parameters of this model can be changed to investigate the performance of the heuristics under different computing systems and under different types of tasks to be mapped. In this part, at first, considering the feature s of the parameters, the behaviour and models of the grid environment, important parameters and different data, based on which the algorithms need to be analyzed, are developed and implemented. Then implementation of the previous algorithms and proposed ones is conducted in the programming and simulating environment. In the end, the results achieved through the implementation of the algorithms within the graphs, based on time and cost for the previous algorithms and proposed ones, are compared and evaluated.

Taking into consideration the formulation in the previous methods and the modifications made to them, and also the codification and simulation of them for all the conditions that have been

regarded for the next condition according to the value of η and pheromone evaporation and the three heuristics for cost, time and cost/time, results have been achieved for each one of the heuristic methods as follows. Comparison of the simulations conducted according to the graphs and the default values and the selection of each one of the heuristic kinds for each of the optimizations mentioned above for the proposed method indicates improvement.

A large number of parameters affect the proposed algorithm, some of which are of greater importance for the execution of the algorithm. In this paper, simulation and analysis are based on $\alpha = 0.4, \beta = 0.7, \ \rho = 0.06, \ q_0 = 0.5, \ \tau_0 = 0.03, \ number\ of\ ants = 100, \ iteration = 100$.
Optimization of all service quality parameters have been shown in Table 1 and 2 considering the heuristic conditions of TG (for time heuristic), CG (for cost heuristic) and TC (for cost/time heuristic).

Four controlling variables play important roles in the algorithm presented. In this part we have experiment results in figures 3-6 as previously mentioned for behaviours of cost, time optimization with Cost, time and time/Cost together. results show with every mentioned heuristics in proposed, adaptive ACS and MOACO algorithms in which proposed is optimal and have minimum cost and time compared with the two mentioned algorithms.

The results of the simulation are as follows:

Simulation for the cost optimization where data set 1 has been used is shown in Figure 3, Simulation for the time optimization where data set 1 has been used is shown in Figure 4, Simulation for the cost optimization where data set 2 has been used is shown in Figure 5 and Simulation for the time optimization where data set 2 has been used is shown in Figure 6.

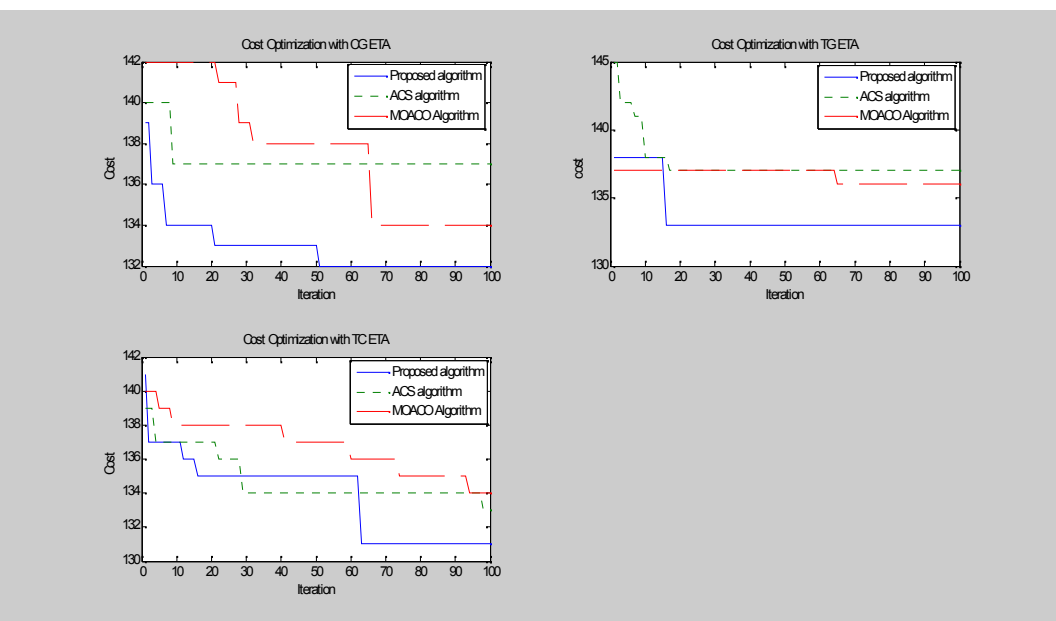

Fig3. Cost optimization with data set 1; cost heuristic (cg), time heuristic (tg), time / cost heuristic (tc)

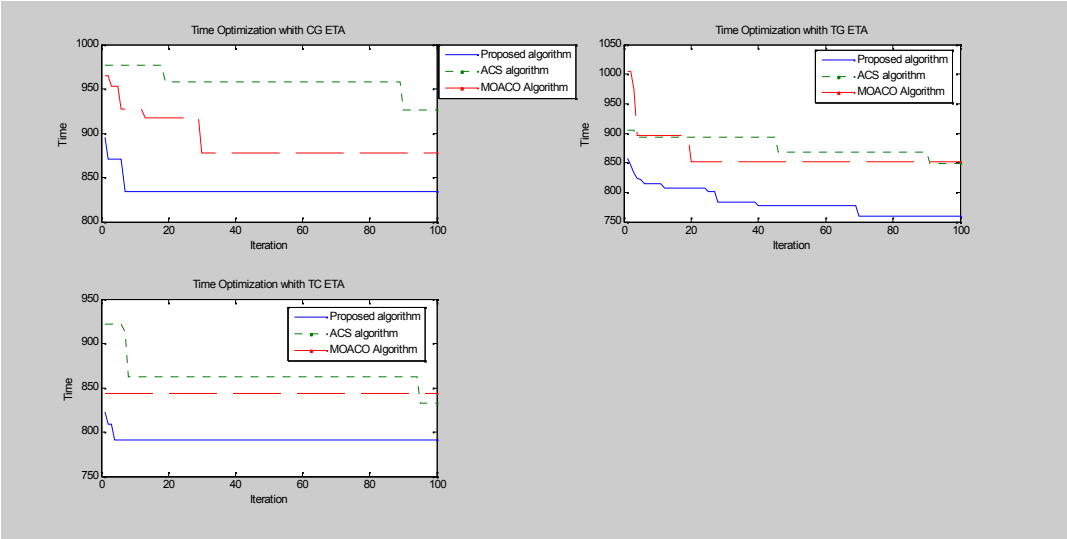

Fig4. Time optimization with data set 1; cost heuristic (cg), time heuristic (tg), time / cost heuristic (tc)

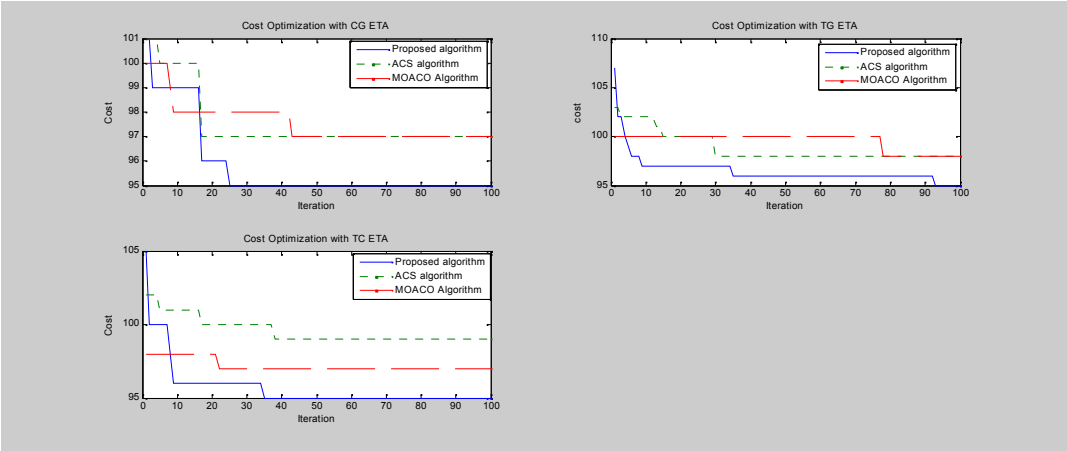

Fig5. Cost optimization with data set 2; cost heuristic (cg), time heuristic (tg), time / cost heuristic (tc)

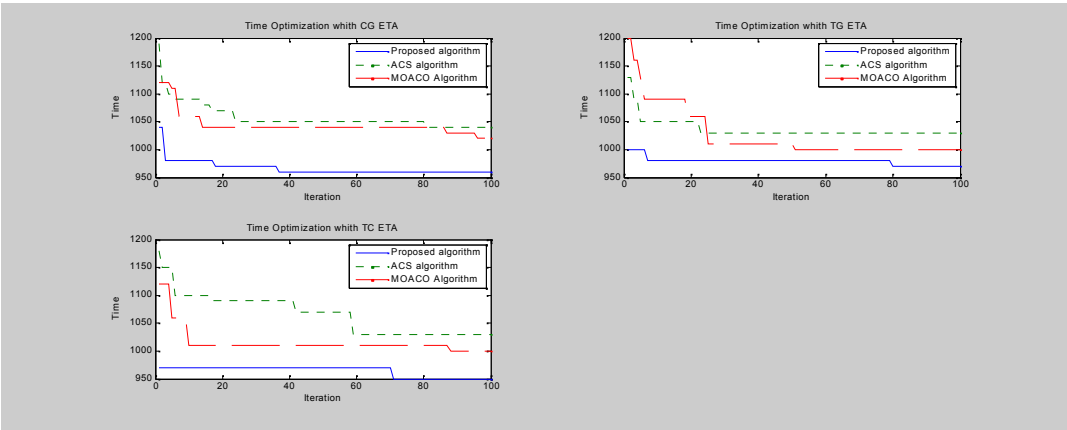

Fig6. Time Optimization with data set 2; Cost Heuristic (CG), Time Heuristic (TG), Time / Cost Heuristic (TC)

The results of simulation and tests for the two sets of data, data sets 1 and 2, have been also shown in Tables 1 and 2, where time optimization and cost optimization are considered as CG, TG, TC respectively for the cost heuristic, time heuristic, and time/cost heuristic together.

Table 1.Simulation results for data set 1

Cost Optimization	TG	CG	TC
Min cost in ACS Algorithm	137.00	137.00	133.00
Min cost in Proposed Algorithm	132.00	133.00	131.00
Min cost in MOACO Algorithm	134.00	136.00	134.00
improved Proposed Algorithm than ACS	3.65	2.92	1.50
improved Proposed Algorithm than MOACO	1.49	2.21	2.24
Time Optimization	TG	CG	TC
Min cost in ACS Algorithm	926.00	849.00	833.00
Min cost in Proposed Algorithm	834.00	759.00	791.00
Min cost in MOACO Algorithm	878.00	852.00	844.00
improved Proposed Algorithm than ACS	9.94	10.60	5.04
improved Proposed Algorithm than MOACO	5.01	10.92	6.28

Table 2.Simulation results for data set 2

Cost Optimization	TG	CG	TC
Min cost in ACS Algorithm	97.00	98.00	99.00
Min cost in Proposed Algorithm	95.00	95.00	95.00
Min cost in MOACO Algorithm	97.00	98.00	97.00
improved Proposed Algorithm than ACS	2.06	3.06	4.04
improved Proposed Algorithm than MOACO	2.06	3.06	2.06
Time Optimization	TG	CG	TC
Min cost in ACS Algorithm	1040.00	1030.00	1030.00
Min cost in Proposed Algorithm	960.00	970.00	950.00
Min cost in MOACO Algorithm	1020.00	1000.00	1000.00
improved Proposed Algorithm than ACS	7.69	5.83	7.77
improved Proposed Algorithm than MOACO	5.88	3.00	5.00

Regarded to results average improvement of the proposed algorithm as compared to ACS algorithm for the cost optimization is 2.69 and for the time optimization is 8.53, and the average improvement of the proposed algorithm as compared to MOACO algorithm for the cost optimization is 1.98 and for the time optimization is 7.40. For data set 2, the average improvement of the proposed algorithm as compared to adaptive ACS algorithm for the cost optimization is 3.05 and for the time optimization is 7.09, and the average improvement of the proposed algorithm as compared to MOACO algorithm for the cost optimization is 2.39 and for the time optimization is 4.63.

5. CONCLUSION

Results of the experiment indicate that the proposed algorithm is more improved, more efficient, and less complicated compared with the adaptive ACS and MOACO algorithms. The proposed algorithm has also reduced time and cost and has come to a better mapping function for the tasks and selection of next state. Proposed algorithm also reduced time and cost, increase efficiency

and achieve better mapping than previous algorithms in the grid environment. However, the proposed algorithm is of fewer time complexities and better efficiency compared with the previous algorithms. Also proposed algorithm in this paper is more efficient in the grid environment. The next work can be focused on the delay made in the distribution of ants and the existing delay in accountability and processing. In addition, the behaviour of these may be studied on heterogeneous and distributed environments.

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, "BPIAG: 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.Ghorbannia Delavar, 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 Murali bhaskaran.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. Murali Bhaskaran, "An Improved Min-Mean Heuristic Scheduling Algorithm for Mapping Independent Tasks on Heterogeneous 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(ISCSCT '10) Jiaozuo, P. R. China, 14-15,August 2010, pp. 179-182.

[8] Fatos Xhafa, Ajith Abraham, "Computational models and heuristic methods for Grid scheduling problems", Future Generation Computer Systems 26 (2010) 608_621.

[9] Rajesh Shrivastava, Shweta Singh and G. C. Dubey, "Multi Objective Optimization of Time Cost Quality Quantity Using Multi Colony", August, 2011

[10] Amit Agarwal, 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] Lianghong Wu,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, October 2010, pp.798–805 Available online at www.jseepub.com.

[12] Claudia Ruth Gatica, Susana Cecilia Esquivel, Guillermo Mario Leguizamon, "An ACO approach for the Parallel Machines Scheduling Problem", Inteligencia Artificial 46(2010), 84-95 doi: 10.4114/ia.v14i46.1550.

[13] 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, May 2008, p. 1246 – 1250.

[14] 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.

[15] M.H.M.A. Jahromi, R. Tavakkoli-Moghaddam, S.A. Jazayeri,R. Jafari and A. Shamsi. "Ant Colony Optimization for Multi-Objective Machine-Tool Selection and Operation Allocation in a Flexible Manufacturing System", World Applied Sciences Journal 15 (6): 867-872, 2011, ISSN 1818-4952 © IDOSI Publications, 2011.

[16] L. M. Nithya, A. Shanmugam, "Scheduling in Computational Grid with a New Hybrid Ant Colony Optimization Algorithm", European Journal of Scientific ResearchISSN 1450-216X Vol.62 No.2 (2011), pp. 273-281 © EuroJournals Publishing, Inc. 2011, http://www.eurojournals.com/ejsr.htm.

[17] C. Garcı ´a-Martı ´nez, O. Cordo ´n, F. Herrera, "A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP", European Journal of Operational Research 180 (2007) 116–148.

AUTHORS

Arash Ghorbannia Delavar received the MSc and Ph.D. degrees in computer engineering from Sciences and Research University, Tehran, IRAN, in 2002 and 2007. He obtained the top student award in Ph.D. course. He is currently an assistant professor in the Department of Computer Science, Payam Noor University, Tehran, IRAN. He is also the Director of Virtual University and Multimedia Training Department of Payam Noor University in IRAN. Dr.Arash Ghorbannia Delavar is currently editor of many computer science journals in IRAN. His research interests are in the areas of computer networks, microprocessors, data mining, Information Technology, and ELearning.

Javad Bayrampoor received the BS, in 2007 and now, he is a Student the MS degree in the department of Computer Engineering and Information Technology in Payam Noor University, Tehran, IRAN. His research interests include computer networks, grid, scheduling and meta-heuristic algorithm.

Ali Reza Khalili Boroujeni received the BS, in 2000 and now, he is a Student the MS degree in the department of Computer Engineering and Information Technology in Payam Noor University, Tehran, IRAN. His research interests include computer networks, grid and scheduling algorithm.

Ali Broumandnia received the Ph.D. degrees in computer engineering from Sciences and Research University, Tehran, IRAN, in 2006. He is currently an assistant professor in the Department of Computer Science of Islamic Azad University-South Tehran Branch. His research interests are in the areas of image processing, computer networks, computer architecture, operating systems and distributed systems.