

## An approach to building energy clusters using particle swarm optimization algorithm for allocating the tasks in computational grid

Rashedul Islam<sup>1</sup>, Md Nasim Akhtar<sup>2</sup>, Badlishah R Ahmad<sup>3</sup>, Utpal Kanti Das<sup>4</sup>, Mostafijur Rahman<sup>5</sup>, Zahereel Ishwar Abdul Khalib<sup>6</sup>

<sup>1,4</sup>Department of Computer Science and Engineering,

International University of Business Agriculture and Technology (IUBAT), Bangladesh

<sup>1,2</sup>Department of Computer Science and Engineering,

Dhaka University of Engineering & Technology (DUET), Bangladesh

<sup>3</sup>Faculty of Informatics and Computing, University Sultan Zainal Abidin (UniSZA), Malaysia

<sup>5</sup>Department of Software Engineering, Daffodil International University (DIU), Bangladesh

<sup>6</sup>School of Computer and Communication Engineering, University Malaysia Perlis, Malaysia

---

### Article Info

#### Article history:

Received Sep 24, 2018

Revised Nov 25, 2018

Accepted Dec 6, 2018

---

#### Keywords:

Computational grid

Energy cluster

Energy cluster header

Particle swarm optimization

---

### ABSTRACT

The proper mapping in case of allocation of available tasks among particles is a challenging job to accomplish. It requires proper procedural approach and effectual algorithm or strategy. The deterministic polynomial time for task allocation problem is relative. The existence of proper and exact approach for allocation problem is void. However, for the survival of the grid and executing the assigned tasks, the reserved tasks need to be allocated equally among the particles of the grid space. At the same time, the applied model for task allocation must not consume unnecessary time and memory. We applied Particle Swarm Optimization (PSO) for allocating the task. Additionally, the particles will be divided into three clusters based on their energy level. Each cluster will have its own cluster header. Cluster headers will be used to search the task into space. In a single cluster, particles member will be of same energy level status such as full energy, half energy, and no energy level. As a result, the system will use the limited time for searching task for the remaining tasks in it if a particular task requires allocating half task to a particle.

Copyright © 2019 Institute of Advanced Engineering and Science.  
All rights reserved.

---

### Corresponding Author:

Rashedul Islam,

Department of Computer Science and Engineering,

International University of Business Agriculture and Technology,

Dhaka- 1230, Bangladesh.

Email: rashed@iubat.edu

---

## 1. INTRODUCTION

Task allocation problem in the grid computing is a NP-Complete problem [1], [10]. No obvious solution is available for any NP-Complete problems. Task allocation in grid computing plays a vital role in terms of executing the job and further scheduling of the job.

The growth of system throughput is proportional to the effective task allocation model [2]. A smarter algorithm for task allocation's most advantageous result in polynomial time doubtfully exists [3]. Therefore, optimal solution strategies must be intelligent as well as fast and energy & memory saving.

PSO is an algorithm that follows a collaborative population-based search model [4]. Each individual node of the entire population called particles; those roam around in a 2D or 3D space. The objective of those particles is to look for the optimal solution within the multidimensional space. Particles have the capabilities

of adjusting and finding the best optimal point based on their own and neighborhood particles' knowledge. Based on that knowledge they move towards their own best position point or neighbors' best position point on the space. PSO act is calculated according to a predefined fitness function [5].

This paper is the upgraded version of [6]. Where we have shown how the fuzzy logic can be used for allocating the task in full and half status among the ants or particles with availability of full, half or no energy level status. However, we have not focused on effective searching technique for finding the task within the grid in a short amount of time. In this paper, we are going to use PSO as our searching technique of available task as well as we will show how the cluster depends on energy level of the particles effects the task allocation procedure.

## 2. LITURATURE REVIEW

In [7], heuristic algorithm called PSO algorithm is proposed for the task assignment problem for homogeneous distributed computing systems. The performance of PSO algorithm is evaluated in comparison with well-known GA algorithm for a number of randomly generated mapping problem instances. The results showed that the PSO algorithm solution quality is better than that of GA in most of the test cases. An improved discrete particle swarm algorithm is proposed in [8] to solve task assignment problem in distributed computer system. In [8], they redefine the coefficient multiplying velocity operator in discrete particle swarm with probabilistic selection for velocity and derive a redefined position updating formula. The position vector is transformed from the continuous values to the discrete values based on SPV (small position value) rules, accordingly, a permutation formed in [9]. Approach of [10] is to generate an optimal schedule to complete the tasks in a minimum time as well as utilizing the resources in an efficient way. A version of Discrete Particle Swarm Optimization (DPSO) algorithm for grid job scheduling is used in [10]. Scheduler of [10] aims at generating feasible solutions while minimizing makespan and flowtime simultaneously.

According to [11], a pheromone-based task allocation and coordination control scheme was introduced. Based on pheromone technique, a prototype implementation system is established to coordinate and control a hybrid shop floor control system. A load balanced task allocation algorithm on social spider optimization is proposed in [12]. The algorithm first balances the load before it schedules the transactions to the appropriate nodes in the grid environment.

## 3. FORMAL PROBLEM DEFINATION AND BACKGROUND

We have shown each individual particle search for the task in the space. However, this approach has a problem lies in it. Suppose, system is planned to allocate a "FULL TASK" status task T among the particle. A particle P with half energy level status arrived and requested for the task. According to algorithm, the existing full task will be split into half ( $T/2$ ) and will be allocated to P. For rest of the half task, system will again start searching for allocating rest of the half part. According to this approach, for allocating a single task system need to search twice and that is wastage of time and memory of the grid.

## 4. PROPOSED METHOD

We have shown how to allocate the task within a minimum time but for searching the tasks effectively, we have not shown any approach. The PSO is the one of the best solution for searching the task within a grid. Many models are been proposed already that we mentioned in our literature review section.

According to our proposed model. The particles or nodes are in different states of energy level. Therefore, by using PSO we are going to make three clusters based on the energy level of the particles. The clusters will be Full Energy Cluster (FEC), Half Energy Cluster (HEC) and No Energy Cluster (NEC). Each cluster will have a header called Energy Cluster Header (ECH). At the global neighborhood level, the particles will individually search for the geographical head of the cluster. The head will store the information about the energy level of the cluster. Initially three heads will be defined by the system. One head will be responsible for gathering the particles who has the full energy available. Rest of the two heads will store the information of all particles that is in half and no energy level state respectively.

Primarily, based on the energy level, a particular particle will search for the geographical head that is shown in Figure 1. Within a period, all particles will be gathered at their designated cluster shown in Figure 2. Then the cluster will search for the task to be assigned from the system shown in Figure 3.

The particles will follow the movement of its own geographical head. When a head of a cluster will be assigned a task, by using fuzzy logic a full task can be divided into two similar half tasks shown in Figure 4.

Let us assume a scenario, where we have a geographical head for half energy level particle  $h_{half}$ . Under  $h_{half}$  there are three particles available, they are  $p_1$ ,  $p_2$  and  $p_3$ . The  $h_{half}$  searched for the “full task” level task  $t_1$  using PSO. However, the particles have half energy available in them. Therefore, at this moment  $t_1$  will be split into half two tasks and will be assigned to  $p_1$  and  $p_2$ . Now both particles  $p_1$  and  $p_2$  are in No Energy Level state. As a result, they will start searching for the No Energy Level geographical head and these procedures will be repeated and continued until all tasks are finished executing by the particles.

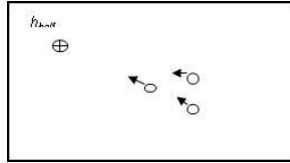


Figure 1. Three particles searching for HEC

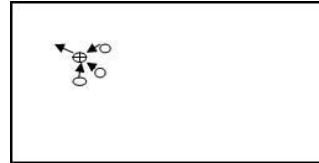


Figure 2. Three particles found HEC

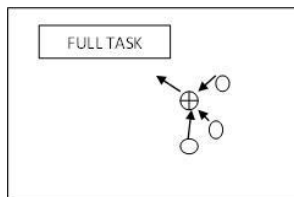
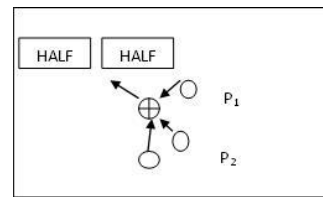


Figure 3. HEC searched and found a FULL TASK

Figure 4. Because of HEC, Task splits in half and all parts of the tasks assigned to  $p_1$  and  $p_2$  to execute

## 5. METHODOLOGY

### a) PSO for Particles

According to our proposed method, initially the particles will be randomly deployed into the space. Then they search for the Energy Clusters Head (ECH) based on their energy level. At the beginning all the particles will consist of full energy. Therefore, they start searching for the Full Energy Cluster Head (FECH). After converging, the FECH starts searching for available tasks available into the space. At this section, we are using the simple PSO algorithm for searching ECH. Algorithm is demonstrated in Figure 5.

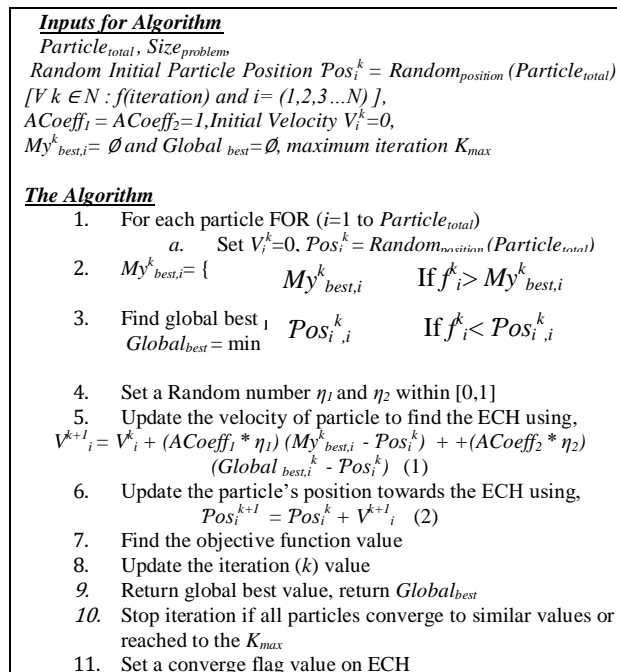


Figure 5. Algorithm for searching ECH by particles

According to the algorithm of Figure 5, all the particles reach towards their designated ECH. If all particles reached to the ECH, then the ECH start searching for tasks using PSO algorithm.

b) PSO for ECH

ECH has particles as followers. The movement of the particles is dependable on the movement of ECH. Based on the flag value ECH understands that all possible particles has arrived to the cluster and now start searching for task start using following PSO algorithm demonstrated in Figure 6.

<p><b><u>Inputs for Algorithm</u></b>  <i>Particle<sub>total</sub>, Size<sub>problem</sub></i>  <i>Random Initial ECH Position <math>Pos_i^k = Random_{position} (Particle_{total})^n [V k \in N : f(iteration) \text{ and } i = (1, 2, 3 \dots N)]</math>,</i>  <i>ACoeff<sub>1</sub> = ACoeff<sub>2</sub> = 1,</i>  <i>Initial Velocity <math>V_i^k = 0, My_{best,i}^k = \emptyset</math> and <math>Global_{best} = \emptyset</math>,</i>  <i>maximum iteration <math>K_{max}</math>, task status <math>Task_i</math> where <math>Task_i = \{full \text{ task } F_F, \text{ half task } F_H\}</math>,</i>  <i>energy level status of particle <math>PAR_{engr} = \{Full \text{ Energy } Engr_F, \text{ Half Energy } Engr_H \text{ and No Energy } Engr_N\}</math>, fuzzy (<math>Task_i, PAR_{engr}</math>)</i></p>
<p><b><u>The Algorithm</u></b></p> <ol style="list-style-type: none"> <li>1. For each ECH FOR (<math>i=1</math> to <math>Particle_{total}</math>)             <ol style="list-style-type: none"> <li>a. Set <math>V_i^k = 0, Pos_i^k = Random_{position} (Particle_{total})</math></li> </ol> </li> <li>2. <math>My_{best,i}^k = \{ \quad My_{best,i}^k \quad \text{If } f_i^k &gt; My_{best,i}^k</math></li> <li>3. Find global be <math>Pos_{i,i}^k \quad \text{If } f_i^k &lt; Pos_{i,i}^k</math></li> </ol> <p style="text-align: center;"><math>Global_{best} = \min \{ My_{best,i}^k \}</math></p> <ol style="list-style-type: none"> <li>4. Set a Random number <math>\eta_1</math> and <math>\eta_2</math> within <math>[0,1]</math></li> <li>5. Update the velocity of particle to find the ECH using,  <math display="block">V_i^{k+1} = V_i^k + (ACoeff_1 * \eta_1) (My_{best,i}^k - Pos_i^k) + (ACoeff_2 * \eta_2) (Global_{best,i}^k - Pos_i^k) \quad (3)</math> </li> <li>6. Update the particle's position towards the ECH using,  <math display="block">Pos_i^{k+1} = Pos_i^k + V_i^{k+1} \quad (4)</math> </li> <li>7. Find the objective function value</li> <li>8. Update the iteration (<math>k</math>) value</li> <li>9. Return global best value, return <math>Global_{best}</math></li> <li>10. Stop iteration if all particles converge to similar values or reached to the <math>K_{max}</math></li> <li>11. Get the task from system and find the status of the task &amp; required energy level.</li> <li>12. Distribute the assigned tasks using fuzzy (<math>Task_i, PAR_{engr}</math>)</li> </ol>

Figure 6. Algorithm for searching and distributing tasks by ECH

Under the computation grid, we have assigned forty swarm particles. In simulation environment, the maximum iteration was set to 100. Upper bound and lower bound was 1.0 and 0.0 respectively.

## 6. RESULT AND DISCUSSION

For implementation our algorithm we coded basic particle swarm optimization algorithm using JAVA. Run under a machine with configuration of Intel Pentium (R) CPU G630 @ 2.70GHz, 2GB memory and OS Windows 10. To measure the efficient searching of the task we measured the iteration number for converging to the similar values by particles and ECHs.

The iteration number is high if we deploy individual particles for searching tasks among the space. On the other hand, the iteration number decreases in terms of using the ECHs for searching the tasks on the space. For Individual Particle following function is used and tried to find x and y that minimize the function:

$$f(x, y) = (3.1482 - x + x * y^4)^2 + (2.81 - x + x * y^2)^2 + (3.2 - x + x * y)^2 \quad (5)$$

Where,  $1 \leq x \leq 4$ , and  $-1 \leq y \leq 1$ .

The simulation is done using Eclipse, Version: Neon.3 Release (4.6.3). For both algorithm the setup of individual particle is same. However, space is different that is defined by functions. Setup for Individual Particles:

- Swarm Particles = 40
- Maximum Iteration= 100
- Dimension of the Problem = 2;
- Value  $C_1 = 1.0$ ;
- Value of  $C_2 = 1.0$ ;
- Upper bound Value = 1.0;
- Lower bound Value = 0.0;

The Figures 7. and 8., show the execution trace of PSO algorithm for individual particle. The simulation took ninety-nine iterations to find the converging point. At 99<sup>th</sup> iteration the best of X is 2.9769656985602913 and Y is -0.08295184247553182. For ECH, we used following function and tried to find x and y that minimize the function:

$$f(x, y) = (3.229 - x + x * y^4)^2 + (2.528 - x + x * y^2)^2 + (2.29 - x + x * y)^2 \quad (6)$$

Where  $1 \leq x \leq 4$ , and  $-1 \leq y \leq 1$

The Figures 9. and 10. show the execution trace of PSO algorithm for ECH particle. The simulation took eighty-four iterations to find the converging point. At 84<sup>th</sup> iteration, the best of X is 3.115943660339216 and Y is 0.31551574685964867.

```

ITERATION 0:
  Best X: 3.2038773837611463
  Best Y: 0.05751476216198248
  Value: 0.18254054021304283
ITERATION 1:
  Best X: 3.2038773837611463
  Best Y: 0.05751476216198248
  Value: 0.18254054021304283
ITERATION 2:
  Best X: 3.1494665776061512
  Best Y: 0.04522497682175752
  Value: 0.14814384667330555
ITERATION 3:
  Best X: 3.0196336345755554
  Best Y: -0.033801158045306826
  Value: 0.06517279335951089
ITERATION 4:
  Best X: 3.0196336345755554
  Best Y: -0.033801158045306826
  Value: 0.06517279335951089
ITERATION 5:
  Best X: 3.0196336345755554
  Best Y: -0.033801158045306826
  Value: 0.06517279335951089

```

Figure 7. Sample of iterations from 0 to 5 when individual Particles searching for tasks

```

ITERATION 96:
  Best X: 2.9769656984743076
  Best Y: -0.08295184228739323
  Value: 0.05139792381024267
ITERATION 97:
  Best X: 2.9769656984743076
  Best Y: -0.08295184228739323
  Value: 0.05139792381024267
ITERATION 98:
  Best X: 2.9769656984743076
  Best Y: -0.08295184228739323
  Value: 0.05139792381024267
ITERATION 99:
  Best X: 2.9769656984743076
  Best Y: -0.08295184228739323
  Value: 0.05139792381024267

Solution found at iteration 99, the solutions
  Best X: 2.9769656984743076
  Best Y: -0.08295184228739323

```

Figure 8. Sample of iterations from 96 to 99, X & Y values and solution for Best X and Y when individual particle searching for tasks

```

ITERATION 0:
  Best X: 3.3017358808793635
  Best Y: 0.4139472915500093
  Value: 0.16987112758869372
ITERATION 1:
  Best X: 3.3017358808793635
  Best Y: 0.4139472915500093
  Value: 0.16987112758869372
ITERATION 2:
  Best X: 2.9034408292700427
  Best Y: 0.25695400249496775
  Value: 0.1657360671925348
ITERATION 3:
  Best X: 3.1853846250773206
  Best Y: 0.28335111972856475
  Value: 0.16547909159257737
ITERATION 4:
  Best X: 3.140026259266341
  Best Y: 0.37882550523987135
  Value: 0.16491476297317303
ITERATION 5:
  Best X: 3.140026259266341
  Best Y: 0.37882550523987135
  Value: 0.16491476297317303
    
```

Figure 9. Sample of iterations from 0 to 5 when ECH searching for tasks

```

ITERATION 80:
  Best X: 3.1159436569252708
  Best Y: 0.31551574645750446
  Value: 0.12257047563535864
ITERATION 81:
  Best X: 3.1159436569252708
  Best Y: 0.31551574645750446
  Value: 0.12257047563535864
ITERATION 82:
  Best X: 3.1159436569252708
  Best Y: 0.31551574645750446
  Value: 0.12257047563535864
ITERATION 83:
  Best X: 3.1159436569252708
  Best Y: 0.31551574645750446
  Value: 0.12257047563535864
ITERATION 84:
  Best X: 3.1159436569252708
  Best Y: 0.31551574645750446
  Value: 0.12257047563535864

Solution found at iteration 84, the solutions is:
  Best X: 3.1159436569252708
  Best Y: 0.31551574645750446
    
```

Figure 10. Sample of iterations from 80 to 84, X & Y values and solution for best X and Y when ECH searching for tasks

Result shows the ECH particles can find the task faster than the individual particles. In case of individual particles, when it gets the task, in the case it will have limited memory.

If a task required full energy level to complete and the particle has only half energy available, then the optimal solution of task allocation not possible to achieve. Because, the system needs to waste time to find another particle that has half or full energy state.

On the other hand, the main benefit of ECH is, system does not need to waste time for searching the other particles for assigning the task. In this situation, system will easily find another desired particle within the cluster.

If the cluster is Full Energy Cluster (FEC) than system does not need to find another particle, because task required full energy. On the other hand, if the cluster is Half Energy Cluster (HEC), then system will be able to assign rest of the half task to another particle that is the member of that same HEC.

Our proposed cluster based algorithm is 15.16% efficient that is shown in Figure 11. The blue bar represents the number of particles that is 40 and red bar represents the total time requires in second to allocate tasks among those 40 particles. The algorithm without the energy cluster system needs 99 seconds (assume, 1 iteration = 1 second). On the other hand, the algorithm with the energy cluster system needs 84 seconds. Therefore, the result indicates that task allocation can be completed within a minimum number of iteration and short time.

On the other hand, Figure12 clearly illustrates that our EC based PSO algorithm is performing better than Discrete Particle Swarm Optimization (DPSO) and PSO. Where DPSO and PSO needed 50 and 7.9 iterations but our proposed EC based PSO needed only 2.1 iterations for each particle.

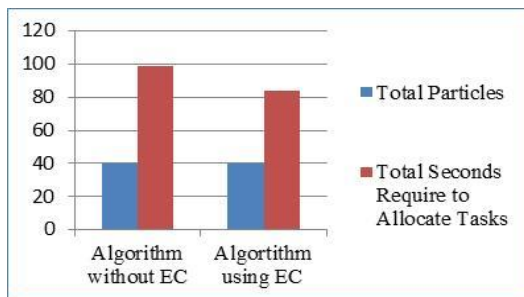


Figure 11. Efficiency of the proposed algorithm using Energy Cluster (EC)

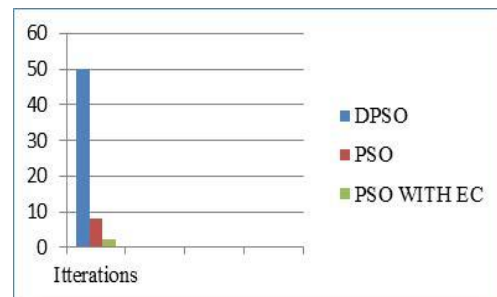


Figure 12. Comparison of number of iteration for each particle with other PSO Algorithms

## 7. CONCLUSION

In this paper, we build an algorithm where initially particles try to find the header of different energy status based cluster within the space. Then energy clusters start searching and executing the tasks. During allocation, system does not waste time to find another particle with desired energy level within the space.

The particle, which satisfies the energy level status, will be found as a member of the energy cluster. So, the task allocation among the particles become efficient and time consumption is minimized effectively. Further research can be done for allocating task in three-dimensional spaces. We did not focus on the issue whether task allocation can be done by system itself or by the energy cluster header.


## ACKNOWLEDGEMENTS

I would like to express my most sincere gratitude to my supervisor Prof. Dr. Nasim Akhtar for his continuing support, professional guidance and for giving me an opportunity to work with him. I also thank him for his very careful reading and insightful suggestions throughout the research period. I am indebted to him for the optimization knowledge he gave to me on my research. Special and deepest gratitude to all the co-authors for their suggestion, guidance and time towards the development of this research.

## REFERENCES

- [1] K. Pathak, and G. Vahinde, "Comparison of particle swarm optimization and genetic algorithm for load balancing in cloud computing environment." *Technology (IJRCIT)* vol. 1, no. 1, 2015.
- [2] Kennedy, James, and Russell Eberhart. "Particle swarm optimization." In *Neural Networks, 1995. Proceedings. IEEE International Conference on*, vol. 4, pp. 1942-1948. IEEE, 1995.
- [3] Li, Hui, Lifeng Wang, and Jianhong Liu. "Task scheduling of computational grid based on particle swarm algorithm." In *Computational Science and Optimization (CSO), 2010 Third International Joint Conference on*, vol. 2, pp. 332-336. IEEE, 2010.
- [4] H. Liu, A. Abraham, and A. E. Hassanien, "Scheduling jobs on computational grids using a fuzzy particle swarm optimization algorithm," *Future Generation Computer Systems*, vol. 26, no. 8, pp. 1336–1343, 2010.
- [5] T. Ma, Q. Yan, W. Liu, D. Guan, and S. Lee, "Grid Task Scheduling: Algorithm Review," *IETE Technical Review*, vol. 28, no. 2, pp. 158–167, 2011.
- [6] M. R. Islam and M. N. Akhtar, "Fuzzy logic based task allocation in ant colonies under grid computing," 2017 *International Conference on Electrical, Computer and Communication Engineering (ECCE)*, 2017.
- [7] K. Pathak, and G. Vahinde, "Comparison of particle swarm optimization and genetic algorithm for load balancing in cloud computing environment." *Technology (IJRCIT)* vol. 1, no. 1, 2015.
- [8] Q. Yang, C. Wang, and C. Zhang, "An efficient discrete particle swarm algorithm for Task Assignment Problems," 2009 *IEEE International Conference on Granular Computing*, 2009.
- [9] Zhang, Lei, Yuehui Chen, Runyuan Sun, Shan Jing, and Bo Yang. "A task scheduling algorithm based on PSO for grid computing." *International Journal of Computational Intelligence Research*, vol. 4, pp. 37-43, no. 1, 2008.
- [10] H. Izakian, B. T. Ladani, K. Zamanifar, and A. Abraham, "A Novel Particle Swarm Optimization Approach for Grid Job Scheduling," *Information Systems, Technology and Management Communications in Computer and Information Science*, pp. 100–109, 2009.
- [11] L. Wang, D.-B. Tang, W.-D. Yuan, W.-B. Gu, and D.-S. Tang, "Pheromone-Based Coordination and Control for Task Allocation," *International Conference on Information Engineering and Computer Science*, pp. 1–4, 2009.
- [12] D. P. Mahato and R. S. Singh, "On maximizing reliability of grid transaction processing system considering balanced task allocation using social spider optimization," *Swarm and Evolutionary Computation*, 2017.
- [13] O. Goldreich, "NP-Completeness," P, NP, and NP-Completeness, pp. 96–141, 2010. R. W. Lucky, "Automatic equalization for digital communication," *Bell Syst. Tech. J.*, vol. 44, no. 4, pp. 547–588, Apr. 1965.

## BIOGRAPHIES OF AUTHORS

	<p>Md. Rashedul Islam has completed his graduation from the IUBAT—International University of Business Agriculture and Technology in Computer Science and Engineering (2009). Since January 2010 he is working as a Senior Lecturer at the IUBAT—International University of Business Agriculture and Technology. His interested research areas are Distributed Algorithm, Distributed Systems, and Wireless Sensor Networks.</p>
---	---



	<p>Md. Nasim Akhtar received the M.Eng and Ph.D degrees from National Technical University of Ukraine, Kiev, Ukraine and Moscow State Academy of Fine Chemical Technology, Russia, in 1998 and 2010, respectively. Currently, he is a Professor in the Department of Computer Science and Engineering, Dhaka University of Engineering and Technology (DUET), Gazipur, Bangladesh. His research interests include Distributed Data Warehouse System On Large Clusters, Digital Image Processing and Water Marking, Peer to Peer Networking, Cloud Computing, Operating System. He has presented papers at conferences both home and abroad, published articles and papers in various journals.</p>
	<p>R. Badlishah Ahmad received the M.Sc Engineering and Ph.D degrees from University of Strathclyde, UK in 1995 and 2000, respectively. Currently, he is a Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin (UniSZA). His research interests in Computer and Telecommunication Network Modelling include WSN and Optical Network using discrete event simulators (OMNeT++), Optical Networking and Embedded System based on GNU/Linux. He has presented papers at conferences both home and abroad, published articles and papers in various journals.</p>
	<p>Utpal Kanti Das is Coordinator and Professor of Department of Computer Science and Engineering at International University of Business Agriculture and Technology. Mr Das received the Ph.D from Jahangirnagar University, Bangladesh and both BSc and MSc from Madras, India. Currently he is researching on Transportation Management. Other interested research areas are AI and applied mathematics.</p>
	<p>Mostafijur Rahman completed his BSc in Computer Science from National University of Bangladesh (2003). He Pursued his MSc (2009) and PhD (2017) in Computer Engineering, from UNIMAP, Malaysia. He worked as Lecturer since 2009 to September, 2017 for School of Computer and Communication Engineering in UNIMAP. Currently he is serving as Assistant Professor in the Department of Software Engineering at Daffodil International University (DIU), Bangladesh. His research interest in Software Testing, Multimedia and Creativity in Medical Science, Computer Security, Cloud Computing, Algorithm Optimization, Parallel and Distributed System, Device Driver for GNU/Linux based embedded OS. He has presented papers at conferences both home and abroad, published articles and papers in various journals.</p>
	<p>Zahereel Ishwar Abdul Khalib completed his Bachelor Degree in Computer Engineering from California State University, Sacramento (2000). Pursuit for MSc in Real Time Software Engineering, UTM (2005) and PhD in Computer Engineering, Unimap (2013)- Real Time Scheduling for soft real time application. Currently lecturing as Senior Lecturer at Unimap. Research interest in Parallel Programming Optimization for Multicore Embedded platform, Real Time Multimedia Applications/Architecture, Intelligent of Things Devices and Real Time Open Source Embedded System in general.</p>