

Multi-Objective Task Scheduling in the Cloud Computing based on the Particle Swarm Optimization

Farnaz Sharifi Milani

Department of Computer, East Azarbaijan Science and Research Branch, Islamic Azad University, Tabriz, Iran
Email: f.sharifi@iauasrb.ac.ir

Ahmad Habibzad Navin*

Department of Computer, East Azarbaijan Science and Research Branch, Islamic Azad University, Tabriz, Iran
Email: habibzad@iauasrb.ac.ir

Abstract— Cloud computing is the latest emerging trend in distributed computing, where shared resources are provided to end-users in an on demand fashion that brings many advantages, including data ubiquity, flexibility of access, high availability of resources, and flexibility. In this type of systems many challenges are existed that the task scheduling problem is one of them. The task scheduling problem in Cloud computing is an NP-hard problem. Therefore, many heuristics have been proposed, from low level execution of tasks in multiple processors to high level execution of tasks. In this paper, we propose a new algorithm based on PSO to schedule the tasks in the Cloud. The results demonstrated that the proposed algorithm has a better operation in terms of task execution time, waiting time and missed tasks in comparison of First Come First Served (FCFS), Shortest Process Next (SPN) and Highest Response Ratio Next (HRRN).

Index Terms— Cloud Computing, Scheduling, PSO, Network

I. INTRODUCTION

The “Cloud” the metaphor is a reference to the ubiquitous availability and accessibility of computing resources via Internet technologies [1]. Cloud computing as a novel and entirely internet-based approach is an important concept in today’s distributed systems [2]. It is used chiefly for business applications in which computers cooperate to perform a specific service [3]. It evolved through the recent advancements in hardware, virtualization technology, distributed computing, and service delivery over the Internet. While Cloud computing may not involve a lot of new technologies, it certainly represents a new way of managing IT. In many cases, this will not only change the workflow within the IT organization, it will often result in a complete reorganization of the IT department. Cost savings and scalability can be highly achieved from cloud computing [4]. Cloud computing, which is aimed at providing infrastructures, platforms and software as services has been introduced and implemented in the last few years. It is widely recognized as the next generation of computing architecture [5]. Generally, Cloud computing services can be categorized into three main types of services:

Infrastructure as a Service, Platform as a Service and Software as a Service. These services can then be accessed through a cloud client which could be a web browser, mobile app, and so on [6]. Cloud computing provides implementation agility, lower capital expenditure, location independence, resource pooling, broad network access, reliability, scalability, elasticity, and ease of maintenance [7, 8]. Also it offers numerous advantages for data and software sharing and thus making the management of complex IT systems much simpler [9, 10].

Job scheduler is a vital part of any distributed system like Grid [11-13], Cloud [14-17] and P2P networks [18-20] which assigns jobs to suitable resources for execution. The scheduling of a task workflow in a distributed computing platform is a well-known NP-hard problem [21]. The problem is even more complex and challenging when the virtualized clusters are used to execute a large number of tasks in a Cloud computing platform [21, 22]. For this reason, many heuristics have been proposed, from low level execution of tasks in multiple processors to high level execution of tasks in Grid and Cloud environments [23]. Recently many papers are published which used evolutionary algorithms like genetic, ant colony, bee colony and PSO (Particle Swarm Optimization) for optimization problems[21]. Because of the success of the PSO algorithms, in this paper we propose a new method based on the PSO algorithm for task scheduling in Cloud computing. The proposed method use a multi-objective function for increasing the effectiveness of algorithm.

The rest of this paper is organized as follow. In section 2, the related works are briefly reviewed. Section 3 describes the proposed method. In section 4 the obtained results are presented. Finally, last section concludes the paper and suggests some suggestions for future researches.

II. RELATED WORKS

In this section, the state of the art mechanisms and approaches of task scheduling in the Cloud systems are briefly reviewed and analyzed based on published years.

- 2011

Garg, Yeo [24] have proposed the near-optimal scheduling policies to exploit heterogeneity across multiple data centers for a Cloud provider. The method is considered a number of energy efficiency factors (such as energy cost, carbon emission rate, workload, and CPU power efficiency) which change across different data centers depending on their location, architectural design, and management system. They have demonstrated that the proposed method can able to achieve on average up to 25% of energy savings in comparison to profit based scheduling policies. Mezmaiz, Melab [25] have investigated the problem of scheduling precedence-constrained parallel applications on heterogeneous computing systems like cloud computing infrastructures. They have proposed a new parallel bi-objective hybrid genetic algorithm based on dynamic voltage scaling (DVS) that takes into account, not only makespan, but also energy consumption.

- 2012

Li, Qiu [26] have proposed two online dynamic resource allocation algorithms for the IaaS Cloud system with preemptable tasks. They algorithms adjusted the resource allocation dynamically based on the updated information of the actual task executions and the experimental results demonstrated that the proposed algorithms can significantly improve the performance in the situation where resource contention is fierce. Also Abrishami and Naghibzadeh [27] have proposed a new QoS-based workflow scheduling algorithm based on a novel concept called Partial Critical Paths (PCP), which tries to minimize the cost of workflow execution while meeting a user-defined deadline. Lu, Jiang [28] have proposed a resources collaboration scheduling model to improve the efficiency of the virtual resources collaboration scheduling. The proposed model is based on virtual organization and makes use of the trust mechanism to estimate the credibility of the virtual organization. Finally, Wang, Zeng [29] have presented a Bayesian method based cognitive trust model, and a trust dynamic level scheduling algorithm named Cloud-DLS (Dynamic Level Scheduling) by integrating the existing DLS algorithm [30].

- 2013

Van den Bossche, Vanmechelen [31] have proposed a workload model which is considers non-preemptible and non-migratable workloads with a hard deadline that are characterized by CPU, memory and data transmission requirements. Also, Wang, Wang [32] have proposed a genetic algorithm for Cloud resource optimization scheduling model that promised the user needs while optimizing resource allocation. Laili, Tao [14] have combined the Service Composition Optimal Selection (SCOS) and Optimal Allocation of Computing Resources (OACR) into one-time decision in one console, named Dual Scheduling of Cloud Services and Computing Resources (DS-CSCR). Wu, Liu [16] have proposed a market-oriented hierarchical scheduling strategy in Cloud workflow systems.

- 2014

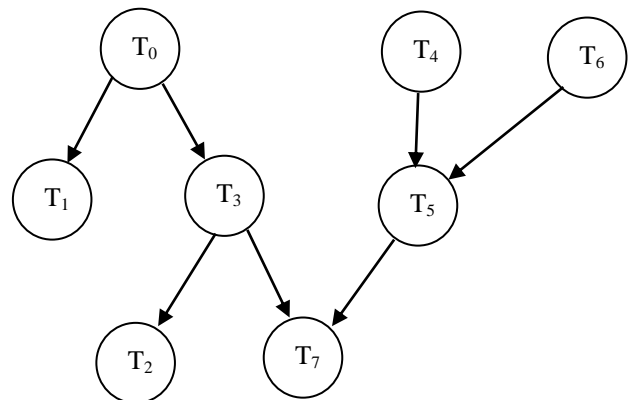
Fr êncu [33] have presented a solution based on genetic algorithm to finding the optimal number of component types needed on nodes so that every type is present on every allocated node. They have tested the algorithm in terms of node load, closeness to the optimal solution and proved the algorithm's efficiency. Tao, Feng [15] have proposed Case Library and Pareto Solution based hybrid Genetic Algorithm CLPS-GA for Optimal Scheduling of Computing Resources (OSCR). Liu, Qu [34] have proposed a fuzzy clustering method to effectively preprocess the Cloud resources. They have proposed a new directed acyclic graph based scheduling algorithm called earliest finish time duplication algorithm for heterogeneous Cloud systems by combining the list scheduling with the task duplication scheduling scheme. Finally, the experiment demonstrates that the proposed algorithm is effectual and well-organized. Finally, Jung, Lim [35] have presented the workflow scheduling scheme that reduces the task waiting time when an instance occurs the out-of-bid situation.

III. PROPOSED METHOD

In this section we propose a new method based on PSO for task scheduling in Cloud computing. First, the problem statement is described.

A. Problem Statement

For stating the problem we use a DAG (Directed Acyclic Graph) which is a directed graph with no directed cycles. It is formed by a collection of vertices and directed edges, each edge connecting one vertex to another, such that there is no way to start at some vertex v and follow a sequence of edges that eventually loops back to v again [36, 37].



$T_0=\{8, 20, M\},$	$T_1=\{6, 12, L\}$
$T_2=\{7, 15, H\},$	$T_3=\{6, 15, H\}$
$T_4=\{8, 18, L\},$	$T_5=\{4, 14, H\}$
$T_6=\{7, 20, M\},$	$T_7=\{6, 10, M\}$

Fig. 1. An example of a DAG with 8 tasks and three resources

In a DAG vertices denote tasks and edges represent precedence constraints and/or data movements among tasks. Given a set of resources, a *schedule* for a DAG is an assignment which specifies the mapping of tasks and

resources and the estimated start time of each task on the mapped resource. Here, the aims of DAG scheduling is to minimize the overall execution time, waiting time and the missed tasks. Fig. 1 represents an example of DAG. In this example, three resources and eight tasks are exist. Each task have three properties {ET¹, deadline, priority}.

B. Swarm Intelligence

Swarm intelligence refers to collective intelligence. Biologists and natural scientist have been studying the behavior of social insects due to their efficiency of solving complex problems such as finding the shortest path between their nest and food source or organizing their nests. In spite of the fact that these insects are unsophisticated individually, they make wonders as a swarm by interaction with each other and their environment. In last two decades, the behaviors of various swarms that are used in finding preys or mating are simulated into a numerical optimization technique. The collective intelligence emerges from a self-organization process of agents evolving autonomously according to a set of internal rules specifying its motion patterns and interaction with the environment and other agents, such that intelligent collective behavior arises from simple individual behaviors [38]. Some important swarm intelligence techniques are ant colony optimizer, particle swarm optimizer, artificial bee colony algorithm, glowworm algorithm, firefly algorithm, cuckoo search algorithm, bat algorithm, and hunting search algorithm [39]. Most of the swarm intelligence-based algorithms are simple and robust techniques that determine the optimum solution of optimization problems efficiently without requiring much of a mathematical struggling. PSO has attracted significant attention since its inception because of its simplicity and effectiveness in solving various optimization problems [40]. Therefore, we choose PSO for solving the stated problem.

C. The proposed method

PSO is a metaheuristic algorithm which is developed by Kennedy and Eberhart [41] to find good solutions in optimization problems which takes its inspiration from the cooperation and communication of a swarm of birds. Inspired by social models and swarming theory, it consists of a swarm that collectively explore the solution search space of an optimization problem and stores the best solutions found. In PSO, each individual in the swarm, called a particle, behaves like an agent of a highly decentralized and intelligent environment. Each particle of the swarm contributes to the environment by following very simple rules, thus cooperating and communicating with other particles of the swarm [42]. Particles are attracted at each iteration to the best solutions found by themselves and to the best solutions found by their neighbors, thereby encouraging the exploration of nearby solutions to potentially find better ones [43]. Analogous to other evolutionary algorithms, such as GA and ACO, PSO is a population-based stochastic optimization

algorithm. A swarm of particles attempts to search for superior solutions through learning, communication and interaction. The position of each particle refers to a solution. The state of particle i is described by its current position $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{iD}]$ and velocity $\mathbf{v}_i = [v_{i1}, v_{i2}, \dots, v_{iD}]$, where D is the number of variables encountered in the optimization problem. In the generic PSO with inertia weight [44], the position and velocity of particle i are updated during the evolutionary process [40]:

$$V_i^d = w \times V_i^d + c1 \times r_1^d \times (pbest_i^d - x_i^d) + c2 \times r_2^d \times (gbest^d - x_i^d) \quad (1)$$

$$x_i^d = x_i^d + V_i^d \quad (2)$$

where x_i^d is the d_{th} variable of the position of particle i ; V_i^d is the d_{th} variable of the velocity of particle i ; $pbest_i^d$ is the d_{th} variable of the personal historical best position found by particle i ; $gbest^d$ is the d_{th} variable of the global best position found by the whole swarm; c_1 and c_2 are the acceleration parameters commonly set to 2.0; r_1^d and r_2^d are two random numbers drawn from a uniform distribution over [0, 1] and w is the inertia weight used to set up the balance between the abilities of global and local search features of PSO [40, 44].

a. Solution Representation

As a first step in PSO, the candidate solutions must be encoded in a suitable form. Here, the solution is represented as a string of length n , where n is the number of the requested tasks. For example, fig. 2 represents a solution with 6 tasks and 4 servers. According to fig. 2, T_1 is assigned to S_1 (first server), T_2 and T_6 are assigned to S_3 and so on.

1	3	4	2	4	3
---	---	---	---	---	---

Fig. 2. An example of solution with 6 tasks and 4 server

b. Particle Fitness

Fitness or quality value shows how fit the solution is, i.e. how well it adapts to its environment. For a maximization problem, the fitness of a solution can be proportional to the value of the objective function [21]. We uses three terms including total execution time (Ttime), average of waiting time (Wtime) and numbers of missed tasks because of their deadline (Missed) for designing fitness function as follow:

$$FF_1 = 1/Ttime, \quad (3)$$

$$FF_2 = 1/Wtime, \quad (4)$$

$$FF = \frac{FF_1 + FF_2}{1 + Missed} \quad (5)$$

Where the aim of PSO is to maximize the value of FF.

c. Termination Condition

The algorithm is ran until no improvement in the fitness of the gbest has been observed for 10 generations or maximum iterations is reached.

¹ Execution Time

d. Pseudo Code

In this section the pseudo code of the proposed algorithm is described as follows:

- For each particle
- Initialize particle
- End For
- Do
- For each particle
- Calculate the fitness value of the particle fp
- /*updating particle's best fitness value so far)*/
- If fp is better than $pbest$
- Set current value as the new $pbest$
- End For
- /*updating population's best fitness value so far)*/
- Set $gbest$ to the best fitness value of all particles
- For each particle
- Calculate the particle velocity according to (1)
- Update particle position according to (2)
- End For
- End Do While maximum iterations OR no improvement in $gbest$ in 10 interactions are reached.

IV. SIMULATION RESULTS

In this section, we present the results to evaluate the performance of the proposed method using Matlab. We suppose the number of resources in the Cloud is 8 and the number of tasks are 40 where their execution and creation time are listed in table 1.

Table 1. Arrival time, service time, deadline and dependency of 40 tasks

Task #	Arrival Time	Service Time	Deadline	Dependency
1	14	4	8	----
2	15	2	4	----
3	19	5	10	1
4	14	8	16	1
5	9	9	18	----
6	9	2	4	2
7	6	1	2	5
8	18	8	16	----
9	20	3	6	1, 7
10	2	5	10	2, 5, 7
11	7	10	20	6
12	18	6	12	10
13	20	3	6	7, 13
14	9	10	20	8, 10
15	15	9	18	5, 3
16	9	9	18	10
17	3	7	14	----
18	4	3	6	11
19	5	5	10	16

20	7	4	8	15, 2
21	8	3	6	15, 11
22	2	1	2	----
23	13	7	14	8, 9, 13
24	5	8	16	5
25	7	8	16	1, 14
26	20	3	6	----
27	13	2	4	7, 11
28	0	10	20	22, 26
29	17	9	18	8
30	11	6	12	21
31	20	6	12	19, 29
32	6	6	12	16, 19, 27
33	11	4	8	16
34	10	7	14	23
35	9	9	18	17
36	18	4	8	16
37	4	1	2	----
38	13	4	8	31
39	17	7	14	12, 24
40	11	2	4	6, 37

To study the effectiveness of the proposed method, we compare it with First Come First Served (FCFS), Shortest Process Next (SPN) and Highest Response Ratio Next (HRRN). Fig 3 illustrates the simulation result based on the overall execution time that the PSO has better performance.

Fig 4 illustrates the simulation result based on the average service time + waiting time that the PSO has better performance.

Fig 5 illustrates the simulation result based on the average of missed tasks that the SPN has better performance.

Finally, Fig 6 illustrates the simulation result based on the resource utilization that the PSO has better performance.

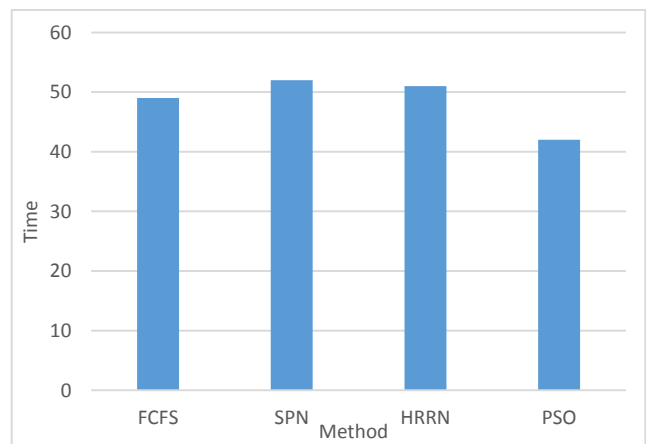


Fig. 3. The simulation result based on the overall execution time

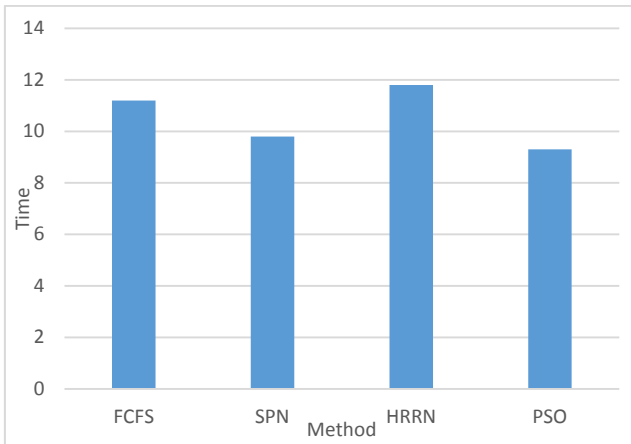


Fig. 4. The simulation result based on the average service time + waiting time

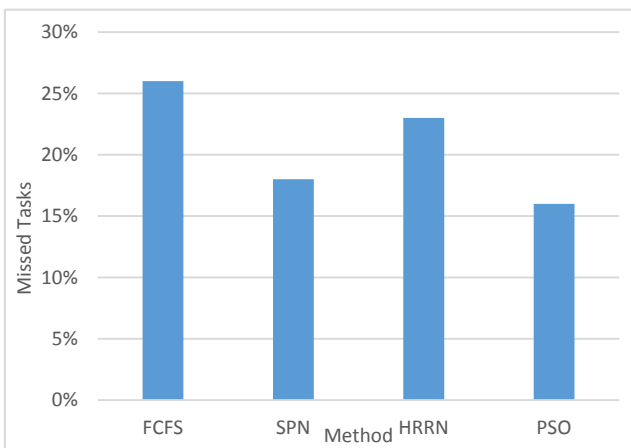


Fig. 5. The simulation result based on the average of missed tasks

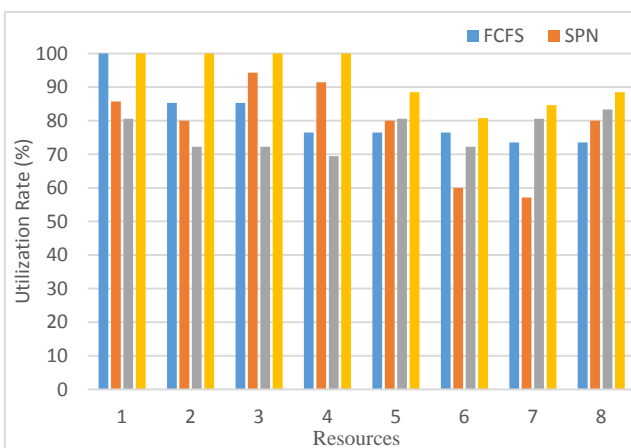


Fig. 6. The simulation result based on the resource utilization

V. CONCLUSION AND FUTURE WORKS

Cloud computing as a novel and entirely internet-based approach is an important concept in distributed systems. In Cloud computing, shared resources are provided to end-users in an on demand fashion that brings many advantages, including data ubiquity, flexibility of access, high availability of resources, and flexibility. In this type

of systems many challenges are existed that the task scheduling problem is one of them. In this paper, a new algorithm based on PSO is proposed to schedule the tasks in the Cloud. The results demonstrated that the proposed algorithm has a better operation in terms of task execution time, waiting time and missed tasks in comparison of FCFS, SPN and HRRN. In the future, we plan to modify the proposed method to support the heterogeneous resources.

REFERENCES

- [1] Oliveira, T., M. Thomas, and M. Espadanal, *Assessing the determinants of cloud computing adoption: An analysis of the manufacturing and services sectors*. Information & Management, 2014. **51**(5): p. 497-510.
- [2] Jafari Navimipour, N., et al., *Expert Cloud: a Cloud-based framework to share the knowledge and skills of human resources* Computer in Human Behaviour, 2014.
- [3] Wang, S.-S. and S.-C. Wang, *The consensus problem with dual failure nodes in a cloud computing environment*. Information Sciences, 2014. **279**(0): p. 213-228.
- [4] Manvi, S.S. and G. Krishna Shyam, *Resource management for Infrastructure as a Service (IaaS) in cloud computing: A survey*. Journal of Network and Computer Applications, 2014. **41**(0): p. 424-440.
- [5] Liu, W., et al., *Adaptive resource discovery in mobile cloud computing*. Computer Communications, 2014. **50**(0): p. 119-129.
- [6] Chong, H.-Y., J.S. Wong, and X. Wang, *An explanatory case study on cloud computing applications in the built environment*. Automation in Construction, 2014. **44**(0): p. 152-162.
- [7] Gurkok, C., *Chapter 4 - Securing Cloud Computing Systems*, in *Network and System Security (Second Edition)*, J.R. Vacca, Editor. 2014, Syngress: Boston. p. 83-126.
- [8] Nandhini, A. and B. Saravana Balaji, *Energy-Efficient PSO and Latency Based Group Discovery Algorithm in Cloud Scheduling*. International Journal of Information Technology and Computer Science, 2014 **6**(10).
- [9] Jou, M. and J. Wang, *Observations of achievement and motivation in using cloud computing driven CAD: Comparison of college students with high school and vocational high school backgrounds*. Computers in Human Behavior, 2013. **29**(2): p. 364-369.
- [10] Jafari Navimipour, N., et al., *Behavioral modelling and automated verification of a Cloud-based framework to share the knowledge and skills of human resources*. Computer in Industry, 2014.
- [11] Jafari Navimipour, N. and L. Mohammad Khanli. *The LGR method for task scheduling in computational grid*. in *Advanced Computer Theory and Engineering, 2008. ICACTE'08. International Conference on*. 2008. IEEE.
- [12] Habibzad Navin, A., et al., *Expert Grid: New Type of Grid to Manage the Human Resources and Study the Effectiveness of its Task Scheduler*. Arabian Journal for Science and Engineering, 2014.
- [13] Jafari Navimipour, N., et al., *Resource discovery mechanisms in grid systems: A survey*. Journal of Network and Computer Applications, 2014. **41**: p. 389-410.
- [14] Laili, Y., et al., *A Ranking Chaos Algorithm for dual scheduling of cloud service and computing resource in private cloud*. Computers in Industry, 2013. **64**(4): p. 448-463.

- [15] Tao, F., et al., *CLPS-GA: A case library and Pareto solution-based hybrid genetic algorithm for energy-aware cloud service scheduling*. Applied Soft Computing, 2014. **19**(0): p. 264-279.
- [16] Wu, Z., et al., *A market-oriented hierarchical scheduling strategy in cloud workflow systems*. The Journal of Supercomputing, 2013. **63**(1): p. 256-293.
- [17] Navimipour, N.J., et al., *Job scheduling in the Expert Cloud based on genetic algorithms*. Kybernetes, 2014. **43**(8): p. 12-12.
- [18] Montazeri, A., B. Akbari, and M. Ghanbari, *An incentive scheduling mechanism for peer-to-peer video streaming*. Peer-to-Peer Networking and Applications, 2012. **5**(3): p. 257-278.
- [19] Rius, J., F. Cores, and F. Solsona, *Cooperative scheduling mechanism for large-scale peer-to-peer computing systems*. Journal of Network and Computer Applications, 2013. **36**(6): p. 1620-1631.
- [20] Jafari Navimipour, N. and F. Sharifi Milani, *A comprehensive study of the resource discovery techniques in Peer-to-Peer networks*. Peer-to-Peer Networking and Applications, 2015: p. 1-19.
- [21] Jafari Navimipour, N. and F. Sharifi Milani, *Task Scheduling in the Cloud Computing based on the Cuckoo Search Algorithm*. International Journal of Modeling and Optimization, 2014.
- [22] Zhang, F., et al., *Multi-objective scheduling of many tasks in cloud platforms*. Future Generation Computer Systems, (0).
- [23] Gutierrez-Garcia, J.O. and K.M. Sim, *A family of heuristics for agent-based elastic Cloud bag-of-tasks concurrent scheduling*. Future Generation Computer Systems, 2013. **29**(7): p. 1682-1699.
- [24] Garg, S.K., et al., *Environment-conscious scheduling of HPC applications on distributed Cloud-oriented data centers*. Journal of Parallel and Distributed Computing, 2011. **71**(6): p. 732-749.
- [25] Mezmaz, M., et al., *A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems*. Journal of Parallel and Distributed Computing, 2011. **71**(11): p. 1497-1508.
- [26] Li, J., et al., *Online optimization for scheduling preemptable tasks on IaaS cloud systems*. Journal of Parallel and Distributed Computing, 2012. **72**(5): p. 666-677.
- [27] Abrishami, S. and M. Naghibzadeh, *Deadline-constrained workflow scheduling in software as a service Cloud*. Scientia Iranica, 2012. **19**(3): p. 680-689.
- [28] Lu, K., et al., *Resources collaborative scheduling model based on trust mechanism in cloud*. in *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*. 2012. IEEE.
- [29] Wang, W., et al., *Cloud-DLS: Dynamic trusted scheduling for Cloud computing*. Expert Systems with Applications, 2012. **39**(3): p. 2321-2329.
- [30] Dogan, A. and F. Ozguner, *Matching and scheduling algorithms for minimizing execution time and failure probability of applications in heterogeneous computing*. Parallel and Distributed Systems, IEEE Transactions on, 2002. **13**(3): p. 308-323.
- [31] Van den Bossche, R., K. Vanmechelen, and J. Broeckhove, *Online cost-efficient scheduling of deadline-constrained workloads on hybrid clouds*. Future Generation Computer Systems, 2013. **29**(4): p. 973-985.
- [32] Wang, Y., et al., *Resource scheduling of cloud with QoS constraints*, in *Advances in Neural Networks-ISNN 2013*. 2013, Springer. p. 351-358.
- [33] Frîncu, M.E., *Scheduling highly available applications on cloud environments*. Future Generation Computer Systems, 2014. **32**: p. 138-153.
- [34] Liu, Z., et al., *Resource preprocessing and optimal task scheduling in cloud computing environments*. Concurrency and Computation: Practice and Experience, 2014.
- [35] Jung, D., et al., *A Workflow Scheduling Technique for Task Distribution in Spot Instance-Based Cloud*, in *Ubiquitous Information Technologies and Applications*, Y.-S. Jeong, et al., Editors. 2014, Springer Berlin Heidelberg. p. 409-416.
- [36] Thulasiraman, K. and M. Swamy, *Acyclic Directed Graphs*. Graphs: Theory and Algorithms, 1992.
- [37] Baiocchi, M., G. Busanello, and B. Vantaggi, *Acyclic directed graphs representing independence models*. International Journal of Approximate Reasoning, 2011. **52**(1): p. 2-18.
- [38] Rebollo-Ruiz, I. and M. Graña, *An empirical evaluation of Gravitational Swarm Intelligence for graph coloring algorithm*. Neurocomputing, 2014. **132**(0): p. 79-84.
- [39] Saka, M.P., E. Doğan, and I. Aydogdu, *2 - Analysis of Swarm Intelligence-Based Algorithms for Constrained Optimization*, in *Swarm Intelligence and Bio-Inspired Computation*, X.-S. Yang, et al., Editors. 2013, Elsevier: Oxford. p. 25-48.
- [40] Wu, G., et al., *Superior solution guided particle swarm optimization combined with local search techniques*. Expert Systems with Applications, 2014. **41**(16): p. 7536-7548.
- [41] Kennedy, J. and R. Eberhart. *Particle swarm optimization*. in *Proceedings of IEEE international conference on neural networks*. 1995. Perth, Australia.
- [42] Alam, S., et al., *Research on particle swarm optimization based clustering: A systematic review of literature and techniques*. Swarm and Evolutionary Computation, 2014. **17**(0): p. 1-13.
- [43] Rada-Vilela, J., M. Johnston, and M. Zhang, *Population statistics for particle swarm optimization: Resampling methods in noisy optimization problems*. Swarm and Evolutionary Computation, 2014. **17**(0): p. 37-59.
- [44] Shi, Y. and R. Eberhart. *A modified particle swarm optimizer*. in *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*. 1998. IEEE.

Authors' Profiles

Farnaz Sharifi Milani is a M.Sc. student in Department of Computer Engineering, East Azarbaijan Science and Research Branch, Islamic Azad University, Tabriz, Iran since 2013. She received her B.S. degree in Information Technology Engineering from Tabriz Branch, Islamic Azad University, Tabriz, Iran in 2010. Her current research interests are focused on wireless sensor networks, Peer to Peer networks, Cloud computing and Network-on-Chip.

Ahmad Habibzad Navin was born in 1971. He received his H.N.D in electronic in 1997 and B.Sc. degree in applied mathematics from Tabriz University, Tabriz, Iran, in 1999. He received his M.Sc. degree in computer architecture from Science and Research Branch of Islamic Azad University, Tehran, Iran, in 2003 and his Ph.D. in computer hardware from Science and Research Branch, Islamic Azad University, Tehran, Iran, in 2007. His research interest includes computer architecture, data-oriented approach, robotic, soft computing and probability and statistic.