Kybernetes

Job scheduling in the Expert Cloud based on genetic algorithms

Nima Jafari Navimipour Amir Masoud Rahmani Ahmad Habibizad Navin Mehdi Hosseinzadeh

Job scheduling in the Expert Cloud based on genetic algorithms

Nima Jafari Navimipour, Amir Masoud Rahmani and
Ahmad Habibizad Navin
*Department of Computer, Science and Research Branch,
Islamic Azad University, Tehran, Iran, and*
Mehdi Hosseinzadeh
Science and Research Branch, Islamic Azad University, Tehran, Iran

Abstract

Purpose – Expert Cloud as a new class of Cloud computing systems enables its users to request the skill, knowledge and expertise of people by employing internet infrastructures and Cloud computing concepts without any information of their location. Job scheduling is one of the most important issue in Expert Cloud and impacts on its efficiency and customer satisfaction. The purpose of this paper is to propose an applicable method based on genetic algorithm for job scheduling in Expert Cloud.

Design/methodology/approach – Because of the nature of the scheduling issue as a NP-Hard problem and the success of genetic algorithm in optimization and NP-Hard problems, the authors used a genetic algorithm to schedule the jobs on human resources in Expert Cloud. In this method, chromosome or candidate solutions are represented by a vector; fitness function is calculated based on response time; one point crossover and swap mutation are also used.

Findings – The results indicate that the proposed method can schedule the received jobs in appropriate time with high accuracy in comparison to common methods (First Come First Served, Shortest Process Next and Highest Response Ratio Next). Also the proposed method has better performance in term of total execution time, service + wait time, failure rate and Human Resource utilization rate in comparison to common methods.

Originality/value – In this paper the job scheduling issue in Expert Cloud is pointed out and the approach to resolve the problem is applied into a practical example.

Keywords Social networks, Networking, Information technology, Architecture

Paper type Research paper

1. Introduction

Human society, information space and the physical world can be fully connected and integrated with the help of various technologies such as Cloud computing (Li *et al.*, 2013). Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of resources that can be rapidly provisioned and released with minimal management effort or service provider interaction (Mell and Grance, 2009). The Expert Cloud (Jafari Navimipour *et al.*, 2014a) as a new class of Cloud systems provides the knowledge and skills of Human Resources (HRs) as a service in Cloud systems. In the Expert Cloud each HR can enjoy all the features that it provides for instance, discovering, ranking, messages sending and knowledge sharing over the internet. HR virtualization exists in many cases. It happens where HR communicates through the Expert Cloud to far-away colleague to get her/his aid. It happens in a virtual society or organizations whose members come together temporarily to meet a specific need. Or it may happen if an HR uses the knowledge and skills of other HRs to perform some tasks. The expertise, knowledge and skills of HR provide a new class of service in the Cloud systems named Expert as a Service (EaaS).

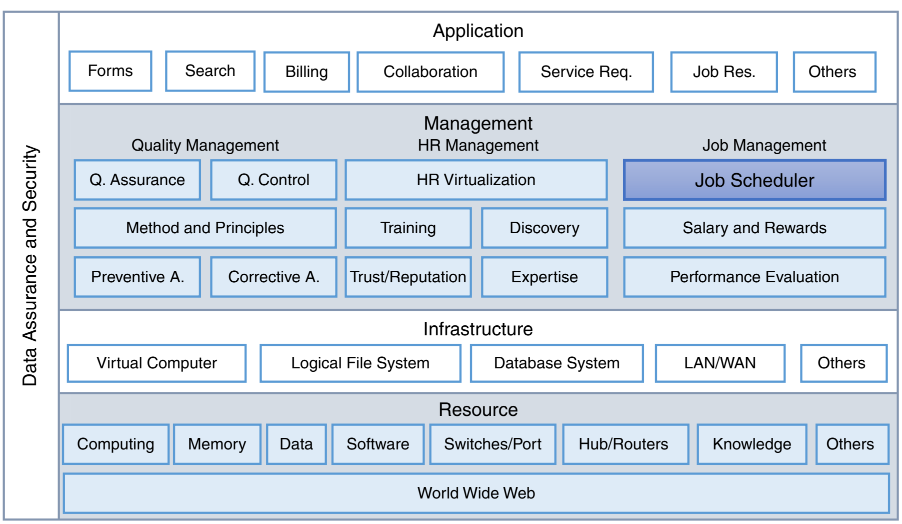
Kybernetes
Vol. 43 No. 8, 2014
pp. 1262-1275

0368-492X
DOI 10.1108/K-02-2013-0018

EaaS provides customers with transparent access to experts, knowledge and skills of HRs remotely over the internet (Jafari Navimipour *et al.*, 2014a).

To virtualize the HR, provide EaaS, and share the expertise and skills of HR, the Expert Cloud utilizes a layered structure that corresponds to the Cloud architecture (Jafari Navimipour *et al.*, 2014a). The architecture of Expert Cloud is depicted as four layers: Application, Management, Infrastructure and Resource layers (see Figure 1.). The Application Layer not only provides the connection between the Expert Cloud, customer and manager by graphical user interface, but also provides some applications to realize EaaS for customers. The Management Layer manages the HR, jobs and quality related issues. The Infrastructure Layer develops the Resource Layer to provide essential tools for Management Layer and handles the organization rules and policies. The Resource Layer provides some resources such as computing devices and storage resources. Also, it transfers information and data using World Wide Web. Data Assurance and Security consists of the policies adopted by the Expert Cloud administrator to prevent and monitor misuse, modification, or denial of service in all four layers. Some of the duties of Data Assurance and Security are authorization, authentication, encryption, decryption, coding and deterrent/detective/preventive/corrective controls (Jafari Navimipour *et al.*, 2014a; Jafari Navimipour *et al.*, 2015).

Job scheduler is a vital part of any distributed system like Grid (Jafari Navimipour and Mohammad Khanli, 2008; Habibizad Navin *et al.*, 2014; Yang *et al.*, 2010; Jafari Navimipour *et al.*, 2014b), Cloud (Laili *et al.*, 2013; Mezmaz *et al.*, 2011; Pandey *et al.*, 2010; Tao *et al.*, 2014; Wang *et al.*, 2012; Wu *et al.*, 2013) and P2P networks (Montazeri *et al.*, 2012; Erdil, 2012; Luo *et al.*, 2012; Rius *et al.*, 2013; Jafari Navimipour and Sharifi Milani, 2015) which assigns jobs to suitable resources for execution. The goal of job scheduler is to minimize the overall execution time of a collection of jobs. In Expert Cloud, as a class of distributed systems, job scheduler (highlighted in Figure 1) determines which HR must do which job at a given period of time. The aim of job scheduler is to shorten the response time and enhance the HR utilization.



Source: Jafari Navimipour *et al.* (2014a)

Figure 1.
The layered architecture of the Expert Cloud

Scheduling issue in the Cloud systems has been well studied; but in the Expert Cloud, it is much more complicated than Cloud systems, because in the Expert Cloud the resources are HRs. These HRs may lead to some difficulties such as: the uncertainty of HR output, trust/expertise/reputation degree of HR, emotional related issues, cost of service and, etc. on the other hand, genetic algorithm is a random search algorithm based on the concept of natural genetics, which is a very popular method for searching the optimum solution of the complex problems (Ran *et al.*, 2010). Therefore, in this paper, we propose a job scheduler mechanism to overcome the stated weaknesses by genetic algorithm.

The rest of the paper is organized as follows: first, we give an overview of the most important scheduling techniques in Cloud systems. In Section 3, we propose a mechanism to schedule the jobs on HRs in the Expert Cloud as well as the related algorithms and examples. In Section 4 the obtained results are presented. Lastly, conclusion and future works are provided in Section 5.

2. Related works

Scheduling in traditional parallel systems has been well studied which are constructed in a stable environment (Luo *et al.*, 2012). However, the scheduling in the Cloud systems is more complicated than that in traditional parallel systems, because Cloud systems can have heterogeneity, dynamicity, intermittent presence and large communication overhead characteriztics. Also, up to now, there is not any specialized paper which studied the scheduling issue in the Expert Cloud. Therefore, in this section, we briefly review and analyze the state of the art mechanisms and approaches in job and task scheduling in Cloud systems.

Pandey *et al.* (2010) have proposed a Particle Swarm Optimization (PSO) based heuristic to schedule applications to Cloud resources that takes into account both computation cost and data transmission cost. They also experimented a workflow application by varying its computation and communication costs. They compared the cost savings when using PSO and Best Resource Selection (BRS) algorithm where the resource is selected based on its performance. The obtained results demonstrated that PSO can achieve as much as three times cost savings as compared to BRS, and good distribution of the workload onto resources.

Mezmaz *et al.* (2011) 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 that takes into account, not only makespan, but also energy consumption. They focussed on the island parallel model and the multi-start parallel model. The results demonstrated that the algorithm outperformed previous scheduling methods in terms of completion time and energy consumption.

Abrishami and Naghibzadeh (2012) 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. The proposed algorithm recursively schedules the PCP ends at previously scheduled tasks. The results demonstrated that the computation time of the proposed algorithm is low for the cost decreasing and the fair policies, but is much longer for the optimized policy.

Wang *et al.* (2012) have presented a Bayesian method based cognitive trust model, and a trust dynamic level scheduling (DLS) algorithm named Cloud-DLS by integrating the existing DLS algorithm (Dogan and Ozguner, 2002). Moreover, a benchmark was structured to span a range of Cloud computing characteristics for evaluation of the proposed method. Theoretical analysis and simulations showed that the Cloud-DLS algorithm can efficiently meet the requirement of Cloud computing workloads in trust, sacrificing fewer time costs and assuring the execution of tasks in a secure way (Wang *et al.*, 2012).

Laili *et al.* (2013) have combined the Service Composition Optimal Selection and Optimal Allocation of Computing Resources into one-time decision in one console, named Dual Scheduling of Cloud Services and Computing Resources (DS-CSCR). Also the mutual relations between the upper layer Cloud services and the underlying infrastructures in the private Cloud are analyzed. Furthermore, for addressing large-scale DS-CSCR problem, a new Ranking Chaos Optimization (RCO) is proposed. With the consideration of large-scale irregular solution spaces, new adaptive chaos operator was designed to traverse wider spaces within a short time. Moreover, dynamic heuristic and ranking selection are introduced to control the chaos evolution in the proposed algorithm. Theoretical analysis and simulations showed that the new DS-CSCR outperformed the traditional two-level decision making with the improvements in both Cloud service composition and computing resource allocation. In addition, RCO can remarkably give many prominent solutions with low time-consuming and higher stability than a few typical intelligent algorithms for solving DS-CSCR in a private Cloud. With the new DS-CSCR and RCO, Cloud services and computing infrastructures can then be quickly combined and shared with high efficient decision (Laili *et al.*, 2013).

Tao *et al.* (2014) have proposed Case Library and Pareto Solution based hybrid genetic algorithm CLPS-GA for Optimal Scheduling of Computing Resources. In this method, the Cloud computing environment is considered to be highly heterogeneous with processors of uncertain loading information. Along with makespan, the energy consumption was considered as one of the optimization objectives from both economic and ecological perspectives. To provide more attentive services, the model seeks to find Pareto solutions for this bi-objective optimization problem. The major components of CLPS-GA included a Multi-Parent Crossover Operator, a two-stage algorithm structure, and a case library. Experimental results demonstrated the effectiveness of CLPS-GA in terms of convergence, stability and solution diversity (Tao *et al.*, 2014).

Liu *et al.* (2014) have proposed a fuzzy clustering method to effectively pre-process 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. Earliest finish time duplication attempts to insert suitable immediate parent nodes of the current selected node in order to reduce its waiting time on the processor. The case study and experimental results revealed that the proposed algorithm is better than the popular heterogeneous earliest finish time algorithms.

Jung *et al.* (2014) have presented the workflow scheduling scheme that reduces the task waiting time when an instance occurs the out-of-bid situation. Also, their scheme executed user's job within selected instances and expands the suggested user budget. The simulation results demonstrated that, compared to various instance types, the proposed scheme achieves performance improvements in terms of an average execution time of 66.86 percent over shortest execution time in each task time interval. Furthermore, they showed that the cost of the proposed scheme is higher than an instance with low performance and is lower than an instance with high performance.

As discussed and mentioned in this section and to the best of our knowledge, all the related works and researches on this scope have not considered the job scheduling on HR. In other words, there is not any detailed research that investigated the job scheduling issue in the Expert Cloud. Therefore, in the next section, we propose an approach for job scheduling in the Expert Cloud where the resources are HR.

3. Job scheduler in expert cloud

To schedule the jobs on HR, first the model of the system must be defined and then the mechanism of scheduling the requested HR based on her/his basic information and trust factor must be proposed. These steps and the primary definitions are described in this section.

3.1 Graph-based model for the Expert Cloud

The Expert Cloud can be modeled as an undirected and weighted graph G = (V, E) where V is the set of vertices and E is the set of edges, formed by pairs of vertices. Each node i in $V = \{1, 2, [\ldots], mm$ represents an HR which contributing his/her resources in the Expert Cloud and $E(i, j)$ (the edge between V_i and V_j) indicates that the HR_i and HR_j are colleagues or they interacted with each other before. Also each node labeled by the information and properties of related HR. As an example, this label can be a four-attributes which are shown the identifier number, skill type, expertise and reputation value of HR. Figure 2 shows a graph model of the Expert Cloud with seven HR that edges and nodes are labeled as discussed (Jafari Navimipour *et al.*, 2014a). For example, V_2 is a Civil Engineer and its expertise and reputation value is 0.458 and 0.375, respectively.

3.2 GA-based job scheduler

Genetic algorithms are among the most popular evolutionary algorithms in terms of the diversity of their applications (Yang, 2014). A vast majority of well-known optimization problems like (Arora *et al.*, 2012; El Dein, 2014; Selvaraj and Anand, 2012; Carro-Calvo *et al.*, 2010; Lim *et al.*, 2012; González-Torres *et al.*, 2013; Qin *et al.*, 2013; Peng and Song, 2010) have been solved using genetic algorithms. Therefore, in this section we present a genetic algorithm for the job scheduling problem in the Expert Cloud. We suppose that the set $BJ = (J_1, J_2, J_3, \ldots, J_n)$ is a batch of jobs arrived in a period. A service request for j_i $(0 \leqslant i \leqslant n)$ is represented as $(T_i, H_i, E_i, Tr_i, R_i, D_i)$,

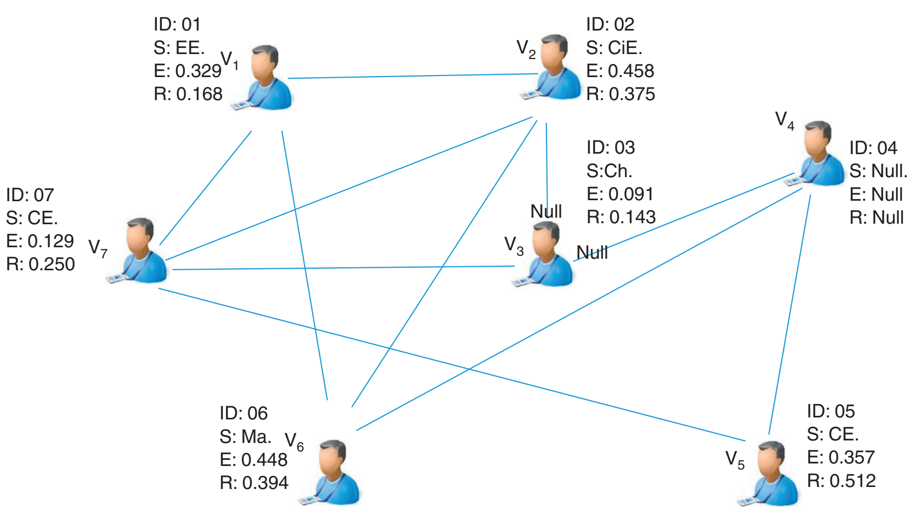


Figure 2.
A graph model of a part of the expert cloud with seven HR

where T_i is the time that the request is received, H_i is a type of required HR, E_i, Tr_i and R_i represent the minimum value of expertise, trust and reputation (in the range of [0,1]) that j_i want, and D_i is the deadline for performing j_i after which the job will be considered as a failed job. Also h indicates the number of HRs.

3.2.1 Chromosome definition. The genetic algorithm is started with a population of strings (represented by chromosomes), which encodes candidate solutions called individuals (Sivanandam and Deepa, 2008). We represent each chromosome by a vector with the length n. For example, if there are 6 jobs and 4 HRs, a sample chromosome is illustrated in Figure 3.

Figure 3 indicates that HR_4 is assigned to J_1 and J_6, HR_1 is assigned to J_2 and J_5, HR_2 is assigned to J_3 and HR_3 is assigned to J_4. Also to validate the created chromosome, the algorithm checks the requirements of the job such as *H, E, T, R* and *D*; and assigns those HRs to a job that satisfy its requirements. Then a population of solutions is created and genetic operators such as mutation and crossover are applied. It should be stated that the population size is 50 chromosomes.

3.2.2 Fitness function (FF). The FF is a way of determining how far or close the chromosome is to achieving the set objectives (Janc *et al.*, 2013). Equation (1) calculates the response time as a FF for *BJ* that the purpose of GA is to maximize its value:

$$FF = \frac{1}{\sum_{i=1}^{n} \sum_{j=1}^{h} R_{ij} A_{ij}} \tag{1}$$

Where n is the number of jobs, h indicates number of HRs, R_{ij} indicates the time that the HR_j responded to the job_i request, and A_{ij} shows the allocation status between HR_j and job_i and obtained from Equation (2):

$$A_{ij} = \begin{cases} 1, & \text{if } HR_j \text{ is assigned to } job_i \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

3.2.3 Initial population. The initial population settings the speed and the convergence of the genetic algorithm (Abdoun *et al.*, 2012). We use a random method to initial the population with 50 chromosomes in which satisfy the criteria of the problem. Where the initial population is generated randomly and, after that, in each iteration the best individuals are selected and the worst ones are replaced with new ones generated randomly (Diaz-Gomez and Hougen, 2007).

3.2.4 Selection operator. Selection is an important part of genetic algorithms since it affects significantly their convergence. To mate and reproduce new population, we must select the parents from the population by an approach. The basic strategy follows the rule: The better fitted an chromosome, the larger the probability of its survival and mating (Lipowski and Lipowska, 2012). The most straightforward implementation of this rule is the so-called roulette-wheel selection (Goldberg, 1989). Roulette wheel selection is one common technique used in GA implementations to select the chromosome. The first step in the selection process is to run all of the chromosomes

4	1	2	3	1	4

Figure 3. A sample chromosome for four jobs and six HRs

from the initial population through the schedule builder. After all of the chromosomes have been scheduled and scored, the sum of all of the individual fitness's is calculated. This will represent the total fitness of the population (Sun *et al.*, 2010). The fitness of an individual determines the probability of its selection:

Definition 1. Let a population contains N individuals where their fitness $fi > 0$ ($i = 1, 2, \ldots, N$) are calculated. The selection probability of the i-th individual is (Lipowski and Lipowska, 2012):

$$pi = \frac{f_i}{\sum_{i=1}^{N} f_i} \qquad (i = 1, 2, \ldots, N) \tag{3}$$

3.2.5 Crossover operator. The crossover operator combines more than one chromosome to generate the chromosomes of the new generation. The new chromosome inherits some features from first parent and the other features from the other parent (Shukla *et al.*, 2010). We define a parameter P_c of a genetic system as the probability of crossover. This probability gives us the expected number P_c. In order to determine the parents for crossover operation, the following process is done repeatedly from $i = 1$ to *pop_size* (the number of chromosomes in the population): generating a random number r from the interval [0, 1], the chromosome C_i is selected as a parent if $r < P_c$) Liu (2009) (We have used one point crossover where a crossover point is randomly chosen for two randomly selected chromosomes as parents by the probability P_c ($P_c = 0.9$). The randomly point is selected between two genes and divides each chromosome into left and right sections. Crossover then swaps the left (or the right) section of the two chromosomes (see Figure 4).

After performing the crossover, in any offspring if any genes are missing, the replicated gene in a chromosome is removed and replaced by missed gene. Also, if the offspring chromosome did not satisfy the determined parameters, it is removed and the process is repeated.

3.2.6 Mutation operator. In GA, mutation operator is used to maintain the diversity of the population by changing chromosome with a small probability, $P_m = 0.1$, which is known as the probability of mutation. This operator chooses two genes randomly and changes the allele value of them (see Figure 5). In a similar manner to the process of selecting parents for crossover operation, the following steps from $i = 1$ to *pop_size* is done: generating a random number r from the interval [0, 1], the chromosome C_i is selected as a parent for mutation if $r < P_m$ (Liu, 2009).

3.2.7 Termination condition and pseudo code. The algorithm is terminated if no further improvement in the fitness value of the best chromosome in the population is not occurring for five iterations or maximum number of generations is reached. The Pseudo code of the algorithm is described by these steps:

(1) initial pop_size chromosomes randomly;

(2) calculate the fitness of each chromosome;

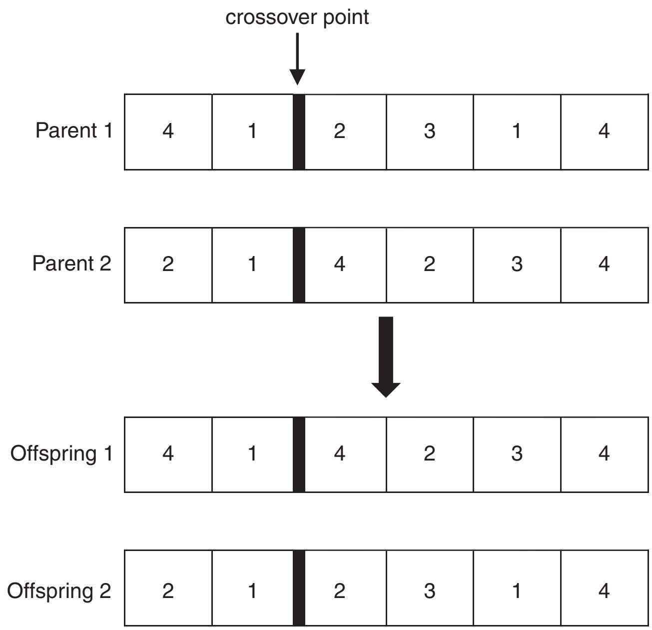
(3) perform crossover;

(4) perform mutation;

(5) evaluate the fitness of the offspring;

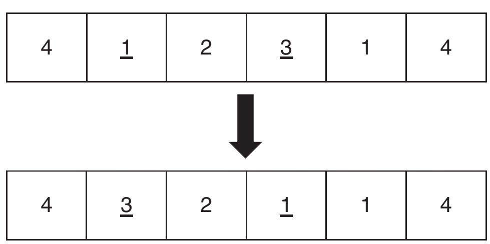
(6) select the survive individuals; and

(7) proceed from three if the termination criteria have not been reached.



Note: The elements of offsprings are ingerited from the parents

Figure 4. One point crossover. The elements of offspring are inherited from the parents



Note: Two gens exchanged thier values

Figure 5. Mutation operator. Two gens exchanged their values

Table I illustrates a summary of the presented steps and the important parameters that are used to experiment the results.

4. Experimental results

The proposed job scheduler method in the Expert Cloud is implemented and evaluated on the Expert Cloud networks using PHP[1]. Section 4.1 indicates two scenarios and the related data sets. Section 4.2 shows the obtained results in term of HR utilization rate and fitness value.

4.1 Scenarios

In this subsection, we develop a data set which links to the content of a table. The table shows the features of incoming jobs where every column of the table represents a particular demand (arrival time, service time, deadline), and each row corresponds to

	GA parameters
Size of population	50
Maximum number of generation	1,000
Selection mechanism	Roulette wheel
Fitness function	$\frac{1}{\sum_{i=1}^{n} \sum_{j=1}^{h} R_{ij} A_{ij}}$
Crossover	One point $(P_c = 0.9)$
Mutation	Swap $(P_m = 0.1)$
Termination conditions	No improvement in the fitness value of the best chromosome for five iterations

Table I.
A summary of the important parameters of presented GA

an agreed job. To better analyze the efficiency of the proposed method, we provide two scenarios. The former considers five HRs and 30 jobs which their properties (arrival time, service time and deadline from arrival time) are provided in Table II.

The latter considers ten HRs and 20 jobs with their properties (arrival time, service time and deadline from arrival time) listed in Table III.

Job#	Arrival time	Service time	Deadline
1	4	3	6
2	5	6	12
3	5	10	20
4	18	10	20
5	7	10	20
6	15	3	6
7	6	4	8
8	9	2	4
9	9	5	10
10	2	9	18
11	13	4	8
12	6	2	4
13	5	5	10
14	3	3	6
15	1	8	16
16	4	8	16
17	10	8	16
18	12	10	20
19	7	3	6
20	18	7	14
21	20	10	20
22	3	10	20
23	4	7	14
24	1	6	12
25	8	10	20
26	13	8	16
27	8	7	14
28	10	3	6
29	11	4	8
30	16	4	8

Table II.
First scenario. 30 jobs and their arrival time, service time and deadline

Job#	Arrival time	Service time	Deadline
1	1	7	14
2	18	1	2
3	17	9	18
4	2	2	4
5	20	3	6
6	0	1	2
7	7	9	18
8	3	4	8
9	7	2	4
10	12	5	10
11	16	10	20
12	19	9	18
13	20	10	20
14	6	10	20
15	15	10	20
16	5	5	10
17	10	6	12
18	18	5	10
19	1	7	14
20	17	8	16

Table III.
Second scenario. 20 jobs and their arrival time, service time and deadline

4.2 Results

In this subsection, we evaluate the performance of the proposed method in comparison to the other classical approaches. First, we consider the execution time in these scenarios. Figure 6 shows the total execution time (TET) of all jobs in First Come First Served (FCFS) (Stallings, 2008), Shortest Process Next (SPN) (Stallings, 2008), Highest Response Ratio Next (HRRN) (Stallings, 2008) and GA.

As a next experiment we consider the average of Service + Wait time (S + W) of all jobs. Figure 7 shows the S + W values in FCFS, SPN, HRRN and GA for the first and second scenarios.

The failure rate of incoming jobs is another important factor for evaluation of scheduler algorithms. Figure 8 shows the failure rate in FCFS, SPN, HRRN and GA for the first and second scenarios.

Figure 6.
The total execution time (TET) of all jobs in FCFS, SPN, HRRN and GA methods

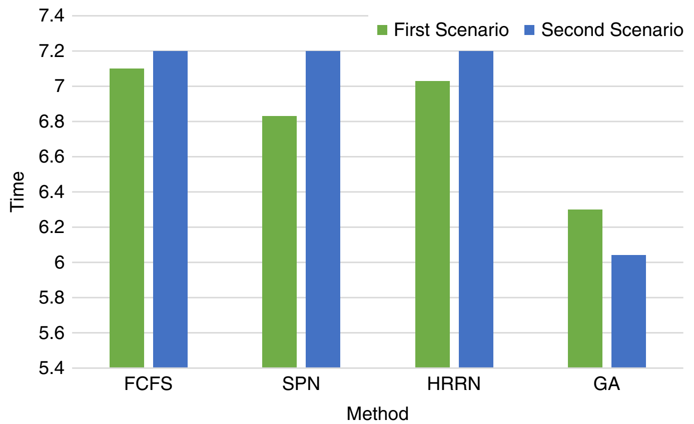


Figure 7.
The average of Service + Wait time (S + W) of all jobs in FCFS, SPN, HRRN and GA methods

Figure 8.
The failure rate values in FCFS, SPN, HRRN and GA methods

The HR utilization rate is another important factor to determine the effectiveness of the proposed job scheduler. Figure 9 shows the average HR utilization rate in FCFS, SPN, HRRN and GA for the first and second scenarios.

The obtained results in this subsection demonstrated that the proposed method has better performance in term of TET, service + wait time, failure and HR utilization rate than common methods (FCFS, SPN and HRRN).

5. Conclusion and future works

In the Expert Cloud, efficient job scheduling mechanism can meet users' requirements, and improve the resource utilization, thereby enhancing the overall performance of the Expert Cloud. In this paper, we introduced a genetic based algorithm to schedule the jobs on HRs in the Expert Cloud. The obtained results showed that the proposed method can schedule the received jobs in appropriate time with high accuracy. In comparison to common methods (FCFS, SPN and HRRN), the proposed method has better performance in term of TET, service + wait time, failure and HR utilization rate. But in term of failure rate, SPN acts slightly better.

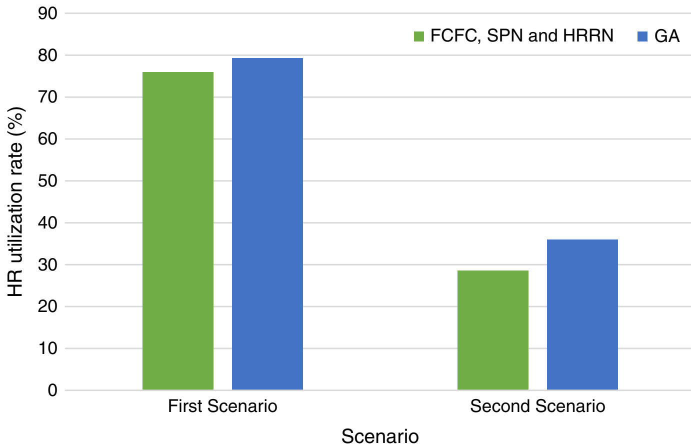


Figure 9.
The HR utilization rate in FCFS, SPN, HRRN and GA methods

In this paper, we used a selection method that selects one of the N chromosomes using search algorithms of $O(N)$ complexity. In the future, we can extend this process by improving it using stochastic acceptance (Lipowski and Lipowska, 2012) which typically has $O(1)$ complexity. Also, we will look into extending the approach to provide some QoS. Moreover, we would like to apply other methods such as fuzzy logics for considering issue and study its effectiveness. Lastly, another direction for further study is to investigate the effects of the size of the Expert Cloud on the performance of the proposed job scheduler mechanism.

Note

1. www.php.net

References

Abdoun, O., Tajani, C. and Abouchabaka, J. (2012), "Analyzing the performance of mutation operators to solve the traveling salesman problem", *International Journal of Emerging Sciences*, Vol. 2 No. 1, pp. 61-77.

Abrishami, S. and Naghibzadeh, M. (2012), "Deadline-constrained workflow scheduling in software as a service cloud", *Scientia Iranica*, Vol. 19 No. 3, pp. 680-689.

Arora, S., Hazan, E. and Kale, S. (2012), "The multiplicative weights update method: a meta-algorithm and applications", *Theory of Computing*, Vol. 8, pp. 121-164.

Carro-Calvo, L., Salcedo-Sanz, S., Portilla-Figueras, J.A. and Ortiz-García, E.G. (2010), "A genetic algorithm with switch-device encoding for optimal partition of switched industrial Ethernet networks", *Journal of Network and Computer Applications*, Vol. 33 No. 4, pp. 375-382.

Diaz-Gomez, P.A. and Hougen, D.F. (2007), "Initial population for genetic algorithms: a metric approach", *GEM*, pp. 43-49.

Dogan, A. and Ozguner, F. (2002), "Matching and scheduling algorithms for minimizing execution time and failure probability of applications in heterogeneous computing", *Parallel and Distributed Systems, IEEE Transactions On*, Vol. 13 No. 3, pp. 308-323.

El Dein, A. (2014), "Optimal arrangement of Egyptian overhead transmission lines' conductors using genetic algorithm", *Arabian Journal for Science and Engineering*, Vol. 39 No. 2, pp. 1049-1059.

Erdil, D.C. (2012), "Simulating peer-to-peer cloud resource scheduling", *Peer-to-Peer Networking and Applications*, Vol. 5 No. 3, pp. 219-230.

Goldberg, D.E. (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-wesley Reading, Menlo Park, Boston, MA.

González-Torres, A., García-Peñalvo, F.J. and Therón, R. (2013), "Human-computer interaction in evolutionary visual software analytics", *Computers in Human Behavior*, Vol. 29 No. 2, pp. 486-495.

Habibizad Navin, A., Jafari Navimipour, N., Rahmani, A.M. and Hosseinzadeh, M. (2014), "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*, Vol. 39 No. 8, pp. 6175-6188.

Jafari Navimipour, N., Habibizad Navin, A., Rahmani, A.M. and Hosseinzadeh, M. (2014a), "Expert cloud: a cloud-based framework to share the knowledge and skills of human resources", *Computer in Human Behaviour* (accepted paper).

Jafari Navimipour, N. and Mohammad Khanli, L. (2008), "The LGR method for task scheduling in computational grid", *Advanced Computer Theory and Engineering, 2008*, *ICACTE 08 International Conference on, IEEE*, Phuket, December 20-22, pp. 1062-1066.

Jafari Navimipour, N. and Sharifi Milani, F. (2015), "A comprehensive study of the resource discovery techniques in peer-to-peer networks", *Peer-to-Peer Networking and Applications*, pp. 1-19.

Jafari Navimipour, N., Habibizad Navin, A., Rahmani, A.M. and Hosseinzadeh, M. (2015), "Behavioural modelling and automated verification of a Cloud-based framework to share the knowledge and skills of human resources", *Computers in Industry* (accepted paper).

Jafari Navimipour, N., Masoud Rahmani, A., Habibizad Navin, A. and Hosseinzadeh, M. (2014b), "Resource discovery mechanisms in grid systems: A survey", *Journal of Network and Computer Applications*, Vol. 41, pp. 389-410.

Janc, K., Tarasiuk, J., Bonnet, A.S. and Lipinski, P. (2013), "Genetic algorithms as a useful tool for trabecular and cortical bone segmentation", *Computer Methods and Programs in Biomedicine*, Vol. 111 No. 1, pp. 72-83.

Jung, D., Lim, J., Yu, H., Gil, J. and Lee, E. (2014), "A workflow scheduling technique for task distribution in spot instance-based cloud", in Jeong, Y.-S., Park, Y.-H., Hsu, C.-H. and Park, J.J. (Eds), *Ubiquitous Information Technologies and Applications*, Springer, Berlin Heidelberg.

Laili, Y., Tao, F., Zhang, L., Cheng, Y., Luo, Y. and Sarker, B.R. (2013), "A ranking chaos algorithm for dual scheduling of cloud service and computing resource in private cloud", *Computers in Industry*, Vol. 64 No. 4, pp. 448-463.

Li, W., Zhong, Y., Wang, X. and Cao, Y. (2013), "Resource virtualization and service selection in cloud logistics", *Journal of Network and Computer Applications*, Vol. 36 No. 6, pp. 1696-1704.

Lim, Y.C., Tan, T.S., Shaikh Salleh, S.H. and Ling, D.K. (2012), "Application of genetic algorithm in unit selection for malay speech synthesis system", *Expert Systems with Applications*, Vol. 39 No. 5, pp. 5376-5383.

Lipowski, A. and Lipowska, D. (2012), "Roulette-wheel selection via stochastic acceptance", *Physica A: Statistical Mechanics and its Applications*, Vol. 391 No. 6, pp. 2193-2196.

Liu, B. (2009), "Genetic algorithms", *Theory and Practice of Uncertain Programming*, Springer, Berlin Heidelberg.

Liu, Z., Qu, W., Liu, W., Li, Z. and Xu, Y. (2014), "Resource preprocessing and optimal task scheduling in cloud computing environments", *Concurrency and Computation: Practice and Experience*.

Luo, F., Jin, H., Liao, X. and Zhang, Q. (2012), "Scheduling in an unstructured peer-to-peer-based high performance computing system", *Kybernetes*, Vol. 41 No. 9, pp. 1209-1215.

Mell, P. and Grance, T. (2009), "The NIST definition of cloud computing", *National Institute of Standards and Technology*, Vol. 53, p. 50.

Mezmaz, M., Melab, N., Kessaci, Y., Lee, Y.C., Talbi, E.G., Zomaya, A.Y. and Tuyttens, D. (2011), "A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems", *Journal of Parallel and Distributed Computing*, Vol. 71 No. 11, pp. 1497-1508.

Montazeri, A., Akbari, B. and Ghanbari, M. (2012), "An incentive scheduling mechanism for peer-to-peer video streaming", *Peer-to-Peer Networking and Applications*, Vol. 5 No. 3, pp. 257-278.

Pandey, S., Linlin, W., Guru, S.M. and Buyya, R. (2010), "A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments", *Advanced Information Networking and Applications (AINA), 24th IEEE International Conference on, April 20-23*, pp. 400-407.

Peng, Z. and Song, B. (2010), "Research on fault diagnosis method for transformer based on fuzzy genetic algorithm and artificial neural network", *Kybernetes*, Vol. 39 No. 123, pp. 5-1244.

Qin, W., Zhang, J. and Sun, Y. (2013), "Multiple-objective scheduling for interbay AMHS by using genetic-programming-based composite dispatching rules generator", *Computers in Industry*, Vol. 64 No. 3, pp. 694-707.

Ran, Y.-F., Xiong, G.-C., Li, S.-S. and Ye, L.-Y. (2010), "Study on deformation prediction of landslide based on genetic algorithm and improved BP neural network", *Kybernetes*, Vol. 39 No. 8, pp. 1245-1254.

Rius, J., Cores, F. and Solsona, F. (2013), "Cooperative scheduling mechanism for large-scale peer-to-peer computing systems", *Journal of Network and Computer Applications*, Vol. 36 No. 6, pp. 1620-1631.

Selvaraj, C. and Anand, S. (2012), "Peer profile based trust model for P2P systems using genetic algorithm", *Peer-to-Peer Networking and Applications*, Vol. 5 No. 1, pp. 92-103.

Shukla, A., Tiwari, R. and Kala, R. (2010), "Genetic algorithm", *Towards Hybrid and Adaptive Computing*, Springer, Berlin Heidelberg, Vol. 307, pp. 59-82.

Sivanandam, S.N. and Deepa, S.N. (2008), *Genetic Algorithm Optimization Problems*, Springer, Berlin Heidelberg.

Stallings, W (2008), "Operating systems: internals and design principles", Test.

Sun, B., Wang, W., Xie, X. and Qin, Q. (2010), "Satellite mission scheduling based on genetic algorithm", *Kybernetes*, Vol. 39 No. 8, pp. 1255-1261.

Tao, F., Feng, Y., Zhang, L. and Liao, T.W. (2014), "CLPS-GA: a case library and Pareto solution-based hybrid genetic algorithm for energy-aware cloud service scheduling", *Applied Soft Computing*, Vol. 19, pp. 264-279.

Wang, W., Zeng, G., Tang, D. and Yao, J. (2012), "Cloud-DLS: dynamic trusted scheduling for cloud computing", *Expert Systems with Applications*, Vol. 39 No. 3, pp. 2321-2329.

Wu, Z., Liu, X., Ni, Z., Yuan, D. and Yang, Y. (2013), "A market-oriented hierarchical scheduling strategy in cloud workflow systems", *The Journal of Supercomputing*, Vol. 63 No. 1, pp. 256-293.

Yang, Q., Wang, H., Hu, W. and Lijuan, W. (2010), "A scheduling model for temporally constrained grid workflow for distributed simulation system on grid", *Kybernetes*, Vol. 39 No. 8, pp. 1344-1350.

Yang, X.-S. (2014), "Chapter 5 – genetic algorithms", in Yang, X.-S. (Ed.), *Nature-Inspired Optimization Algorithms*, Elsevier, Oxford, pp. 77-88.

Corresponding author
Professor Ahmad Habibizad Navin can be contacted at: a.habibizad@srbiau.ac.ir