International Journal of Mechatronics, Electrical and Computer Technology

Vol. 3(9), Oct, 2013, pp. 286-313, ISSN: 2305-0543
Available online at: http://www.aeuso.org

Security and Fault aware Scheduling in Computational Grid

Mansour Noshfar[1*], Reza Javidan[2] and Mohammad Hossein Yektaei[3]

[1]Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Khuzestan, Iran

[2] Department of Computer Engineering and IT, Shiraz University of Technology, Shiraz, Iran

[3]Department of Computer engineering, Abadan Branch, Islamic Azad University, Khuzestan,Iran

*Corresponding Author's E-mail: mansour.noshfar@gmail.com

Abstract

Grid Computation is an issue that has received much attention from researchers in recent years. Its aim is to use the computational power of idle resources which have been distributed in different places and under different policies and security conditions. Therefore, one of the challenges facing this technology is the issue of security of jobs and the computational sites. Distributed jobs in computational sites may become problematic due to some infections and malwares. As a result, the risks and security levels should be considered; computing resources must be evaluated by resource owners for task execution, and scheduling should be based on requested users' security levels. This is the matter that has been ignored in the previous scheduling algorithms, which leads to waste of time and overhead. In this paper, a new method based on a combination of Genetic and Imperialism Competitive algorithm is presented to implement a security-aware scheduling and failure algorithm. The proposed method is compared with the previous methods such as Min-Min, Suffrage and genetic algorithms, has become near optimal and led to reduce the overhead caused by violation of security conditions. Additionally, Due to the usage of fault tolerance mechanisms, the performance of these mechanisms has been evaluated and it was found that the replication mechanism had the lowest failure rate and the check point mechanism had a direct effect on the performance and it should be fully supervised and be smart.

Keywords: Computational Grid, Security, Failure, Fault tolerance, Genetic Algorithm, Imperialism Competitive Algorithm

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 3(9), Oct, 2013, pp. 286-313, ISSN: 2305-0543
Available online at: http://www.aeuso.org

1. Introduction

In a large-scale grid [1], distributed resources belong to different administrative domains. Job executions are usually carried out between many virtual organizations in business applications or scientific applications for faster execution or remote interaction. However, grid security is a main hurdle to make the job scheduling secure, reliable and fault tolerant. If a host in the grid is under attack or malicious usage, its resources may not be accessible from remote sites. Thus, the jobs scheduled to that host may be failed because of the system infections or crashes.

Many algorithms have been developed for scheduling jobs in grids [2-8]. Unfortunately, most of the existing proposed scheduling algorithms had ignored the grid security while scheduling jobs onto geographically distributed grid sites with a handful of exceptions [3, 9, 10]. In a real life scenario, security threats always exist and the jobs are subject to failures or delays caused by infected hardware, software vulnerability, and distrusted security policy [3].

Despite the fact that many heuristics have been suggested for large-scale job scheduling [11_17], as pointed out by Song et al. [18], they are not applicable in a risky environment. Therefore, Song et al. developed security assurance and risk-resilient strategies and proposed eight job scheduling algorithms for use under various risky conditions to address these problems. Their work [18, 19] is built upon related works on Grid security, trust management and job scheduling. Their proposed security-assured job scheduling strategies consider the risk relationship between jobs and nodes using the security demand (SD) and the trust level (TL). The SD quantifies how much a user's job requires the assurance of secure computing services by a grid site or a cluster node. The TL quantifies how much a user can trust a site for successfully executing a given job. A job is expected to be successfully carried out when SD and TL satisfy a security assurance condition (SD $\leq$TL)

Vol. 3(9), Oct, 2013, pp. 286-313, ISSN: 2305-0543
Available online at: http://www.aeuso.org

during the job mapping process. Furthermore, they also proposed a space-time genetic algorithm (STGA) based on the messy encoding concept with a space-time guided search mechanism. Because its search is history-sensitive, STGA converges much faster than a traditional GA. We propose a combination of genetic algorithm and imperialism competitive algorithm for job scheduling. In this algorithm, we consider four kinds of fault-tolerance mechanisms, because each fault-tolerance mechanism has different requirements for gene encoding, we propose A new structure is used which in addition to supporting fault tolerance mechanisms, is applicable both as chromosomes in genetic algorithm and as country in the imperialism competition algorithm.

Simulation results show that our algorithm has shorter makespan then the Min-Min , Suffrage and Genetic algorithms and fault tolerance driver is presented , which attempts to repair the job using the tolerance mechanisms, in case of any job failure, at the time of violating security conditions. The rest of the paper is organized as follows: Section 2 presents a brief review of related work. In Section 3, we present a Security and failure Scheduling Model. Section 4 a brief review of ICA and genetic algorithm. We present proposed scheduling algorithm and the simulation results in Section 5 and 6. In finally we summarize the Conclusion and further research in Section 7.

2. Related works

Some scheduling strategies especially emphasized the importance of security awareness. Azzedin and Maheswaran [20] proposed a trust model that incorporates the security implications into scheduling algorithms. Humphrey and Thompson [21] proposed usage models for security-aware Grid computing but they did not elaborate on how to design a scheduler by incorporating the security concerns into collaborative computing over distributed cluster environment. Song et al. [18, 19]

Vol. 3(9), Oct, 2013, pp. 286-313, ISSN: 2305-0543
Available online at: http://www.aeuso.org

thought that a job distributed to a remote node may suffer from some infections or malicious attacks. Therefore, a job scheduler must consider the risk when dispatching jobs to remote nodes [18,19]. They proposed a job failure model to represent the risk level in job scheduling. There were two parameters in this model: the security demand (SD) and trust level (TL). SD represents a job's security requirement level, higher SD value represents this job has higher security requirement, so the job needs a more reliable node for job execution.

TL represents a node's secure level, higher TL value represents this node can provide a more reliable environment. So different jobs assigned to different nodes would have different risk levels. Based on the job failure model, they proposed three schedule strategies based on different risk levels: (1) *Secure* mode-jobs were only scheduled to those nodes which can ensure security. (2) *Risky* mode-jobs were scheduled to any available nodes without considering the risks between jobs and nodes, so it took all possible risks. (3) *f-risky* mode-jobs were scheduled to available nodes to take at most *f-risk*, where *f* was a probability.

Moreover, in [18], Song et al. proposed four types of scheduling strategies: (1) *Risky mode*-jobs were scheduled to any available nodes without considering risks between jobs and nodes, so it took all possible risks. (2) *Preemptive mode*-failed jobs would be moved to another node, and then restarted from the beginning. (3) *Replicated mode*-in order to accomplish jobs safely; a job would be duplicated to multiple nodes for better success rate of job execution. (4) *Delay-Tolerant mode* we would wait for a period of time to allow the nodes to have more time to deal with job execution. The simulation results mentioned that most fault tolerance algorithms are better than those algorithms without fault-tolerance techniques in makespan, except Replication. Daly examined methods of approximating the optimum checkpoint

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 3(9), Oct, 2013, pp. 286-313, ISSN: 2305-0543
Available online at: http://www.aeuso.org

restart strategy for minimizing application runtime on a system exhibiting Poisson single component failures [22]. In [23], Chtepen et al. proposed several heuristics to enhance the efficiencies of two fault-tolerant techniques: job checkpointing and job replication. They dynamically adapted the checkpointing frequency and the number of replicas to strong variations in grid availability based on monitored grid state, job characteristics, and collected historical information. Moreover, in order to combine the advantages of both techniques, they also proposed a hybrid scheduling strategy that switches at runtime between checkpointing and replication depending on the system load.

3. Security Scheduling Models in Computational Grid

In the proposed structure, firstly, security evaluation of sites, and demand for performing jobs under the specified security conditions are examined. Then the resources are selected. Thus, in this way, high overload due to selecting unauthorized resources is eliminated. The proposed structure is illustrated in Figure (1):

- **Job Presentation Unit**

This is the only unit that interacts with the user and after receiving a description of the job by the user, sends the job to the Resources and Security Evaluation Unit. This description includes job definitions, job id, security requirements.

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 3(9), Oct, 2013, pp. 286-313, ISSN: 2305-0543
Available online at: http://www.aeuso.org

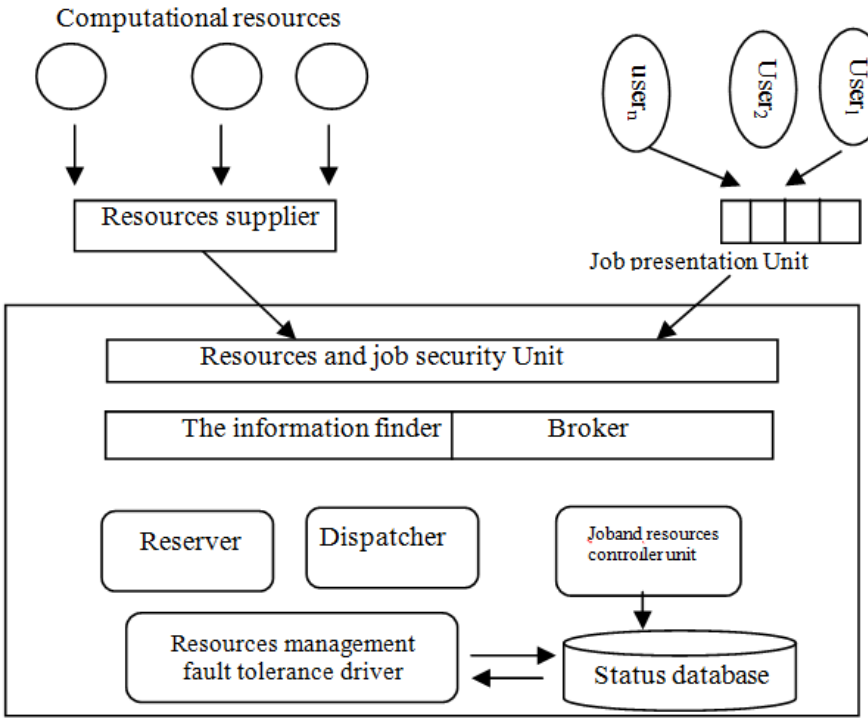


Figure 1: Security and Failure Scheduling Model

- **Resources Supplier**

It is the only unit that interacts with resources suppliers and keeps the information required for resources including processing speed of the resource, security and trust, for predefined resources and information produced for reservation.

- **Resources and job Security Evaluation Unit**

This unit receives job specifications from the job presentation unit and resources based on security conditions, and provide a list of possible resources for sending to the broker unit.

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 3(9), Oct, 2013, pp. 286-313, ISSN: 2305-0543
Available online at: http://www.aeuso.org

- **Reserver**

Reservation mechanism should be able to handle three cases [24]: 1) Preservation recognition; 2) Demand for preservation processing, and 3) Addressing confirmed preservation. In this method, the first two abilities are addressed by other units that are not included in the objectives of this paper. However, the reserver unit controls the third ability. At the time of run, the reserver unit will dispatch job and reserved resources, to the dispatcher unit.

- **Broker**

A broker is needed to support the grid concept in implementing jobs. In fact, this unit is responsible for scheduling jobs on different administrative domains.

- **Job and Resources Security Controller**

It is expected that after assigning resources in a certain interval, the desired service is be evaluated under security conditions and quality of services. In case of conflicting user demands under security conditions with resources in different intervals, this unit recognizes failure and introduces the resource as “failed” to the fault tolerance driver unit, and then demands for resources from the resources supplier unit.

- **Resources management Unit and Fault Tolerance Driver**

This unit is responsible for maintaining information about sources and scheduling and fault tolerance mechanisms using this information. Based on this responsibility, all job information such as failures history and the number of successful and unsuccessful jobson resource in the past is maintained. Resources management is in contact with fault tolerance driver.

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 3(9), Oct, 2013, pp. 286-313, ISSN: 2305-0543
Available online at: http://www.aeuso.org

3.1. Jobs Modeling with Security Requirements

Suppose there is a task T_i submitted by a user, T_i is modeled as a set of rational parameters, e.g., $T_i = (a_i, E_i, l_i, S_i)$, where a_i is the arrival time, l_i is data size (number of instructions). E_i is a vector of execution times for task T_i on each site in M, $E_i = (e_i^1, e_i^2, ..., e_i^m)$, which is resulted from l_i / c_i and S_i is a vector of $S_i = (s_i^1, s_i^2, ..., s_i^q)$ security levels and it is assumed q security services is needs. It is also assumed that there are $M = (M_1, M_2, ..., M_m)$ sites in the grid and each site is shown with $M_j = C_j, P_j)$ where C_j is the computing power and $P = (p_j^1, p_j^2, ..., p_j^q)$ is a vector of security levels which is provided by the site j. In this paper, we are considering (data integrity, Encryption and authentication) from security services for the security of resources and it is clear that these parameters may change if other security possibilities are provided by resources owners.

$$P_j = \omega_1 p_j^e + \omega_2 p_j^i + \omega_3 p_j^q \quad (1)$$

$$0 \le \omega_i \le 1 \qquad i = 1,2,3 \qquad \sum_{i=1}^{3} \omega_i = 1 \qquad i = 1,2,3$$

Where, p_j^e = Encryption, p_j^i =data integrity, p_j^a = authentication

To integrate security and job scheduler, a security deficiency function is used, and this function calculates and returns into values the differences between security levels required by the user to execute the job i and security levels provided in j computational sites. Security deficiency function of job T_i and site M_j is specified in Equation (2): this function is adapted from model [25]

$$SV(Si, Pj) = \begin{cases} 0 & if\ S_i \le p_j \\ S_i - P_j & otherwise \end{cases} \quad (2)$$

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 3(9), Oct, 2013, pp. 286-313, ISSN: 2305-0543
Available online at: http://www.aeuso.org

In scheduling, values obtained from this function are used to assign jobs to computational sites. The *0* return of this function means satisfaction and the difference between them shows deficiency and security shortages.

3.2. Job Failure Model

Job failure model is adapted to the model by Song et al [18]. In this model it is assumed that risk rates is a function of security level and the probability of job failure which is shown in Equation (3).

$$P_{fail}(i,j)=\begin{cases}0 & if \quad s_i \le p_i \\ 1-e^{\gamma(s_i-p_i)} & if \quad s_i > p_i\end{cases} \tag{3}$$

where,

s_i *: Job security demand,* p_i *:Site trust level,* γ *Failure coefficient*

According to Bawa's classification [26] from fault tolerance, expected running times of each of these mechanisms are discussed below:

Retry: is the simplest technique to repair failure, and this simplicity is due to retrying the same job after the failure in the same site. Assuming assign *i*-th job to site j, the expected running time is in equation (4):

$$E(T_j^i)=E_{\text{Retry}}(T_j^i)=e_i+\frac{1}{2}e_i.p_{fail}^1+\frac{1}{2}e_i.p_{fail}^2 \tag{4}$$

Alternate Site: in this technique the failed job is re-assigned to another site.

The expected running time for job i on x $x\in\{j,k,q\}$، site is obtained from equation (5):

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 3(9), Oct, 2013, pp. 286-313, ISSN: 2305-0543
Available online at: http://www.aeuso.org

$$E(T_x^i) = E_{Alternate}(T_x^i) = (1 - p_{failx}^i).e_i + p_{failx}^i.(\frac{1}{2}e_x + Mig_{y,z}^i) \quad (5)$$

$$x \in \{j,k,q\}, \qquad y,z \in \{j,k)$$

Estimated transfer time for a job transfer from site i to z can be defined through equation where, l_i is job size of i and $BandWidth_{y,z}$ is the network bandwidth between sites y and z.

$$Mig_{y,z}^i = \frac{l_i}{BandWidth_{y,z}}$$

Check point: in this technique, the failed job is clearly transferred on another site and will re-run from the stored check point onward.

The expected running time of job i in sites j, k, and q are as equation (6):

$$E(T_X^i) = E_{chkpnt_{i,k,q}}(T_X^i) = (1 - p_{failx}^i) \times (e_X + \left\lfloor \frac{e_X}{PRT} \right\rfloor \times OH_X) + p_{failx}^i \times (\frac{e_X}{2} + \left\lfloor \frac{e_X}{PRT} \right\rfloor \times OH_X + Mig_X^i \quad (6)$$

$$x \in \{j,k,q\}$$

In case of any failure, the remaining job i running, which has not been executed in site j, should be transferred into sites k and q, which is defined as Equation (7):

$$RM_j^i(jmkmq) = e_X - \frac{e_X}{2} \quad (7)$$

PRT is time period for check point and OH_x is the overload resulted from check point in site x, which is $x \in \{j,k,q\}$. Check point operation $\left\lfloor \frac{e_X}{PRT} \right\rfloor$ is done after the successful implementation of the job. As a result, the total overload of check point will be as:

$$\left\lfloor \frac{e_X}{PRT} \right\rfloor \times OH_X$$

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 3(9), Oct, 2013, pp. 286-313, ISSN: 2305-0543
Available online at: http://www.aeuso.org

Replication: this technique works to ensure simultaneous implementation of jobs on multiple sites with all security requirements even if $(s_i > p_i)$. In this method, if one job is completed, all other replica will stop working, and in the event of failure in one of the replications, the results of executing the next replication are to be considered. In this mechanism, the expected running time is in compliance with model [18].

4. Basic Evolutionary Algorithms

4.1. Genetic Algorithm

In GA [27], a candidate solution for a specific problem is called an individual or a chromosome and consists of a linear list of genes. Each individual represents a point in the search space, and hence a possible solution to the problem. A population consists of finite number of individuals. Each individual is decided by an evaluating mechanism to obtain its fitness value. Based on this fitness value and undergoing genetic operators, a new population is generated iteratively with each successive population referred to as a generation. Sexual reproduction allows some exchange and re-ordering of chromosomes, producing offspring that contain a combination of information from each parent.

This is the recombination operation, which is often referred to as crossover because of the way strands of chromosomes crossover during the exchange. Diversity in the population is achieved by mutation. Genetic algorithms are ubiquitous nowadays ,having been successfully applied to numerous problems from different domains, including optimization, automatic programming, scheduling, machine learning, operations research, bioinformatics, and social systems.

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 3(9), Oct, 2013, pp. 286-313, ISSN: 2305-0543
Available online at: http://www.aeuso.org

4.2. Imperialist Competitive Algorithm

In ICA [28], a candidate solution for a specific problem is called a country in which some having the least cost are selected as imperialist states and rest form the colonies of these imperialists. The division of all the colonies of initial countries is based upon the fitness value. The imperialist states together with their colonies form some empires. The colonies in each of the empire start moving towards their imperialist countries, based upon a simple model of assimilation policy. The total power of an empire is defined by the power of imperialist country and percentage of mean power of its colonies. Then the imperialistic competition begins among all the empires.

Any empire that is not able to succeed in this competition cannot increase its power shall be eliminated. Based upon this competition, the power gradually increases for some empires and decreases for others. This results in the collapse of weak empires. The movement of colonies toward their relevant imperialist states along with competition among empires and also the collapse mechanism cause all the countries to converge to a state in which their exist just one empire in the world and all the other countries are colonies of that empire. In this new world, colonies have the same position and power as the imperialist.

5. The Proposed Algorithm

In combining GA and ICA, some of the best chromosomes of population are selected as emperors and the remaining population is also considered as colony. Emperors draw colonies toward themselves due to their strength level. Also, the total power of an emperor depends on its both constituent parts: emperors (the core) and colonies. The imperialism competition begins in this level in a way that more powerful empires are seeking to attract colonies that belong to weaker imperials and

Vol. 3(9), Oct, 2013, pp. 286-313, ISSN: 2305-0543
Available online at: http://www.aeuso.org

finally weaker empires are gradually eliminated until a unique emperor can be produced or an acceptable optimum solution is reached.

5.1. New solutions encoding

In the grid system, each computational site supports one of the following three mechanisms: Retry, Alternate, check point and Replication. Furthermore different fault-tolerance mechanisms have different requirements for encoding in the genetic algorithm, we propose a structure is used which in addition to supporting fault tolerance mechanisms, is applicable both as chromosomes in genetic algorithm and as country in the imperialism competition algorithm. Figure (2) is showing a sample way of encoding solutions. This structure is a list of variable-length integer sequences.

Each integer sequence represents a gene, whose length depends on the type of the adopted fault tolerance mechanism. If Retry is supported, the corresponding integer sequence consists of two integers, where the first integer denotes the job id and the second denotes the allocated node id. If Alternate or check point techniques is supported, there are four integers in the integer sequence, where the first integer represents the job id and the remaining integers represent the sequence of node ids. All of the nodes specified in the same sequence support the same fault-tolerance mechanism, either alternate or check point. According to the sequence of node ids, the job will move to the next node for re-execution whenever a failure occurs.

5.2. Objective function

The objective of job safety scheduling model is to minimize the job finishing time and to increase safety with minimal security deficiency. The mathematical model is as follows:

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 3(9), Oct, 2013, pp. 286-313, ISSN: 2305-0543
Available online at: http://www.aeuso.org

$$Minmize: \sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij}.FinishTime(T_i,P_j),\quad Maxmize: \sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij}.SV(S_i,P_j) \tag{8}$$

$$xij = \begin{cases} 1 & Job\ T_i\ is\ allocated\ to\ site\ M_j \\ 0 & else \end{cases} \quad , \sum_{j=1}^{m} x_{ij} = 1, i = 1,2,...,n$$

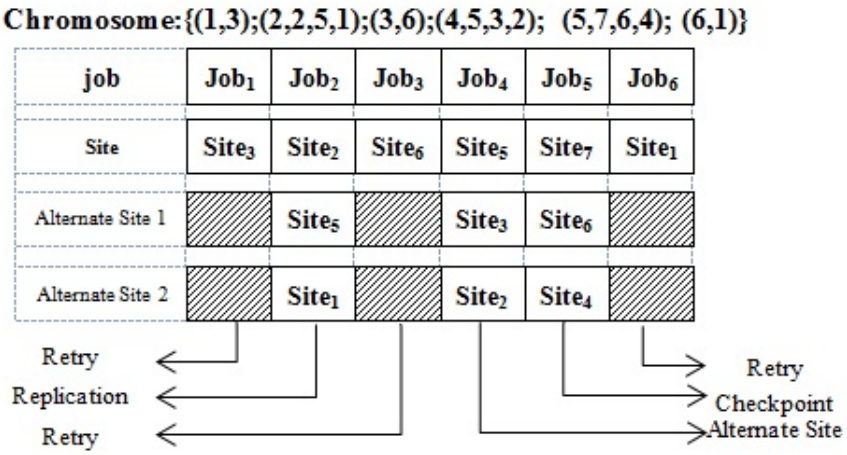


Figure 2: new structure encoding

the multi-objective optimization model can be transformed into the following single-objective model:

$$\min = \sum \left[\omega_1.FinishTime\ (T_i,P_j) + \omega_2.(1 - SV(S_i,P_j))\right] \tag{9}$$

$$0 \le \omega_1,\omega_2 \le 1, \omega_1 + \omega_2 = 1$$

Given the importance of evaluation criteria of scheduling, ω_1,ω_2 is showing weight in the finish time and the security. In the above mentioned equations $FinishTime(T_i,P_j)$.

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 3(9), Oct, 2013, pp. 286-313, ISSN: 2305-0543
Available online at: http://www.aeuso.org

$$FinishTime(T_i, P_j) = a_i + b_{ij} + E(T_j^i) \quad (10)$$

Where a_i is the arrival time, b_{ij} time of connecting and data transfer to site j, $E(T_j^i)$ job of execution time on the site j

5.3. Description proposed method

After genetic procedure, some of the best chromosomes of population are selected as emperors and the remaining population is also considered as colony. The colonies are distributed among imperialists based on imperialist's power. For calculating the power of imperialists, first, the normalized cost of an imperialist is applied based on Equation (11).

$$\#\# \quad N.T.P_{I_i} = \max\{T.P_{I_i}\} - T.P_{I_n} \# \quad (11)$$

Where, $T.P_{I_n}$ is the cost of nth imperialist and $N.T.P_{I_i}$ is its normalized cost which is equal to the deviation of the maximum total completion time from the nth imperialist cost. Then the power of each imperialist is calculated according to equation (12).

$$P_n = \left| \frac{N.T.P_{I_i}}{\sum_{i=1}^{N_{imp}} T.P_{I_j}} \right| \quad (12)$$

This method, considering its numerous computations, results in prolonged algorithm running time. In addition, the empire relative power where $\max\{T.P_{I_j}\} = T.P_{I_i}$ equals zero and results in the elimination of the imperialist in the preliminary stage of algorithm execution. While its Cost may not be much higher than other imperialist costs, here we attempt to propose another method for calculating relative power (Equation 13).

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 3(9), Oct, 2013, pp. 286-313, ISSN: 2305-0543
Available online at: http://www.aeuso.org

$$P_n = \frac{1 - N.T.P_{I_i} \Big/ \sum_{i=1}^{N_{imp}} N.T.P_{I_i}}{N_{imp} - 1} \quad (13)$$

This method, compared to the previous method has a smaller volume of computations and is much faster in execution. In addition, this method does not result in zero empire relative power with the highest Cost in preliminary stages. Through calculating the probability of taking each empire, the process of giving a colony to any of the empires is proposed based on figure (3).

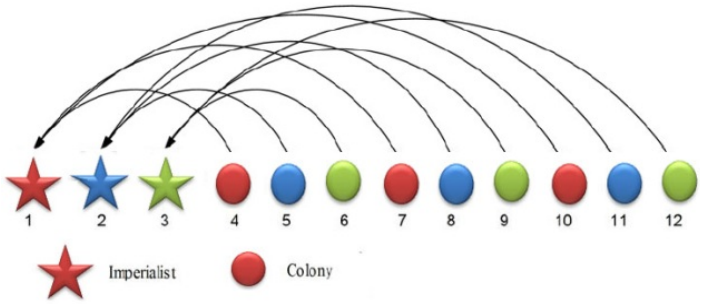


Figure 3: process of giving a colony to empires

5.4. Assimilation

Imperialists countries started to improve their colonies. This fact has been modeled by moving all the colonies toward the imperialist. The assimilating operator is similar to the crossover operator used in genetics. The proposed pseudo-code algorithm is described as follows:

Step 1: Initializing parameters;

Step 2:

2.1. Generate some random Chromosome

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 3(9), Oct, 2013, pp. 286-313, ISSN: 2305-0543
Available online at: http://www.aeuso.org

2.2. use **Equation (9)** to Define the optimization problem;

Step 3: For i = 1, 2, … ,*Npop* do:

3. 1. *Selection*;

3. 2. *Crossover*;

3. 3. *Mutation*;

Step 4:

4.1. Select some powerful Chromosome as empires;

4.2. Randomly allocate remain Chromosome to different empires;

4.3. Initialize the empires with imperialists cost function $T.P_i$

Step 5: Decade loop: $N_{generation}=N_{generation}+1$

Step 6: For i = 1,2, … , N_{imp} do

6.1. *Countries assimilation policy*: Move the Chromosome of each country toward their empires.

6.2. *Countries revolutionary*;

6.3. *Competition*; Pick the weakest country from the weakest empire and give it to the rightful empire.

6.4. *Elimination*; Eliminate the powerless empires.

Step 7: *Terminating Criterion Control*; Repeat Steps 6-7 until a terminating criterion is satisfied.

6. Simulation results

To evaluate our proposed algorithm, we have constructed a simulator written in the C# language. This program includes major classes such as JOB, SITE, SCHEDULE, EVL_SEC, and FAULT_TL considered for simulating resources and users of three scenarios. Which are shown in Table 1.

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 3(9), Oct, 2013, pp. 286-313, ISSN: 2305-0543
Available online at: http://www.aeuso.org

Table 1: simulating resources and users of three scenarios

	Small – Medium –Large
Site	32-64-128
Jobs	512-1024-2048
Data size	50-80 MB ,800MB-1GB,1-2GB
Network bandwidth	2-8 Mbps
Job security demands(S_i)	U[0,1]
Node trust level (Pj)	U[0,1]
Failure coefficient	3
Security services	Integrity , Encryption and Authentication
Weight security services	Integrity =0.3,Encryption =0.5,Authentication =0.2

The Min-Min and sufferage algorithms [27] are adopted for comparison with our proposed algorithm. The Min-Min algorithm assigns jobs one at a time to the computational nodes. At each step, first it selects the unscheduled job with the smallest size and then assigns that job to the computational node having the earliest completion time after executing that job. On the other hand, the sufferage algorithm selects the unscheduled job that has the largest sufferage value at each step, where the sufferage value of a job is the difference between its second earliest completion time and its earliest completion time. The selected job will then be assigned to the computational node that has the earliest completion time for that job.

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 3(9), Oct, 2013, pp. 286-313, ISSN: 2305-0543
Available online at: http://www.aeuso.org

6.1. Parameter tuning

One of important component of metaheuristic algorithm is calibration of parameters which impresses on performance of algorithm. In this investigation we employed the Taguchi method for this goal.

Table 2 –Factors and Levels

Factors	Symbol	Levels	Small(256;16)	Medium(512;32)	Large(1024;64)
$(N_{country}, N_{imp})$	A	3	A(1)=(100,8) A(2)=(120,10) A(3)=(200,16)	A(1)=(200,16) A(2)=(300,24) A(3)=(500,30)	A(1)=(500,30) A(2)=(800,32) A(3)=(1000,40)
P_R	B	3	B(1)=0.3 B(2)=0.5 B(3)=0.7	B(1)=0.1 B(2)=0.3 B(3)=0.4	B(1)=0.3 B(2)=0.5 B(3)=0.7
Zeta(ε)	C	3	C(1)=0.01 C(2)=0.03 C(3)=0.05	C(1)=0.01 C(2)=0.05 C(3)=0.07	C(1)=0.01 C(2)=0.05 C(3)=0.09
$N_{generation}$	D	3	D(1)=500 D(2)=700 D(3)=800	D(1)=700 D(2)=900 D(3)=1200	D(1)=900 D(2)=1200 D(3)=1500

Taguchi [29] developed a family of FFE matrices which eventually reduce the number of experiments, but still provide sufficient information. In Taguchi method, orthogonal arrays are used to study a large number of decision variables with a small number of experiments. In Taguchi method, the word "optimization" implies "determination of best levels of control factors". In turn, the best levels of control factors are those that maximize the Signal-to-Noise ratios. The Signal-to-Noise ratios are log functions of desired output characteristics. The experiments that are conducted to determine the best levels, are based

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 3(9), Oct, 2013, pp. 286-313, ISSN: 2305-0543
Available online at: http://www.aeuso.org

on "Orthogonal Arrays", are balanced with respect to all control factors and yet are minimum in number. This in turn implies that the resources (materials and time) required for the experiments are also minimum. Taguchi has created a transformation of the repetition data to another value which is the measure of variation. The transformation is the signal-to-noise (S/N) ratio which explains why this type of parameter design is called robust design [29, 30]. Here, the term ‘’signal’’ denotes the desirable value (mean response variable) and ‘’noise’’ denotes the undesirable value (standard deviation). So the S/N ratio indicates the amount of variation presents in the response variable. The aim is to maximize the signal-to-noise ratio. Taguchi classifies objective functions into three categories: the smaller-the-better type, the larger-the-better type, and nominal-is best type. Since almost all objective functions in scheduling are classified in the smaller-the-better type, their corresponding S/N ratio [29] is:

$$S/N\ ratio = -10 \times \log\left(\frac{1}{k}\sum_{i=1}^{k}(objectivefunction)^2\right) \tag{14}$$

That k represents the number of experiment for each problem.

The parameters in our proposed calibration of parameters and their levels are shown in Table 2. There are three replicates for each combination, summing up to 9 instances. After experimental design for mentioned problem, the results obtained by Taguchi method after obtaining the results of the Taguchi experiment for all the trials, the optimal level of the factors A, B, C, and D are shown in Table 3. Results are indicated in Figures 4, 5, and 6.

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 3(9), Oct, 2013, pp. 286-313, ISSN: 2305-0543
Available online at: http://www.aeuso.org

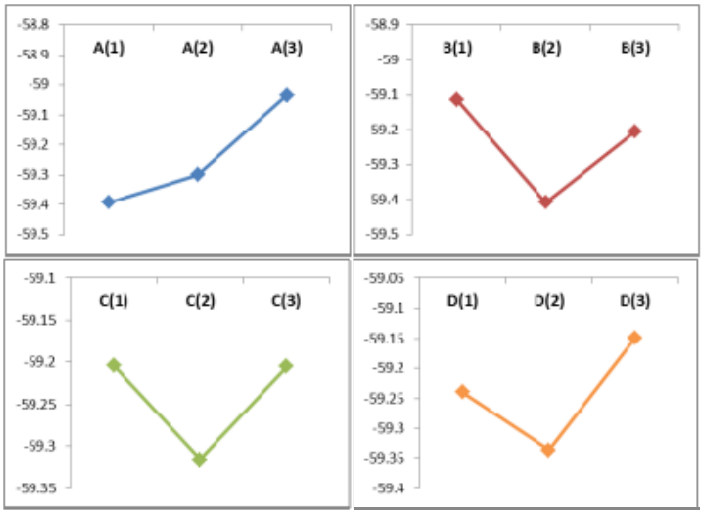


Figure 4: the mean of S/N ratio-Small

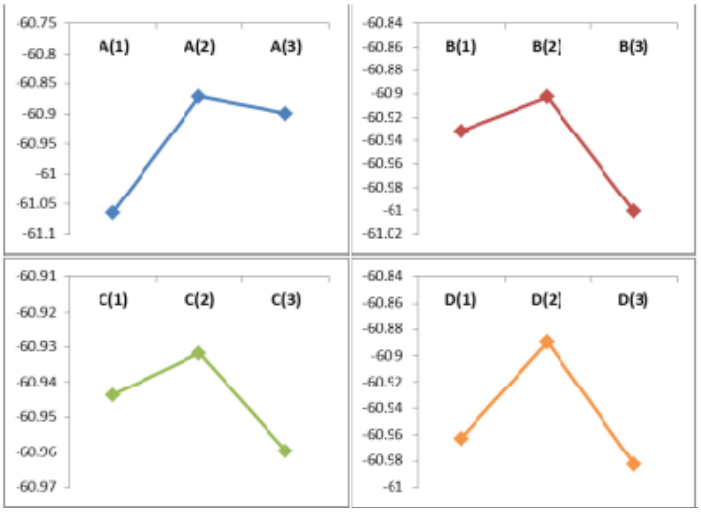


Figure 5: the mean of S/N ratio -Medium

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 3(9), Oct, 2013, pp. 286-313, ISSN: 2305-0543
Available online at: http://www.aeuso.org

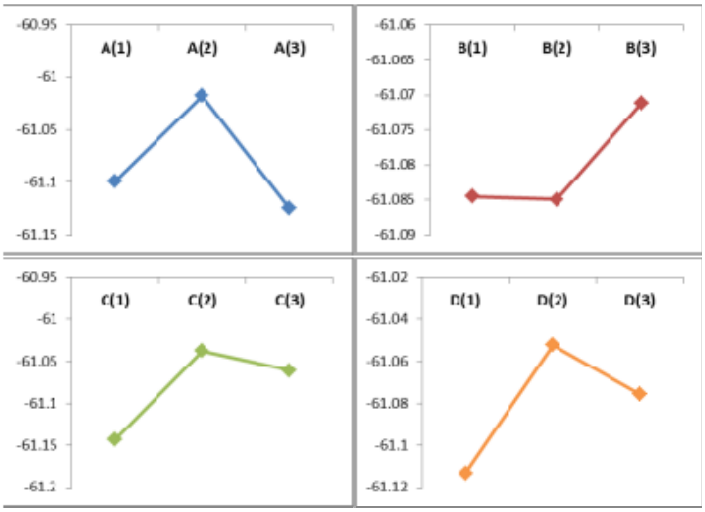


Figure (6) the mean of S/N ratio-Large

Table 3: Optimal levels of factor

Factor	Optimal Level
A	Small:(100,8) , Medium: (100,8),Large: (1000,40)
B	Small:0.5 , Medium: 0.4, Large: 0.5
C	Small:0.03 , Medium: 0.07, Large: 0.01
D	Small:800 , Medium: 1200, Large: 900

6.2. Makespan

The results in figure (7) show that the performance of a hybrid algorithm are composed of genetic and imperialism competition (G-ICA) in terms of makespan in three scenarios of grid, small, medium and large sizes acts better that Min-Min algorithms and Suffrage, yet another observable point is that an increase in the number of jobs at the size of the problem results in an increase in makespan for all algorithms.

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 3(9), Oct, 2013, pp. 286-313, ISSN: 2305-0543
Available online at: http://www.aeuso.org

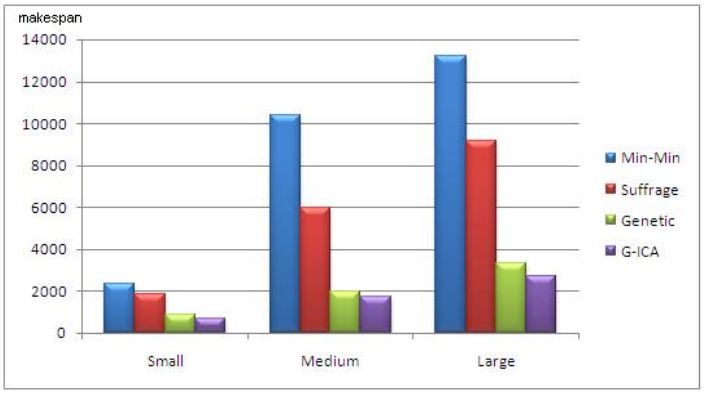


Figure 7: makespan(s)

6.3. Failure rate

In figure (8), failure rates for scenarios of low and high failure probability have been evaluated in the order of security demands, in a range of $U[0,0.7]$ and $U[0.7,1]$, and a reliance level in the range which:

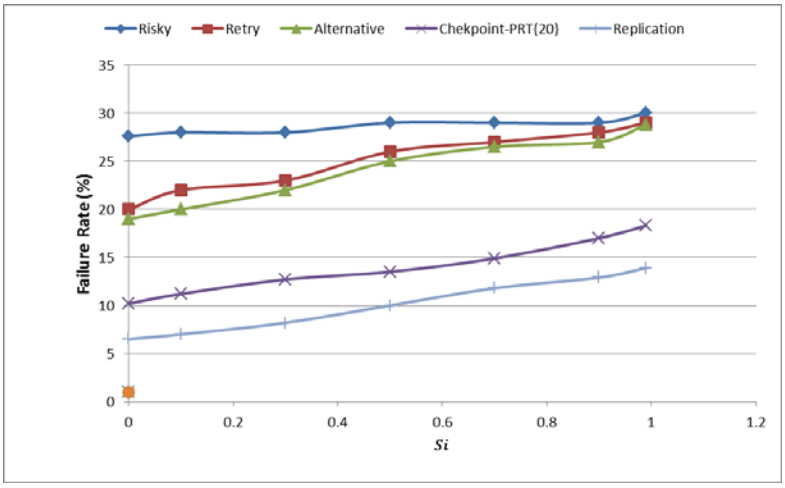
In *low failure probability*: in this scenario the best results have been obtained from replication fault tolerance mechanisms, checkpoint, and help site. In the check point mechanism, storing check points in $PRT = \{20,100,500\}$ periods have been examined. Since the failure probability is low in this case, in times of *20* and *100* seconds, the system is involved with storing conditions of un-needed jobs and replication mechanism failure rate is low compared with other mechanisms due to simultaneous execution of one job in several computational sites. In this mechanism a computational reserved site is also used as a guarantee of performance.

In *high failure probability*: the failure job rates in this experiment are at best, related the replication and check point mechanism. To increase the velocity as well as avoiding repetitive jobs in check point mechanism at high failure probability, more check points are

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 3(9), Oct, 2013, pp. 286-313, ISSN: 2305-0543
Available online at: http://www.aeuso.org

needed to store performed jobs to be used in probable migrations. Thus, a short interval in this case, should be considered providing check point. In terms of failure, help site and retry mechanisms have less priorities in this process.



	Low failure probability	High failure probability
S_i	0-0.5	0.7-1
Risky	27.6-29	29-30
Retry	20-26	27-29
Alternate Site	19-25	26.5-28.8
Chekpoint-PRT{20}	10.2-13.5	14.9-18.3
Replication	6.5-11	11.8-13.9

Figure 8: Failure Rate

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 3(9), Oct, 2013, pp. 286-313, ISSN: 2305-0543
Available online at: http://www.aeuso.org

Conclusion

Development and use of applications that are executed on computational grids are growing and due to the dynamic, heterogeneity and distribution characteristics of resources in different places with different administrative and security policies, paying attention to the safety and reliability of the resources is very critical. Due to this, in scheduling, suffice to reduce the overall makespan in this environment is not sufficient thus, paying attention to the quality of services and security is of great importance. In this paper, in addition to address the concept of security based scheduling, proposing to combine genetic and Imperialism Competitive Algorithms, benefits of both algorithms is used to solve the scheduling problem which is a NP-hard problem.

The results of the experiments indicated that in addition to improving jobs makespan compared to Min-Min , suffrage and genetic algorithms, security resources at the time of assigning jobs to them are examined and assigning jobs in Risky conditions and thus, imposing additional overload is avoided and this results in running jobs at the highest security levels. Through examining jobs failure rates, the study found that replication mechanism owns the lowest failures and check point mechanism must have a direct effect on the efficiency and should be under the supervision of the number of failure events at the time of storing check points. As a future strategy, establishing a smart way for determining the number or interval of check points can be pointed out.

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 3(9), Oct, 2013, pp. 286-313, ISSN: 2305-0543
Available online at: http://www.aeuso.org

References

[1] I. Foster, C. Kesselman, & S. Tuecke. The anatomy of the grid: enabling scalable virtual organizations, International Journal of High Performance Computing Applications, 15(3), 2001, 200-222.

[2] A. Dogana, & F. Ozguner, Scheduling of a meta-task with QoS requirements in heterogeneous computing systems, Journal of Parallel and Distributed Computing, 66(2), 2006, 181-196.

[3] S. Song, K. Hwang, & Y. Kwok, Risk-resilient heuristics and genetic algorithms for security-assured grid job scheduling, IEEE Transactions on Computers, 55(6), 2006, 703-719.

[4] J.H. Abawajy, Fault-tolerant scheduling policy for grid computing systems, Proc. IEEE 18th International Parallel and Distributed Processing Symposium (IPDPS04), Santa Fe, NM, 2004, 238-244.

[5] J. Kim, S. Shivle, H. J. Siegel, A.A. Maciejewski, T.D. Braun, M. Schneider, S. Tideman, R. Chitta, R.B. Dilmaghani, R. Joshi, A. Kaul, A. Sharma, S. Sripada, P. Vangari, S.S. Yellampalli, Dynamically mapping tasks with priorities and multiple deadlines in a heterogeneous environment, Journal of Parallel and Distributed Computing, 67(2), 2007,154-169.

[6] K. Kaya, C. Aykanat, Iterative-improvement-based heuristics for adaptive scheduling of tasks sharing files on heterogeneous master-slave platforms, IEEE Transactions on Parallel and Distributed Systems ,17 (8) ,2006,883-896.

[7] S. Baskiyar, & C. Dickinson, Scheduling directed a-cyclic task graphs on a bounded set of heterogeneous processors using task duplication, Journal of Parallel and Distributed Computing, 65(8) ,2005, 911-921.

[8] X. Qin, & H. Jiang, A novel fault-tolerant scheduling algorithm for precedence constrained tasks in real-time heterogeneous systems, Parallel Computing, 32(5-6), 2006, 331-356.

[9] C. Jiang, C. Wang, X. Liu, & Y. Zhao, Adaptive Replication Based Security Aware and Fault Tolerant Job Scheduling for Grid Computing, accepted to appear in Proc. 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD07), Qingdao, 2007.

[10] C. Jiang, C. Wang, X. Liu, & Y. Zhao, A Fuzzy Logic Approach for Secure and Fault Tolerant Grid Job Scheduling, accepted to appear in Proc. 4th International Conference on Autonomic and Trusted Computing (ATC- 2007), Hongkong, 2007.

[18] S.Song, Y.-K. Kwok, K.Hwang,Risk-resilient heuristics and genetic algorithms for security-assured grid job scheduling, IEEE Transactions on Computers 55 (6) (2006) 703_719.

[19] S.Song, Y.-K. Kwok, K. Hwang, Security-driven heuristics and a fast genetic algorithm for trusted grid job scheduling, in: Proc. 19th IEEE International Parallel and Distributed Processing Symposium, IPDPS'05, IEEE Computer Society, Washington, DC, USA, 2005, 65.1_65.1.

Vol. 3(9), Oct, 2013, pp. 286-313, ISSN: 2305-0543
Available online at: http://www.aeuso.org

[20] F. Azzedin, M. Maheswaran, Integrating trust into grid resource management systems, in: Proc. Intl Conf. Parallel Processing, 2002, pp. 47_54.

[21] M. Humphrey, M. Thompson, Security implications of typical grid computing usage scenarios, in: Proc. High Performance Distributed Computing, 2001, pp.355_362.

[22] J. Daly,A higher order estimate of the optimum checkpoint interval for restart dumps, Future Generation Computer Systems 22 (3) (2006) 303_312.

[23] M. Chtepen, F. Claeys, B. Dhoedt, F.D. Turck, P. Demeester, P. anrolleghem, Adaptive task checkpointing and replication: Towards efficient fault-tolerant grids, IEEE Transactions on Parallel and Distributed Systems 20 (2) (2009)180_190.

[24] JingCh.,Ling-fu K.,2006,Integrating Reservation Mechanism into Grid Resource Management, Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'06);pp,470-473.

[25] T.Xie and X. Qin, "Performance evaluation of a new scheduling algorithm for distributed systems with security heterogeneity," J. Parallel Distrib.Comput., vol. 67, pp. 1067-1081, 2007.

[26] K.Bawa,R.Singh."Comparative Analysis of Fault Tolerance Techniques inGrid Environment," International Journal of computer Application vol. 41, pp. 21-25, March 2012 2012.

[27] T. Braun, D. Hensgen, R. Freund, H. Siegel, N. Beck, L. Boloni, M. Maheswaran, A. Reuther, J. Robertson, M. Theys, B. Yao, A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems, Journal of Parallel and Distributed Computing 61 (6) (2001) 810_837.

[28] E.Atashpaz-Gargari and C. Lucas, "Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition," in Evolutionary Computation, 2007. CEC 2007. IEEE Congress on, 2007, pp. 4661-4667.

[29] P. J. Ross, Taguchi techniques for quality engineering: loss function,orthogonal experiments, parameter and tolerance design . 2005.

[30] M. Phadke, Quality Engineering using Design of Experiments, in: K. Dehnad (Ed.), Quality Control, Robust De-sign, and the Taguchi Method, Springer US, ISBN 978-1-4684-1474-5, 31–50, doi:"10.1007/9781468414721 3", "http://dx.doi.org/10.1007/9781468414721 3", 1988.

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 3(9), Oct, 2013, pp. 286-313, **ISSN**: 2305-0543
Available online at: http://www.aeuso.org

Authors

Mansour Noshfar received the B.Sc degree in Computer Engineering (Software) from Islamic Azad University, Mahshahr Branch. He is currently M.Sc. student in Computer Engineering (Software) in Islamic Azad University Science and Research Branch of Ahvaz, Iran. His research interests include scheduling, network, grid computing and Computer programming.

Reza Javidan received the B.Sc degree in Computer Engineering (hardware) from Isfahan University in 1993. He received M.Sc. and Ph.D. degree in Computer Science and Engineering (Artificial Intelligence) from Shiraz University in 1996 and 2007, respectively. He currently is an assistant professor and works as a lecturer in Shiraz University of Tecnology.He has more than 30 papers that were published in different national and international conferences and Journals. He also published two books about underwater systems. Dr Javidan major research interests include Pattern Recognition, Image Processing, Artificial Intelligence, sonar systems and Computer Vision.

Mohammad Hossein Yektaie has been graduated in BSc of Mathematics and its application in computer science from Isphahan University, Isphahan, Iran and Master Degree in Software Engineering from La Rochelle University, La Rochelle, France. He has been obtained his Doctorate or PH.D. in Computer Science from La Rochelle University. He has taught several courses graduate and undergraduate in the field of computer engineering and IT. He has published a lot of articles about his work. His research interests are AI, Pattern recognition, Cloud Computing and NLP. Recently he is working on a Machine translation program.