

# A Scheduling Algorithm for Increasing the Quality of the Distributed Systems by using Genetic Algorithm

Arash Ghorbannia Delavar, Vahe Aghazarian, Sanaz Litkouhi and Mohsen Khajeh Naeini

**Abstract**—This article will present a scheduling algorithm for increasing the quality of the distributed system by using genetic algorithm. The proposed algorithm OQSG is more efficient in comparison with the previous algorithms. This proposed algorithm can be reliable like the other algorithms, regarding the effective parameters used for the available scheduling. OQSG algorithm can create more efficient parameters by using the priority function that is more reliable by using the scheduling the most used works. The proposed algorithm can use the least possible time in comparison with the previous algorithm, using new techniques in the purpose function. This condition helps us to use the effective parameters, which increase the system efficiency, after several assessment used in the heterogeneous purpose function. The proposed algorithm can have the optimum selection in processing and allocating the resources, considering maximum and minimum scheduling of works. This work is fulfilled when we decrease the complexity of time. Finally, we could create the efficiency of distributed systems with high quality according to different simulation results.

**Index Terms**—RQSG-I; Genetic Algorithm; Scheduling; Grid Systems; Dependent Task; OQSG; Grid Systems; RQSG.

## I. INTRODUCTION

The main purpose of grid systems was optimized use from sources and paralleling of time computing for gaining the less Return average time. In the grid environments is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities [1].

In year of 2000 were done typical studies on scheduled algorithms that their Problem was summation of nodes as prioritized queue. For this purpose we should study dependent tasks with scheduled algorithms to decrease waiting in processing sources. There is a problem in task sending with exploring of RQSG algorithm at multi surface type. This method in done by a process which can be followed by specifications for doing it with dependent tasks and graphs weighting.

In the past Distributed systems were used from the sources and parallelize the identified process time. In information

systems, the most expensive item is the information itself, if the amount of information would be high, by layering the information, you can classified the amount of information to optimize the searching time. That's why in the distributed systems because huge amount of information the speed will be decreased [1].

To achieve the objectives distributed systems should be classified in two phases we studied:

The first phase of software, with previous exact algorithms optimizes a new algorithm to be presented to the previous case with the fault tolerance and dependable more done.

Proposed algorithm must be the same environmental conditions until obtained output is optimal than previous mode. But in the second phase of the hardware, network's communications environment and processors that can speed run processing time processes on independent processing than similar processes with regard to the above conditions reduce and with using the proposed algorithm can to reduced allocation of Time allocation process program and we can with the above algorithm, faster output program to optimize. Also, consuming time for a number of programs to implement applications with high volumes of data taken, with limiting the scheduler program and the proposed algorithm in the Process Scheduler program sweep mode with independent processing resources, we could able put it to use [2].

The initial algorithm and RQSG one at the first we reviewed the related tasks on dynamic tasks scheduling in calculation grid then we introduced ourselves method for solving the reconstruction problem and discovery at grid tasks scheduling .we prioritized the task priorities at RQSG algorithm using a new technical method and graph index parameters. This task prevented from processing time losses.

In this paper is used from roulette wheel schema for creating the first generation and also used from Gridsim for simulations.

## II. THE INITIAL ALGORITHM AND RQSG-I

One at the first we reviewed the related tasks on dynamic tasks scheduling in calculation grid then we introduced ourselves method for solving the reconstruction problem and discovery at grid tasks scheduling .we prioritized the task priorities at RQSG algorithm using a new technical method and graph index parameters. This task prevented from processing time losses.

At the first we reviewed the related works on dynamic task scheduling at grid calculations, then we introduced own method for solving the reconstruction problem and also discoveries at grid tasks scheduling. Although most of

Manuscript received March 20, 2011  
Arash Ghorbannia Delavar, Payam Noor University, faculty of engineering, Tehran, Iran, a\_ghorbannia@pnu.ac.ir  
Vahe Aghazarian, Islamic Azad University, Central Tehran Branch Tehran, Iran, v\_aghazarian@iauctb.ac.ir  
Sanaz Litkouhi, Payam Noor University, Tehran, Iran, slitkouhi@pnu.ac.ir  
Mohsen Khajeh Naeini, Payam Noor University, faculty of engineering, Iran, Tehran, mohsen\_kh\_n@yahoo.com.

researchers have different comments on tasks dynamic scheduling but the most common of real function at task scheduling problems is makspan function. In grid there is this possibility that makspan from non optimized scheduled has more time than interest makspan because the networks calculation power varies with time. [4]

In the past task scheduling was proposed for super computers and parallel computers .[5][6][7][8]

Hwang and Kesselman say that grid environment isn't reliable by nature. They proposed a failure detector service and a flexible item for failure controlling as a failure tolerant mechanism on grid. In grid environment breaking the task running is done because of some reasons such as network incompetence, bad condition of sources, unreliability to elements of needed software's. So the grid system should be able to detect and control the failure and also be able to protect of reliable running at the same time.

The scheduling of task on grids is an active area for research subjects and is focused on heuristic techniques. Using the executive model which select the sources and the allocated tasks to it prepare the limited and reliable algorithms for grid sources. Hwang and his workers proposed the failure detection service and also trade of with grid failures. Limaye and his co workers developed checkpoint/restart mechanism that the checkpoint places were obtained according to system reliability.

Dongara and his co workers used from failure protection result and running time homogenous actions for leading the independent task scheduling on non homogenous clusters. There are many sources management and scheduling models in grid calculations that their difference is in their scheduling strategy.

Scheduling strategy based on sureness mechanism:

Scheduling strategy based on sureness mechanism collects the sources information by some of the grid system tools such as MDS and NWS and then returns them. these strategies can upgrade some of items such as cost and task finishing time, they also have a success proportion from task running.

They work in local scheduling better than large scales. The structure of sureness mechanism is difficult is large scales. The other type of scheduling strategy is done by algorithm that in this strategy all of tasks are divided to smaller parts equally and then these parts are allocated to processing sources with a constant proportion. This type of scheduling works well and at non homogenous environments such as grid is suitable but because of calculations and communications of overload the efficiency of system is lower than optimized position. Other scheduling strategies: There are many classic strategies that are used in calculation grids such as PUNCH,Optimal Job Scheduling, Using Legacy Codes,[5]

In all of these strategies are determined the number of sending tasks to any processing source by scheduled module. The variations which are accrued cant effect on scheduled, really there isn't any feedback and these strategies cant scheduled without interruption.

### III. THE NEW ALGORITHM OF MODIFIED RQSG-I

In this algorithm we introduced a suitable method in graph leveling for accurate using with task leveling in proposed algorithm. This leveling of graphs is done in 3 levels.

Zero level: in this level the graphs should create scheduled order using new algorithm for prioritizing of each node. This process leads a more optimized scheduling order rather than previous methods.

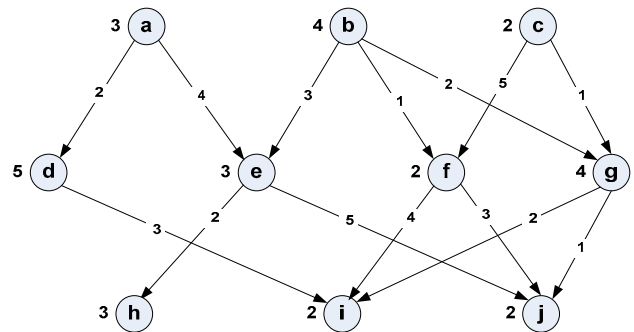


Figure 1 the initial 3 level graph of proposed RQSG-I algorithm

TABLE 1 THE SPECIFICATIONS OF 3 LEVEL GRAPH OF FIG.1 IN OQSG ALGORITHM

Nodes	b	a	c	e	f	g	d	h	i	j
Level	1	1	1	2	2	2	2	3	3	3
Dependency	0	0	0	2	2	2	1	3	6	6
Weight	0	0	0	7	6	3	2	9	20	25
Process	4	3	2	3	2	4	5	3	2	2
Total Process	4	3	2	10	8	10	8	13	22	22

The index parameters of modified RQSG in 3level graphs: 10 nodes are studied according to fig.1 that node B is selected as first priority based on time prioritizing with RQSG algorithm. Then node A is used. In this figure its weighting factors are used accurately so is a better method for creating prioritized functions rather than RQSG algorithm.

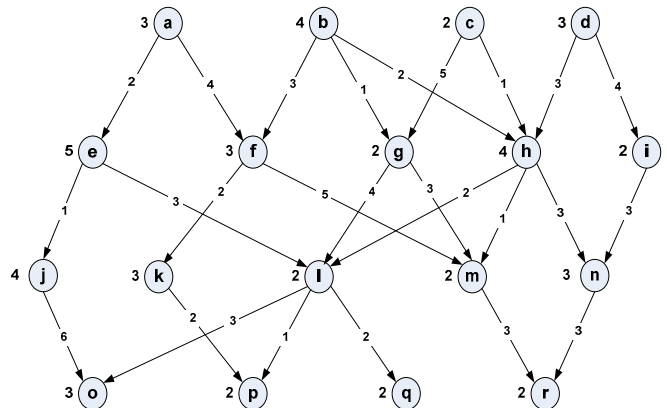


Figure 2: 4 level graph with proposed OQSG algorithm

Now will be defined the proposed algorithm that are created by index parameters

$E_o$  : the number of input vertex

$E_i$ : the number of output vertex

$L$  : level

$R$ : needed processing power amount

$W$ : Amount of sent information between nodes

1)  $E_o$  : the number of input node arrived each node.

In RQSG algorithm are done static tasks that these tasks create a communication between top and bottom layer node according used  $E0$  parameter. This parameter causes to have

an accurate priority choosing rather than tasks and also this parameter can be effective on priority function.

2)  $E_i$  : the number of output vertex from each node. Another index parameter which is used in RQSG algorithm is the number of output vertex that is studied by graph. This parameter determines the release amount of nodes which are finished their tasks and now is allocated a selected task to it in processing source.

3)  $L$ : the node level

Another index parameter is the number of graph leveling . For creating an optimized level is used from mixed function for covering all its states. For node level of  $L$  will create  $2L$  doesn't finished. This priority process will repeat for next levels.

4)  $R$  : the needed processing power value:

One of other index parameter is  $R$  in RQSG algorithm that introduces the needed processing power value for running the processing source. This parameter is used is certification function because of its positive effect on running the tasks in scheduling processing.

5)  $W$  : Amount of sent information between nodes:

Amount of sent information between nodes is an index parameter that is added in proposed algorithm rather than RQSG algorithm for exploring of using amount of width band between processing elements. With this task is revised States for priority function. This parameter causes to the second level of nodes doesn't start to work until the first level qualification function in output result improvement and also the final result from this function will reach to real value

According to above definitions is used this function:

$$\alpha_j = \frac{E_j}{E_1 + 2^L} \quad F_j = R_j * \alpha_j * W_j$$

$j = 1, \dots, n$                        $j = 1, \dots, n$

Prioritizing program code:

Stage 1: we calculate alpha coefficient for each node of graph

Stage 2: after alpha calculation we will calculate P value for each node of graph

Stage 3: we organize P values from high to low

Stage 4: End.

$$\alpha_a = \frac{2}{0 + 2^1} \quad P_a = 3 * 1 * 6 = 18$$

$$\alpha_b = \frac{3}{0 + 2^1} \quad P_b = 4 * \frac{3}{2} * 6 = 36$$

....

b , d , a , c , h , e , f , g , l , j , i , n , k , m , o , p , q , r

In modified RQSG algorithm sending of initial data to processing node and result returning to scheduled is done by message passing method with exploring of delay and width band. For having sureness from arriving of data to processing node and also arriving of result to scheduled is used from a data box ,scheduled will choose another source for task doing if data sending will not be sent in obtained time.

The number of processors can be various at the time. The tasks are dependent and can be processed according to proposed prioritizing by different processors. The transmission time that is divided to task time transmission on processing node and result returning to scheduled and certified time, is introduced in this algorithm. Each processing node that enters into grid environment its width band is calculated rather than scheduled for calculating of transmission time.

When the tasks reach into grid environment are placed in none prioritized tasks queue then this queue is processed as batch type and tasks will be allocated to processing node by proposed algorithm.

#### A. Chromosome presentation

Each chromosome shows a possible solving for scheduler. The chromosomes are shown with a list of parameters

This type of chromosome representation helps us to have the needed expected execution time for doing any task that allocated to a computing node (without considering the transmission time). The first line of table.1 is the identities of computing node that the task is allocated to it. The second line is the mentioned task number and the second one is the expected time for doing the mentioned task on the relative computing node.

#### B. Fitness function

For evaluating the efficiency level of selected computing node, is used from fitness function for doing the purposed task from obtained number. This function involves these parameters:

$N$ : Number of computing node which joining the computati

$NTR_j$ : Number of tasks which processed by computing node with  $ID=j$

$CR_j$ : Computing capability of computing node with  $ID=j$

$WT_i$ : Workload of task  $i$

$W_i$ : Amount of sent information between nodes

$R_j$ : the needed processing power value

$D_i$  : Primary file size of task with  $ID=i$

$RB_j$ : Bandwidth between the scheduler and computing node with  $ID=j$

The expected time for  $WT_i$  task doing on  $R_j$  computing node is calculated from followed equation:

$$(1) \quad T_i = D_i^f$$

$$(2) \quad WT_i = R_j * \alpha_j$$

$$(3) \quad EET_{i,j} = \frac{W_i(R_j * \alpha_j)}{CR_j} + \frac{T_i}{RB_j}$$

$$(4) \quad TR_j = \sum_{i=1}^{NTR_j} EET_{i,j} \quad j=1,2,\dots,n$$

That  $D_i$  is the transmitted data from scheduled to destination processing node,  $RB_j$  is the band with between them and  $TR_j$  is the sum of the  $WT_i$  tasks that are allocated to number  $j$ .

The maximum value of  $TR_j$  shows the makespan and the fitness function is defined as followed equation [2]:

$$(5) \quad fitnessValue = \left( \frac{1}{\max\{TR_j\}_{j=1}^N} \right)$$

The important note is we could introduce a Seri of parameters with their communication costs (going and returning tasks time and certifications) and also we could decrease the failure probability of processing sources according to proposes RQSG algorithm rather than previous ones

C. Procedures of the algorithms

1- Making the primary population: a population is a set of chromosomes in which each of them represents one possible scheduling solution which is the order of mapping between source and tasks.

2- Chromosome evaluation: amount of fitness value for each chromosome will computed the goal in GA is to find chromosome with optimal fitness value.

3- Crossover operation: one pair of chromosome will randomly be selected and chooses a random point in the first chromosome and replaces each other spaces of second part of the tow chromosome with each other. And calculated fitness value for avoiding to generate useless generation come up with the worst fitness value in contrast with its own parents, the function will be repeated. This function will be done in a limited cases.

4- Mutation operation: a chromosome will be chosen randomly and a task of it will be chosen randomly and is given to another node, this function causes that algorithm do not stopped in the local minimum.

5- Finally, the resulted population will be evaluated again. If one of the stopping condition will be done, algorithm will be finished otherwise, we go back to 3rd step. Conditions to end this process will be:

- (1)no improvement in recent evaluations
- (2)all chromosomes converges to the same mapping
- (3)a cost bond is met.

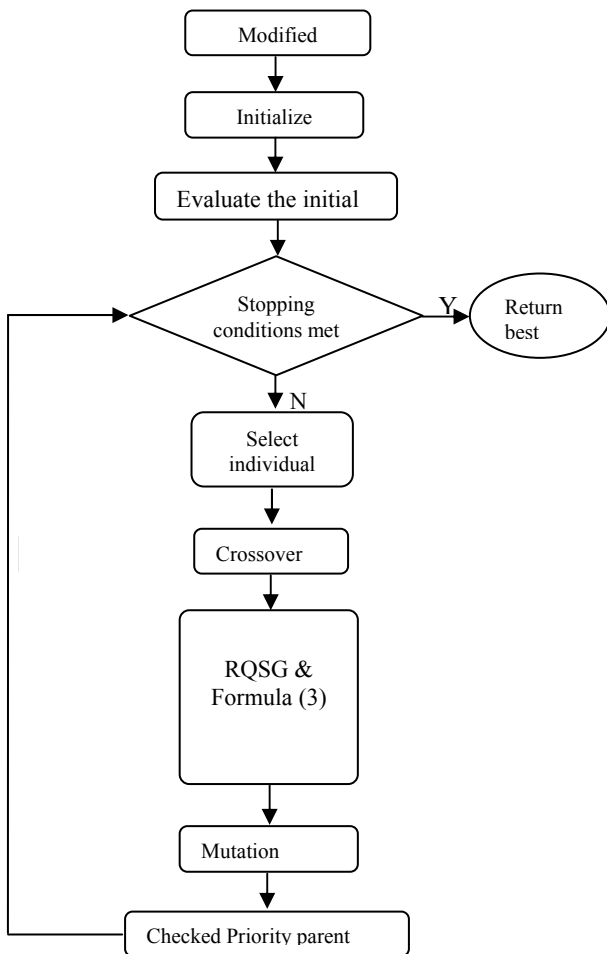


Figure 3: flowchart RQSG-I algorithm

IV. SIMULATION RESULTS

The used tool for simulations is Gridsim. In this paper is tried to select the simulation environment similar to previous studied algorithms as far as possible.

In figure.4 were studied a multipurpose simulation with task number=25 using the index parameter.

The proposed algorithm has had the more optimized the more optimized makespan rather than other two algorithms. Using the index parameters and also priority of static independent tasks by the proposed algorithm has been decreased the scheduling survey rather than the studied algorithms.

TABLE 3 SHOWING THE NUMBER OF TASKS 25 USING AVAILABLE NODES.

25 job	Number of Machines					
	2	3	4	5	6	7
NG	934	875	819	756	679	612
SGA	835	789	736	699	645	590
RQSG	800	760	720	685	632	582
RQSG I	790	754	712	678	625	575

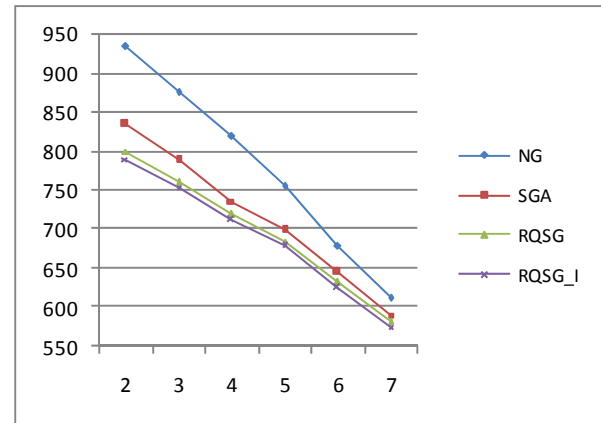


Figure 4: the simulating results from NG, SGA, RQSG, RQSG-I algorithms

Explanation about table 3: this table shows the number of static independent tasks using the studied various nodes.

TABLE 4 SHOWING THE NUMBER OF TASKS 125 USING AVAILABLE NODES.

125 job	Number of Machines					
	2	3	4	5	6	7
NG	4149	3916	3747	3523	3240	3840
SGA	3697	3528	3305	3140	2883	2615
RQSG	3412	3256	3066	2966	2776	2477
RQSG I	3352	3198	2998	2931	2713	2468

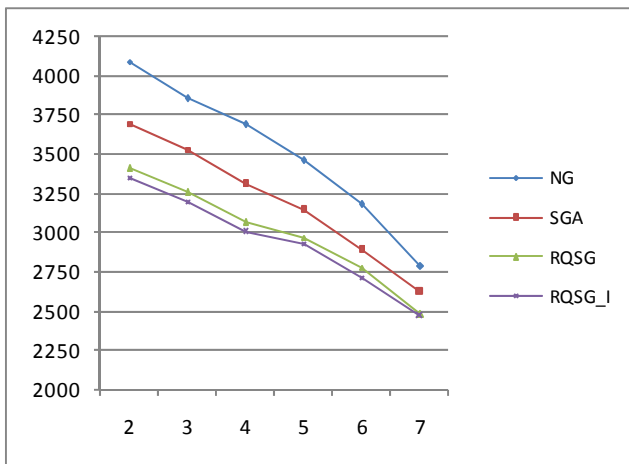


Figure5: the simulating results from NG, SGA, RQSG, RQSG-I algorithms

## V. CONCLUSION

We could create suitable changes in fitness function using various simulations rather than scheduling common algorithm and also it's optimized algorithm. Then we introduced a method with mixing of fitness function and scheduling queue for having more optimization than RQSG algorithm. It has been created the better abilities rather than SGA and NG algorithm that will be decreased the going and returning tasks time and certifications according to results from various simulations.

Finally, we introduced the tasks prioritizing with a technical method that has been increased 1% rather than RQSG algorithm in the worst case and also in the best case has been improved 3%. So the losses time has been an acceptable reduction than similar previous algorithms.

## REFERENCES

- [1] A.Ghorbannia Delavar and Mohsen Khajeh naeini, A new algorithm by index parameters at grid environment with static scheduling, manila, philippines(ICCSM 2010)
- [2] A.Ghorbannia Delavar and V. Aghazarian and S.Sadighi, ERPSD: A New Model for Developing Distributed, Secure, and Dependable Organizational Softwares, CSIT'2009)
- [3] A.Ghorbannia Delavar and M.Nejadkheirallah and M.Motalleb, A New Scheduling Algorithm for Dynamic Task and Fault Tolerant in Heterogeneous Grid Systems Using Genetic Algorithm, IEEE ICCSIT' 2010)
- [4] A.Ghorbannia Delavar and M.Rahmany and M.Nejadkheirallah, GQSD: The new algorithm for optimizing the Quality of service heterogeneous possessing sources at distributed systems, ICSTE 2010)
- [5] N. Fujimoto and K. Hagihara, "Near-Optimal Dynamic Task Scheduling of Independent Coarse-Grained Tasks onto a Computational Grid," Proc. Int'l Conf. Parallel Processing, 2003.
- [6] K. Krauter, R. Buyya, and M. Maheswaran, "A Taxonomy of Grid Resource Management Systems for Distributed Computing," Software-Practice and Experience, vol. 32, no. 2, pp. 135-164, 2005.
- [7] Y.K. Kwok and I. Ahmad, "Link-Constrained Scheduling and Mapping of Tasks and Messages to a Network of Heterogeneous Processors," Cluster Computing, vol. 3, no. 2, pp. 113-124, Sept. 2003
- [8] M. Maheswaran, S. Ali, and H.J. Sigel, "Dynamic Mapping and Scheduling of Independent Tasks onto Heterogeneous Computing Systems," J. Parallel and Distributed Computing, pp. 107-131, 2001.
- [9] S. Hwang and C. Kesselman, "A Flexible Framework for Fault Tolerance in the Grid," J. Grid Computing, vol. 1, no. 3, pp. 251-272, 2003.



**Arash Ghorbannia Delavar** received the MSc and Ph.D. degrees in computer engineering from Sciences and Research University, Tehran, IRAN, in 2002 and 2007. He obtained the top student award in Ph.D. course. He is currently an assistant professor in the Department of Computer Science, Payam Noor University, Tehran, IRAN. He is also the Director of Virtual University and Multimedia Training Department of Payam Noor University in IRAN.

Dr. Arash Ghorbannia Delavar is currently editor of many computer science journals in IRAN. His research interests are in the areas of computer networks, microprocessors, data mining, Information Technology, and E-Learning.