

DGBSA : A BATCH JOB SCHEDULING ALGORITHM WITH GA WITH REGARD TO THE THRESHOLD DETECTOR

ArashGhorbannia Delavar¹, AkramRaei Ezabadi², Vahe Aghazarian³ and Reza Abbasi⁴

^{1,2,4}Department of Computer, Payame Noor University, PO BOX 19395-3697, Tehran, IRAN

a_ghorbannia@pnu.ac.ir , Raeemail.a@gmail.com , Abbasi_Raza@yahoo.com

³Department of Computer, Islamic Azad University, Tehran, Iran
Aghazarian_v@yahoo.com

ABSTRACT

In this paper, we will provide a scheduler on batch jobs with GA regard to the threshold detector. In The algorithm proposed in this paper, we will provide the batch independent jobs with a new technique, so we can optimize the schedule them. To do this, we use a threshold detector then among the selected jobs, processing resources can process batch jobs with priority. Also hierarchy of tasks in each batch, will be determined with using DGBSA algorithm. Now, with the regard to the works done by previous, we can provide an algorithm that by adding specific parameters to fitness function of the previous algorithms, develop a optimum fitness function that in the proposed algorithm has been used. According to assessment done on DGBSA algorithm, in compare with the similar algorithms, it has more performance. The effective parameters that used in the proposed algorithm can reduce the total wasting time in compare with previous algorithms. Also this algorithm can improve the previous problems in batch processing with a new technique.

KEYWORDS

Genetic Algorithm, Batching, Threshold Detector, Distributed System, Grid Computing, Scheduling Independent Task & DGBSA

1. INTRODUCTION

In decade of the twentieth, academic elites have used the batch processing techniques to optimize the total time of processing, to reduce processing time.

A distributed processing system is a collection of processors that are interconnected by a communication network that each processor has its own local memory and peripherals and between each pair processors of system, network communication is done through message passing. Distributed systems are point of view of modern scientific research to achieve increased computational requirements. 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 [3].

Resource allocation is an important problem in distributed systems, and the same is true in grid computing. In this context, the problem is finding resources to execute the tasks that compose an application. Each resource is assumed to be available during a limited time and it is also

assumed that typically there are not enough resources to execute all tasks. Scheduling tasks to the resources is usually made by schedulers that get the information about the available resources from an information service and use this information to choose which resources will use to execute the tasks. This paper proposes a completely different scheduling scheme [2].

Grid systems are kind of distributed systems which in it a “virtual supercomputer” is used involving cluster loose network of computers that act as coordinator. This technology to solve large computational problems, scientific, mathematical and academic computing through the volunteer computing has been and in doing business work in various as drug discovery, economic forecasting, seismic analysis and electronic data processing within an organization to E-commerce support and web services also has been used.

The essence of grid computing is to provide efficient access to resources, maximizing the resource utilization while trying to minimize the job total execution time. The performance of the grid depends strongly on the efficiency of the scheduling. A schedule is an assignment of the tasks of a job to a set of resources. Each job consists of a set of tasks, and each task has to be executed by one of the grid resources for a certain time [2].

We believe that task scheduling is the most important of these issues because inappropriate scheduling of tasks can fail to exploit the true potential of a distributed system and can offset the gains from parallelization due to excessive communication overhead or under-utilization of resources [1]. So we thought about make a schedule that can use distributed system resources effectively and reduce the total run time.

Many heuristic algorithms exist for specific instances of the task scheduling problem, but are inefficient for a more general case [1,7]. Use of Holland’s genetic algorithms [7] (GAs) in scheduling, which apply evolutionary strategies to allow for the fast exploration of the search space of schedules, allows good solutions to be found quickly and for the scheduler to be applied to more general problems [1]. Generic Algorithms, is the technique to search meta heuristic by complex poly-nominal and also while having numerous and very extra alternatives to out problems makes it possible to take the most appropriate and optional choice [4,5]. GA uses the bases of Darvins natural evolutionary selecting method in order to find the optimal answer and to solve the problems with the NP-Complemet complex and scheduling grid resources in which includes in this group ,which is very useful [4,6].

Genetic algorithms are one of the best ways to solve a problem for which little is known. They are a very general algorithm and so will work well in any search space. All you need to know is what you need the solution to be able to do well, and a genetic algorithm will be able to create a high quality solution. Genetic algorithms use the principles of selection and evolution to produce several solutions to a given problem.

Several studies using genetic algorithm to optimize the overall performance of the total run time is done .in this paper with proposed the new algorithm DGBSA with GA, we will provide a new scheduler with regard to threshold detector on batch jobs that with adding new parameters to fitness function of previous algorithm will reduce waste time of processing resources and total run time.

In the section 2 we will investigate the tasks we have done before section 3 explains the new schedule algorithm ; section 4 describes the computer simulation ;section 5 gives concluding remarks.

2. RELATED WORKS

As many researchers have different opinions in the scheduling tasks. The most common real function of the tasks scheduling problem is the makespan function.

However, in the grid, makespan of a non-optimal scheduler might be much longer than the optimal makespan, because the computing power of the network changes over time [11].

To make effective use of the tremendous capabilities of the computational resources distributed within grid environments and minimize the makespan of the grids, efficient task scheduling algorithms are required. Task scheduling algorithms are commonly applied by the grid manager to optimally dispatch tasks to the grid resources. Typically, grid users submit their own tasks to the grid manager to take full advantage of the grid facilities. The grid manager in a computational grid tries to distribute the submitted tasks among the grid resources in such a way that the total response time is minimized. There are several task scheduling algorithms aimed at minimizing the makespan of the distributed systems [15].

The different presented algorithms have been taken into consideration some parameters such as: estimated task execution time, task computational requirement (MFLOP) or communication cost. For example, in heuristic min-min (MM) and max-min (MX) algorithms, that are begun by batch of ready and unmapped tasks, compute the execution time of each task on each processor and add to ETC(i,j) matrix with M*N cells (M is amount of tasks, N is amount of processors). The ETC matrix is directly equivalent to the makespan. Then for each task the processor which will compute it with minimum amount of time will be selected, and added to set. In MX, the tasks are assigned to the processor queue by the largest to smallest completion time. This process is repeated until all tasks are mapped to processors. The MM scheduler is similar to MX except that, after the set of minimum completion time is found, tasks are assigned to the queue in ascending completion time order. In MXC, MMC, LLXC and LLMC algorithms, communication cost is also considered. The makespan is set of $ETC(i,j) + C(i,j)$, where $C(i,j)$ is the estimated communication overhead associated with executing task i on processor j . Each method is suitable for different situations. MX is good when there are more large tasks than small tasks, and MM is opposite of it [12].

Also Many heuristic algorithms exist for specific instances of the task scheduling problem, but are inefficient for a more general case. [8,1] Meta-heuristic search techniques such as GAs [7], tabu [6], and ant colony search [3] are most applicable to the task scheduling problem because we wish to quickly search for a near optimal schedule out of all possible schedules. Good results have resulted from the use of GAs in task scheduling algorithms [1].

A GA is a meta-heuristic search technique which allows for large solution spaces to be partially searched in polynomial time, by applying evolutionary techniques from nature [1,7]. GAs use historical information to exploit the best solutions from previous searches, known as generations, along with random mutations to explore new regions of the solution space. In general a GA repeats three steps (selection, crossover, and random mutations) as shown by the pseudo code in Fig. 1. [1]

```

Initialize population
Do{
Crossover
Random mutation
Selection
} while (stopping conditions not met)
Return best individual
    
```

Figure 1. Pseudo code for genetic algorithm

standard GA(SGA) is too slow to be used in a realistic scheduling due to its time consuming iterations. Scheduling a group of independent tasks on heterogeneous processed resources in grid has been studied by a lot of researchers And obtain the optimal answer [3] But in most of the scheduling algorithms by the support of GA was suggested to grid computing, there are some limitations to general aspects for example homogeneous processing resources, instantaneous message passing, or in these papers and researches, existed communication network without delaying time have been assumed [3,1].Also the Size of works dedicated to the processing resources is not considered. Some research has been done with consider some of these parameters like GRIDTS, GQSD and ESDQGalgorithms .GQSDalgorithm decrease the maximumnumber of repetitions at generations also we will optimize theprocessing sources schedule at offered algorithm rather thanprevious optimization algorithms[4].GQSD algorithm considers delay and network band width. GRIDTSalgorithm , provides fault-tolerant schedulingby combining a set of fault tolerance techniques to copewith crash faults in components of the system. The solutionis mainly based a tuple space, which supports the schedulingand also provides support for the fault tolerance mechanisms. In GRIDTS, use a tuple space for supporting task scheduling. Tuples describing the tasks to be executed are placed by the user in the tuple space. The grid resources retrieve from the tuple spaces tuples that describe tasks that they are able to execute, and execute them. After each execution, the result is placed in the tuple space, becoming available via broker to the user who submitted the tasks to the grid [2].

our research,DGBSA,for scheduling independent tasks in batch and with consider threshold detector as workload dedicated to each processor with regard to power of processing, bandwidth and delay will survey.we will compare result of our research with some researches like SGA,GRIDTS and GQSD.

3. NEW ALGORITHM DGBSA

When tasks arrive to grid environment are placed in a queue of unscheduled tasksand with proposed algorithm, dedicated to processors in batch mode and randomly. For dedicating next batch to each processor, workload of previous batch is checked and if it was bigger than a threshold, for next batch ,the tasks will be selected that their workload are less than threshold. this causes balancing of task on the processors and And prevents them from wasting time and beingunemployed. In view of chromosome using GA, each chromosome from a nationality presents a possible Solution to scheduling. this view , can help us to have expected run time jobs dedicated to each node .

Table 1 , displays sample of data set ,it displays batch jobs dedicated to each processor regard to threshold detector, power of all processor are same , size of each batch is 3 jobs and for all processor is same.

Table 1.sample of data set, batch jobs dedicated to each processor with regard to threshold detector

Number of Node	P1			P2			P3			TD
batch size Bi	50	30	20	40	60	80	70	90	10	
Avg	50			90			85			70
batch size Bi +1 regard to TD	70	90	80	50	30	20	40	10	60	
Avg	80			50			55			65
batch size Bi +2 regard to TD	40	50	35	80	90	70	100	90	89	
Avg	41.5			80			93			67.25

m: Number of processors

$$TD = \frac{\text{Min}_{j=1}^m (\text{Avg}) + \text{Max}_{j=1}^m (\text{Avg})}{2}$$

$$\text{Avg} \begin{cases} \geq TD \text{ then} \\ W_k + Bi + 1 | w_k < TD \\ < TD \text{ then} \\ W_k + Bi + 1 | w_k > TD \end{cases}$$

W_k : workload of job k

Avg_j : average of workload on processor j

$$\text{Avg}_j = \frac{1}{n} \sum_{k=1}^n W_k$$

n: number of jobs given to processor j (size of batch)

3.1.Fitness function

To evaluate the appropriation of selected computational node to execute desired task from the result numbered assessed from the fitness function. This function has these parameters:

m : Number of processors

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

CP_j : Computing capability of computing node with ID=j

W_k : Workload of task i

T_k^f : expected time to transfer job with id=k from scheduled to target node

T_k^o :expected time to receive results job with id=k from target node

CB_j : bandwidth between the scheduler and computing node with ID=j

P :Failure probability of processing resources in environment Grid

$$EET_{k,j} = \frac{W_k}{CP_j} + \frac{T_k^f + T_k^o}{CB_j}$$

$$TC_j = \sum_{k=1}^{TP_j} EET_{k,j} (P + 1) \quad j=1,2,\dots,m$$

TC_j is the sum time of the tasks W_k which excludes the process by the number j. maximum amount of TC_j represents makespanan fitness function will be defined as:

$$\text{Fitness value} = \frac{1}{\max(TC_j)} \quad j \in N$$

3.2.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 calculatedfitness value for avoiding generating 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.

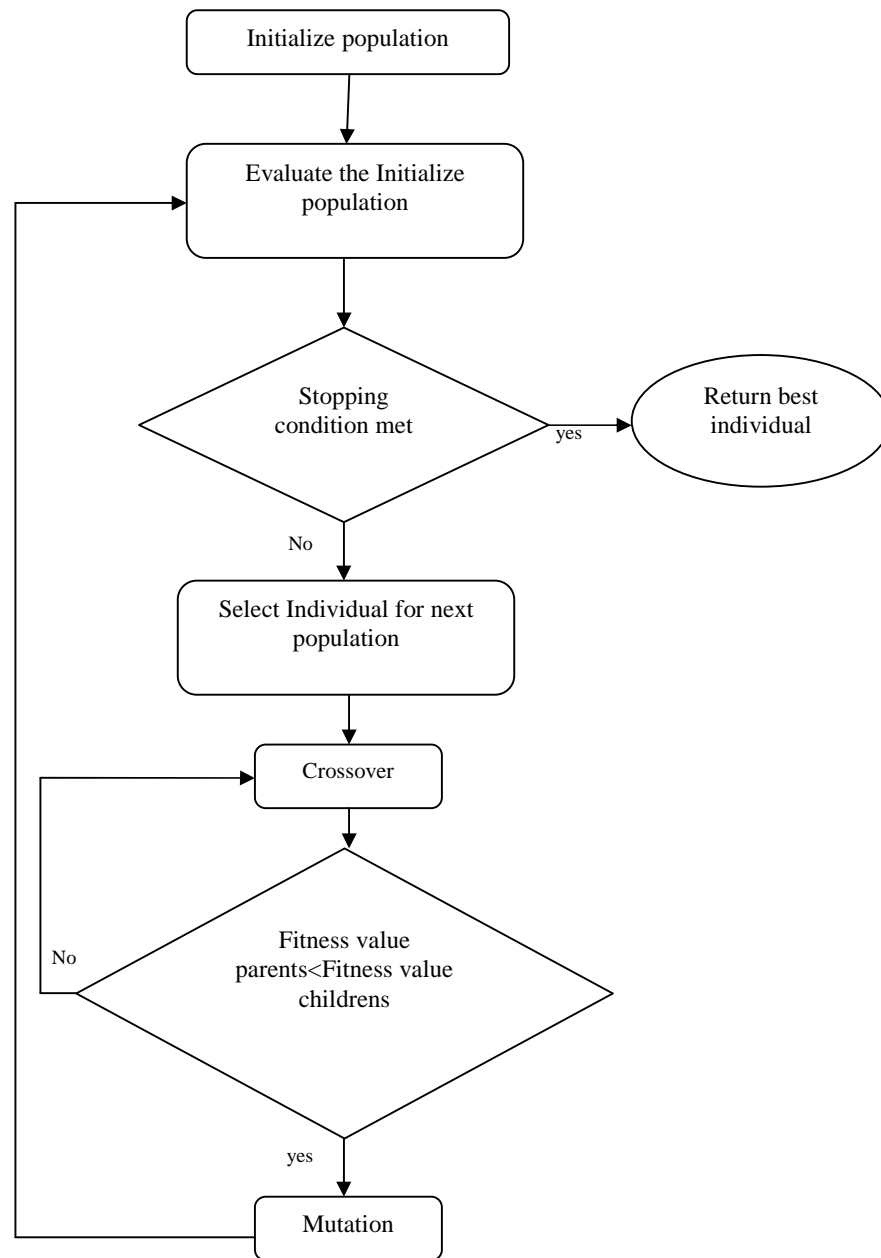


Figure 2. DGBSA algorithm flowchart based standard GA

4.SIMULATION

The used tool for simulations is Gridsim. In this paper I tried to select the simulation environment similar to previous studied algorithms as far as possible. Simulation has been done with 100 and 200 jobs and size of each job is between 4000 to 4500 instruction. Operating power of all nodes is similar.

our research, DGBSA, for scheduling independent tasks in batch and with consider threshold detector as workload dedicated to each processor with regard to power of processing, bandwidth

and delay will survey. We have simulated and compared result of our research with some researches like SGA,GRIDTS and QSD

The proposed algorithm has themore optimized makspan rather than other similar algorithms.Using threshold detectorwith GA forstaticindependent tasks by the proposed algorithm has beendecreased the scheduling survey rather than the studied algorithms.

Figure 3 and 4 show chart of simulation with 100 jobs and figure 5 and 6 show chart of simulation with 200 jobs.As you see in all figures standard GA(SGA) and GQSD have high makespan and low efficiency, others algorithms show better efficient and proposed algorithm (DGBSA) shows better results and higher efficiency.

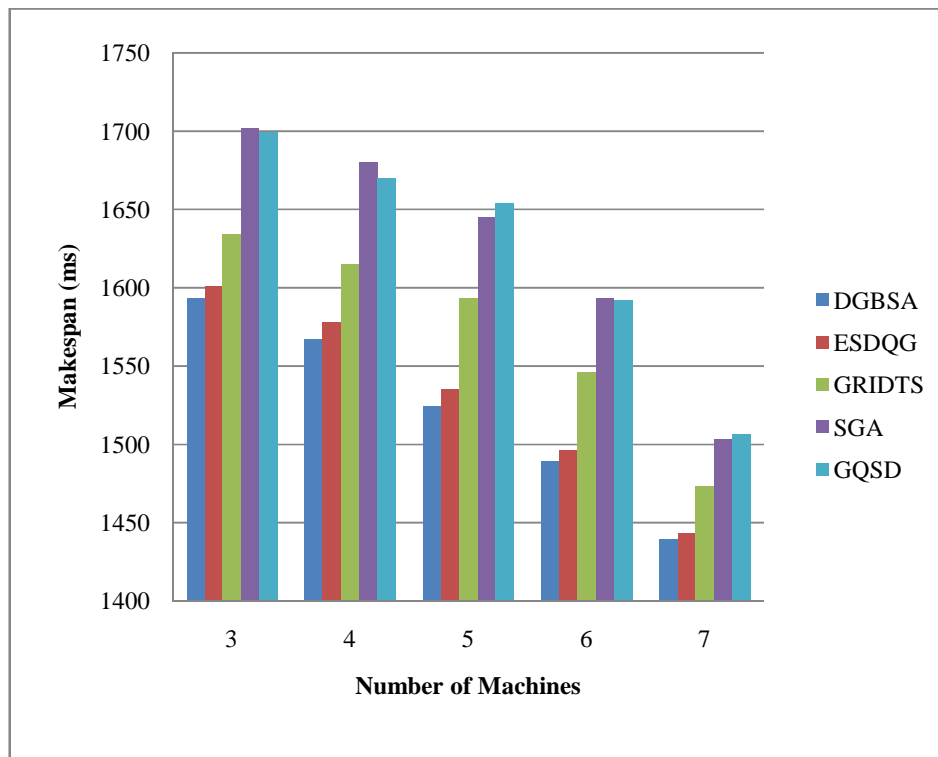


Figure 3.Results of proposed simulated algorithm in contrast to previous algorithms, assumed the 100 jobs

As you see in figure3, our algorithm DGBSA has low makespan and high efficiency. This simulation has been done with 100 jobs and workload of each job is between 4000 to 4500 instruction.Nodes have the same operational power.

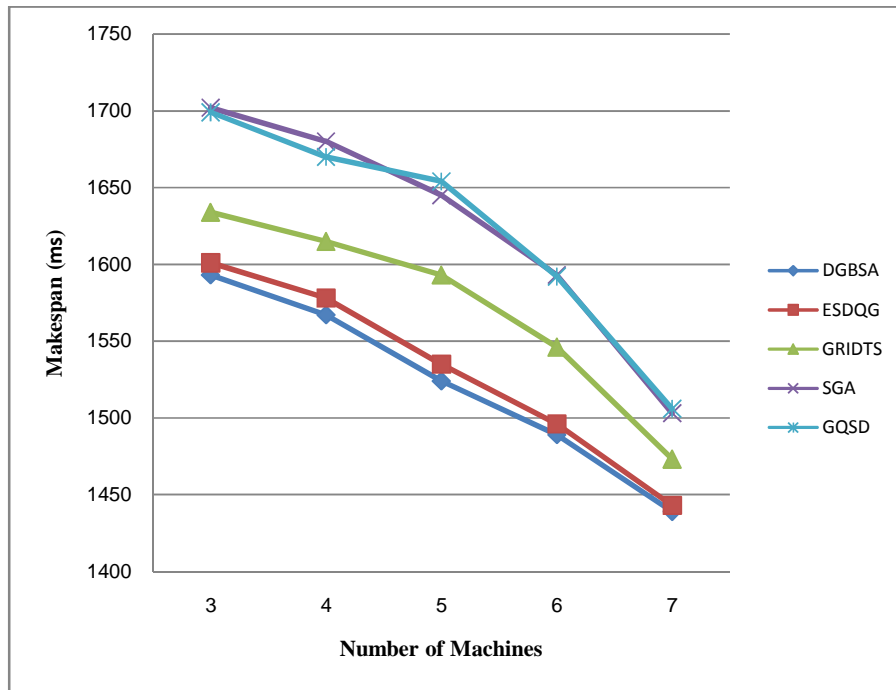


Figure 4. Results of proposed simulated algorithm in contrast to previous algorithms ,assumed the 100 jobs (line chart)

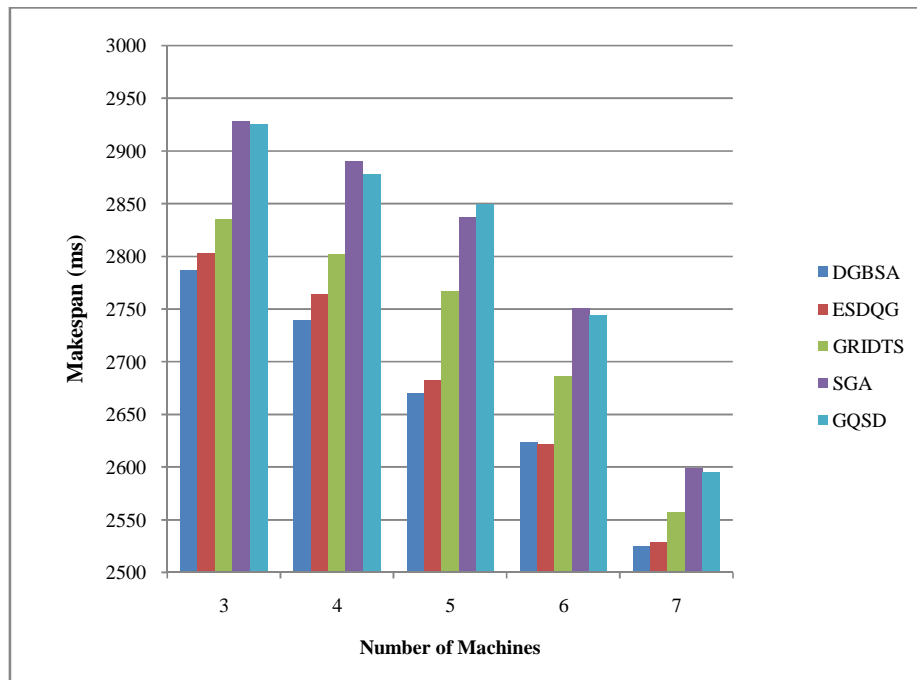


Figure 5. Results of proposed simulated algorithm in contrast to previous algorithms ,assumed the 200 jobs

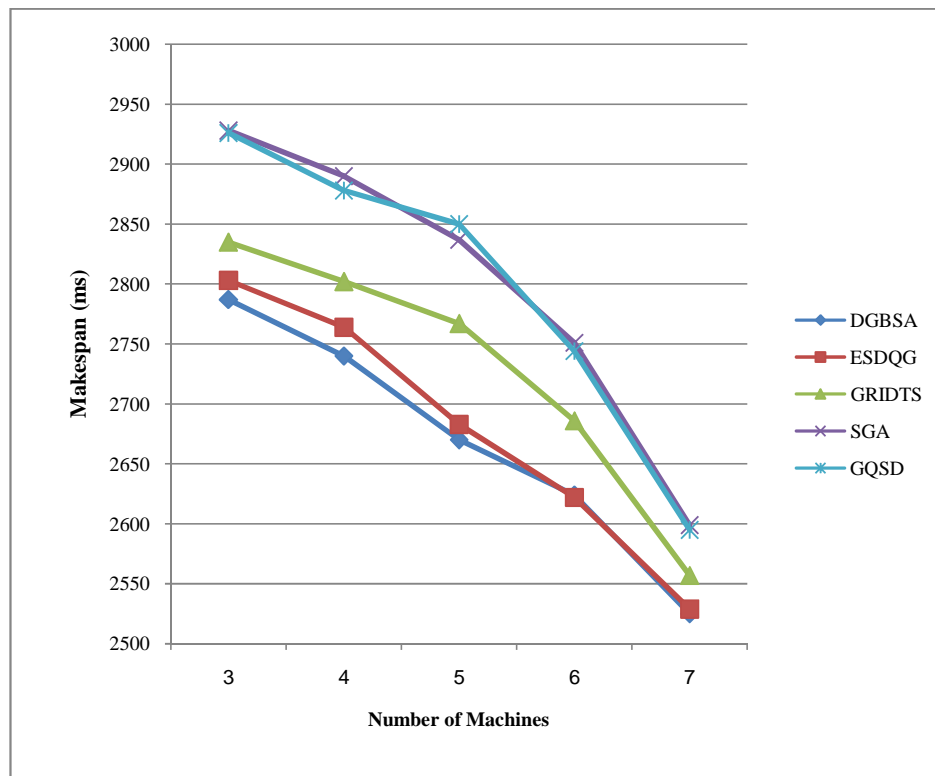


Figure 6. Results of proposed simulated algorithm in contrast to previous algorithms ,assumed the 200 jobs(line chart)

5. CONCLUSION

In this paper, we provided a scheduler on batch jobs with GA regard to the threshold detector. In The algorithm proposed in this paper,we will provide the batch independent jobs with a new technique,so we could optimize the schedule them.To do this, we used a threshold detector then among the selected jobs, processing resources can process batch jobs with priority.

According to new algorithm DGBSA we could calculate the accurate value of going and return time and with comparing its results with other used algorithms the makespan is decreased rather than general genetic algorithm.

In average for 200 jobs,DGBSA Algorithm in compare with ESDQG about 0.4 percent is improved, Also in compare with GRIDTS about 2 percent, in compare with SGA and GQSD about 5 percent is improved. Also for 100 jobs,DGBSA Algorithm in comparewith ESDQG about 6.5 percent is improved, Also in compare with GRIDTS about 3 percent, in compare with SGA and GQSD about 6.2 percent is improved.

REFERENCES

- [1] Andrew J. Page and Thomas J. Naughton, (2005) Dynamic task scheduling using genetic algorithms for heterogeneous distributed computing.
- [2] FábioFavarim , Joni Silva , Fraga Lau , Cheuk Lung , Miguel Correia, (2007) GRIDTS: A New Approach for Fault-Tolerant Scheduling in Grid Computing.
- [3] A. GhorbanniaDelavar, (2010)A new algorithm by index parameters at grid environment with static scheduling.
- [4] A. GhorbanniaDelavar and M. Rahmany and M.Nejadkheirallah, (2010) GQSD: The new algorithm for optimizing the Quality of service heterogeneous possessing sources at distributed systems.
- [5] O. C. H. Liu, A. Abraham and S. Moon, (2006) Variable neighborhood particle swarm optimization for multi-objective flexible job-shop scheduling problems. In The Sixth International Conference on Simulated Evolution And Learning (SEAL06), pp 197–204, China, Springer Verlag, Germany.
- [6] S.BaghavathiPriya and M.Prakash and Dr.K.K.Dhawan, (2007) Fault Tolerance-Genetic Algorithm for Grid Task Scheduling using Check Point.
- [7] J. H. Holland, (1992) Adaptation in Natural and Artificial Systems. MIT Press, Cambridge, MA, USA.
- [8] E. Hou, N. Ansari, and H. Ren,(1994) A genetic algorithm for multiprocessor scheduling. IEEE Transactions on Parallel and Distributed Systems, pp113–120.
- [9] F. Glover. (1986) Future paths for integer programming and links to artificial intelligence. Computers and Operations Research,13:533–549.
- [10] A. Coloni, M. Dorigo, and V. Maniezzo.(1992) Distributed optimization by ant colonies. In Proceedings of the First EuropeanConference on Artificial Life, pp 134–142, Paris,France, Elsevier.
- [11] A.GhorbanniaDelavarand M.Nejadkheirallahand M.Khajehnaeini.(2011) Dynamic Scheduling Reduction Algorithm for DistributedProcessing Resources through Effective Factors Using
- [12] A.GhorbanniaDelavarand Yalda Aryan.(2011)A Synthetic Heuristic Algorithm for Independent Task Scheduling in Cloud Systems.
- [13] A.GhorbanniaDelavar and V. Aghazarian and S.Sadighi.(2009) ERPSD: ANew Model for Developing Distributed, Secure, and DependableOrganizational Softwares, 2009.
- [14] A.GhorbanniaDelavar and Ali Reza KhaliliBoroujeni and JavadBayrampoor.(2011) BPISG: A Batching Heuristic Scheduling Algorithm WithTaking Index Parameters for Mapping Independent Taskson Heterogenous Computing Environment.
- [15] R.Entezari-Maleki, A.Movaghar. (2011) A Genetic Algorithm to Increase the Throughput of the Computational Grids.
- [16] Hai ZhuYuping Wang and Lei Fan and Xiaoli Wang. (2010) Grid Independent Task Scheduling Multi-Objective Optimization Model and GeneticAlgorithm