IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 4, Ver. IV (July – Aug. 2015), PP 69-74
www.iosrjournals.org

A Genetic Algorithm For Scheduling JobsWith Burst Time And Priorities

Shuchi Juyal Bhadula, Bhanvi Rohilla, Bhaskar Nautiyal
Department of Computer Engineering, Graphic Era University, India
Department of Mechanical and Automation Engineering, IGDTUW Delhi, India
Department of Computer Engineering, Graphic Era University, India

Abstract: *Scheduling play extremely important role in our day-to-day life, same as the performance of system is highly affected by the CPU scheduling. For the better scheduling the performance is depend upon the parameter of jobs (arrival time, burst time, priority etc). In this paper we proposed a method for process scheduling using a deadline aware approximation algorithm (Genetic Algorithm) to provide efficient process scheduling, where required schedule is nearby or equal to SJF schedule. We made a comparison among PreemptivePriority algorithm, Preemptive SJF algorithm and GA in terms of number of test case and Total-Turnaround-Time (TAT). The results demonstrate that GA approach produces solutions very close to the optimal SJF algorithm and far better than Priority algorithm.*
Keywords:*CPU Scheduling, Optimization, Genetic Algorithm, Crossover.*

I. Introduction

CPU scheduling is one of the most important key factors of the operating system, the performance and efficiency of CPU is affected by the scheduling of jobs. Earlier we have many scheduling algorithms which work on their specific properties (such as in FCFS the process which come first, execute first; in SJF the process with minimum burst time execute first) which provide sequence of jobs according to it but the scheduling is consider as NP-Complete problem which takes exponential time to compute. We have approximation algorithms which compute it in polynomial time.Genetic algorithm is one of approximation base algorithm which is inspired by nature. It were first developed by Holland[2] and De Jong[3].

A basic GA algorithm consists of five components. These components are as follows: a random number generator, a fitness evaluation unit, the genetic operators Selection (Reproduction), Crossover and Mutation operations. The basic algorithm is summarized below:

1: Initialize Population
2: **repeat**
3: Evaluation
4: Selection
5: Crossover
6: Mutation
7: U**ntil** requirements are met

1.1. Scheduling Criteria

1.1.1. CPU Utilization: The amount of time CPU is busy in performing useful task divide by total amount of time. At 99 % of jobs, 99 % of the time is spent on a particular bunch of lines. To increase utilization the page which consist those lines much always be in main memory.

1.1.2. Throughput: Throughput is the rate at which number of processes are completed their execution per unit time.

1.1.3. Turnaround time: It is a total time when a process is arrive on ready state till the process is complete, its execution and move to terminate state.

1.1.4. Waiting time: Waiting time is the total time spent by process in ready queue.

1.1.5. Response time: Response time is the time when a process is brought into main memory and a time it spent first time in ready queue, that time is contribute in response time.

1.1.6. Tardiness: It is time which is difference of job completion time and its deadline time.

1.1.7. Makespan: When several jobs are together competing of execution, then the starting of first job time to the end of last job run that time is contributing to makespan.

1.2. The following is the problem statement

Let there be N Preemptive processes (jobs) (j_1,j_2,...,j_n) [13] that are waiting to be processed by a single processing system. Each job j_i has its arrival time a_i, burst time b_i and priority p_i respectively.
The objective is to find the schedule that satisfies the following constraints.

1. CPU should process every job.
2. Processes are independent and compete for resources.
3. We have to find a schedule which is nearby or equal to SJF schedule.

II. Related Work

GA is widely used in many engineering fields and its effective, optimization performance was verified in the literature [1]. GA's were first developed by Holland [2] and De Jong [3] and based on mechanics of natural selection in the biological system.

[4] Linet Ozdamar, in this paper the author present resource constrained project scheduling model in which resources are renewable as well as non-renewable. The objective of this paper is minimize the project duration (i.e. makespan) and it is achieved by hybrid GA.

[5] Annie S. Wu and co-authors, in this paper they present an incremental genetic algorithm approach to multiprocessor scheduling in which their approach requires minimal information specific to problem specific operators or repair mechanisms.

[6] S Kamalapur, N Deshpande, in this paper they presents and evaluates a method for process scheduling using genetic algorithm.

[7] vikas singh and co-authors, in this paper they present solution to solve travelling salesman problem using combination of Artificial-Bee-Colony (ABC) and Genetic algorithm (GA) which shows improvement in both precision and computational time.

[8] Zafril Rizal M Azmi and co-authors, in this paper they present a comparison of priority rule scheduling algorithms using different inter arrival time jobs in grid environment in which author provide a justification on the selection of particular priority rule algorithms over another.

[9] Vikas Gaba, Anshu Prashar, in this paper authors present a comparison of processor scheduling algorithms using genetic approach in which aim is to find out the minimum average waiting time over the comparison of First-Come-First-Serve (FCFS) and Shortest-Job-First (SJF) scheduling.

[10] Mendi Neshat and co-authors, in this paper they present a method of adaptive CPU scheduling using Fonseca and Fleming's GA in which they consider three parameters burst time, I/O devices and priority. The proposed algorithm optimizes the average waiting time and response time for the processes.

[11] Rajveer Kaur, Supriya Kinger, in this paper they analysis the job scheduling algorithm in cloud computing in which author compare the existing algorithm for job-scheduling in term of heavy load or traffic.

[12] Devanshu Tiwari, Prof. Damodar Tiwari, in this paper they present hybrid SJF and priority based scheduling of jobs in cloud-computing in which he main focus is on reduce waiting time and response time for the users request which finally leads to reduce the latency and communication overhead.

III. Proposed Work

Earlier we have many scheduling algorithms which work on their specific properties (such as in FCFS the process which come first, execute first; in SJF the process with minimum burst time execute first) which provide sequence of jobs according to it but what if we require a schedule which fulfill or compatible with our deadline time/ required time. Here we have two schedules one according to SJF and the other according to priority, we know that SJF gives us optimal Total Turnaround Time(TAT) but the scheduling is consider as NP-Complete problem which takes exponential time to compute. We have genetic algorithms which compute it in polynomial time.

In our proposed algorithm, We know schedule (sch_sjf) according to SJF and find a new schedule (sch_ga) which is nearby or equal to SJF schedule using genetic algorithm.

Let required time,

RT = sch_sjf (1)

where,

sch_sjf = Total Turnaround Time (TAT) according to Shortest–Job-First(SJF) scheduling.

sch_ga = Total Turnaround Time (TAT) according to Priority algorithm whose priorities are generated using genetic algorithm.

Require Time(RT) is the Total Turnaround Time(TAT) of SJF schedule and we have to find that schedule or a schedule whose TAT is nearby RT.

a. **Fitness Function**
b. Find schedule with minimum tardiness.

Tardiness =modulus [sch_ga – required time (RT)]. (2)
Fitness value = tardiness value.
Required time(RT) in eq(1).

c. Genetic Algorithm

1. Choosing an Encoding scheme.
2. Initialize population

The initial population contain 50 chromosome representing schedule of N jobs, each chromosome contain priorities of N jobs and each priority is unique and priority assign to each job is randomly using poison distribution.

3. Repeat

4. Evaluation

Fitness function which is Tardiness eq(2).

5. Selection
6. Crossover
   Hybrid-Crossover: combination of single point and two point crossover for better converse.
7. Mutation
8. **until** requirements are met

Encoding Scheme

The basis of genetic is chromosome. A chromosome is consist of genes, to represents a gene we required some encoding schemes. Various types of encoding schemes are:

- Binary encoding
- Permutation encoding
- Value/Real encoding
- Octal & Hexadecimal encoding

In our algorithm we use Value encoding scheme.

Crossover Operator Phase

Generate the parent population of food source by applying tournament selection; Select a certain amount of worse chromosomes based on the crossover probability value; for each selected chromosome do Select two parents randomly from the parent population Produce two chromosomes by crossing (hybrid crossover) the selected parents; Apply greedy selection for the selected chromosome and the newly produced chromosome.

Mutation Operator Phase

If mutation criteria is met(it is randomly 3 times out of 100 times) then,Select two individuals based on minimum fitness value from current population for mutation operation. Apply mutate operation (swap operation) to generate new chromosomes (new offspring). Compute the fitness value for that offspring, Replace the worst parent and associated chromosome with new best offspring and its chromosome if it is better update individuals.

IV. Experiment And Result

For the experiment we have randomly generated input set using poison distribution with some factors which are taken for experiment are shown in table 1, number of test cases are 20 with random number of processes respectively. The algorithm is implemented in C with system having 4GB ram, i5 2.6GHz.

Table 1

	MEAN	VARIANCE	RANGE
Process	**50**	**30**	**[20-80]**
Arrival	**5**	**5**	**[0-10]**
Burst	**20**	**19**	**[1-39]**
Priority	**25**	**24**	**[1-49]**

In GA, the number of chromosomes is 50, the mutation probability is 0.03.
The algorithmis tested with different number of cycle (MAXCYCLE): 100, 500, 1000, 5000, 10000.
The crossover probability is 0.2 and the tournament population size is 30.
Number of iteration indicates solution converse after such number of iterations.

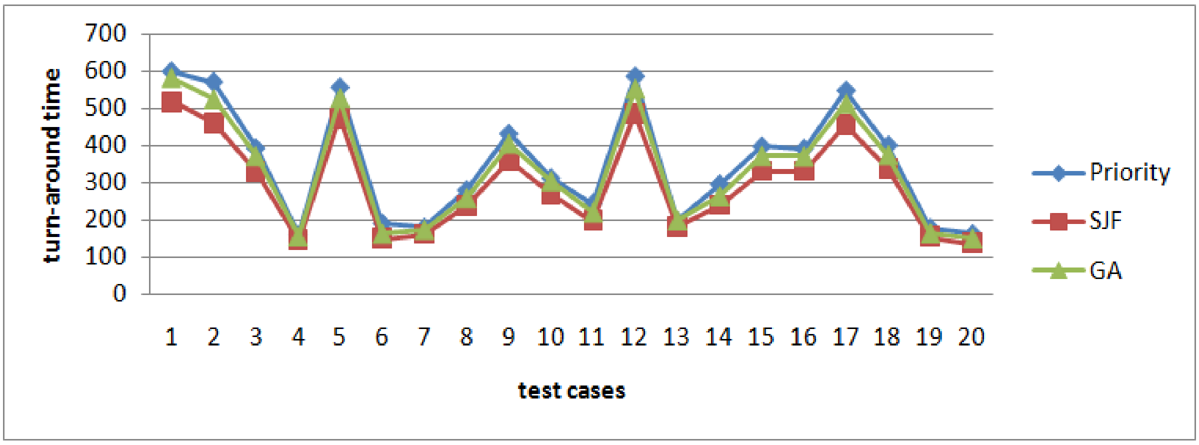

Figure 4.1

In above figure, number of cycle (MAXCYCLE): 100 and graph demonstrate the total turnaround time of all three algorithms with different number of test cases.

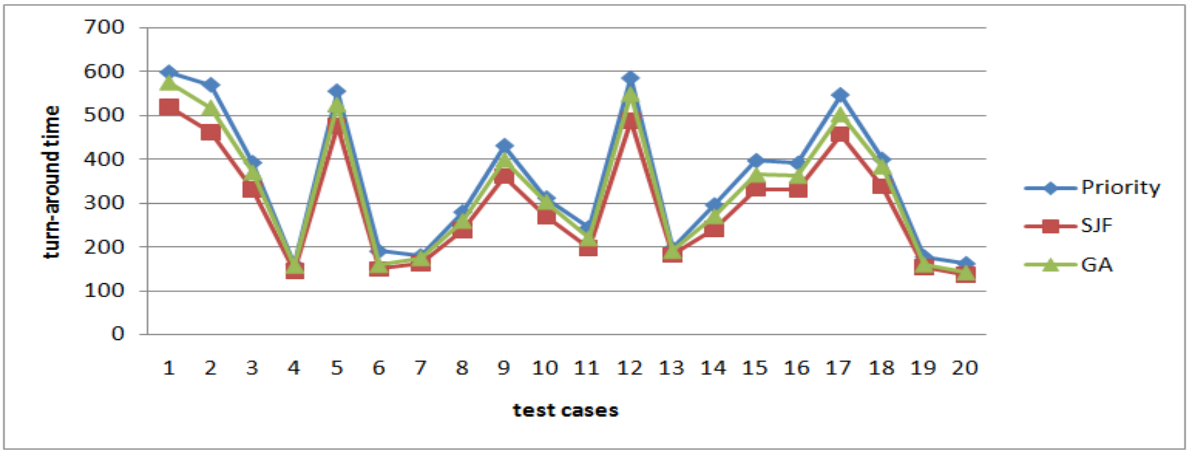

Figure 4.2

In above figure, number of cycle (MAXCYCLE): 500 and graph demonstrate the total turnaround time of all three algorithms with different number of test cases.

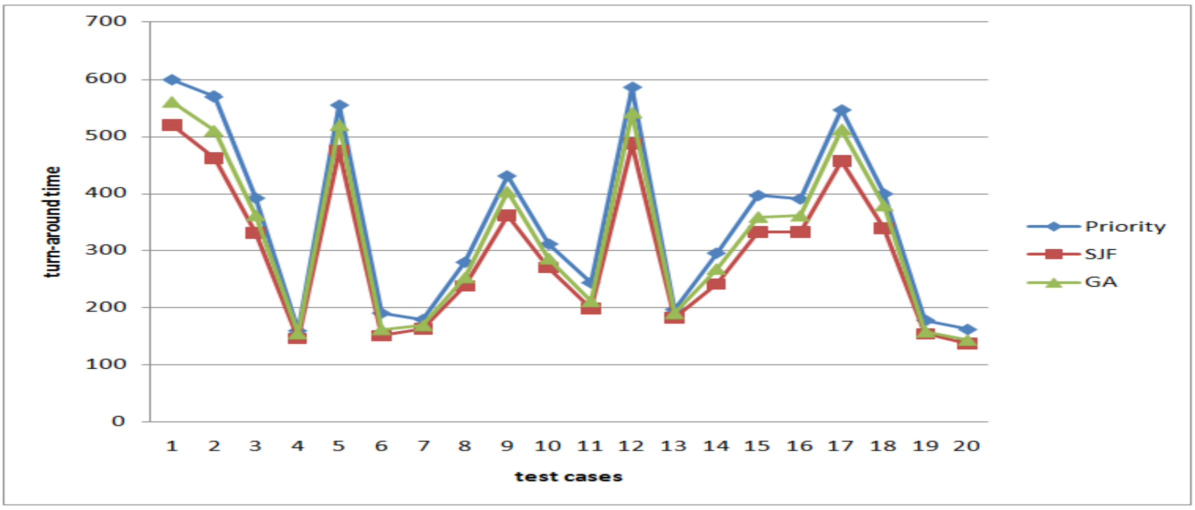

Figure 4.3

In above figure, number of cycle (MAXCYCLE): 1000 and graph demonstrate the total turnaround time of all three algorithms with different number of test cases.

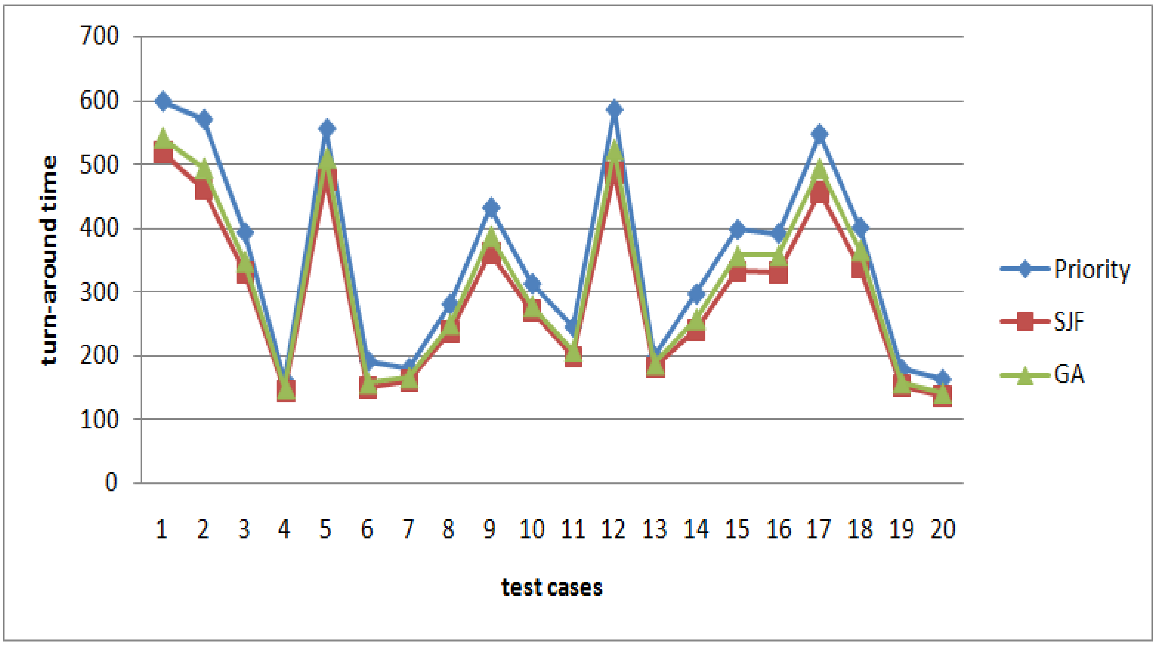

Figure 4.4

In above figure, number of cycle (MAXCYCLE): 5000 and graph demonstrate the total turnaround time of all three algorithms with different number of test cases.

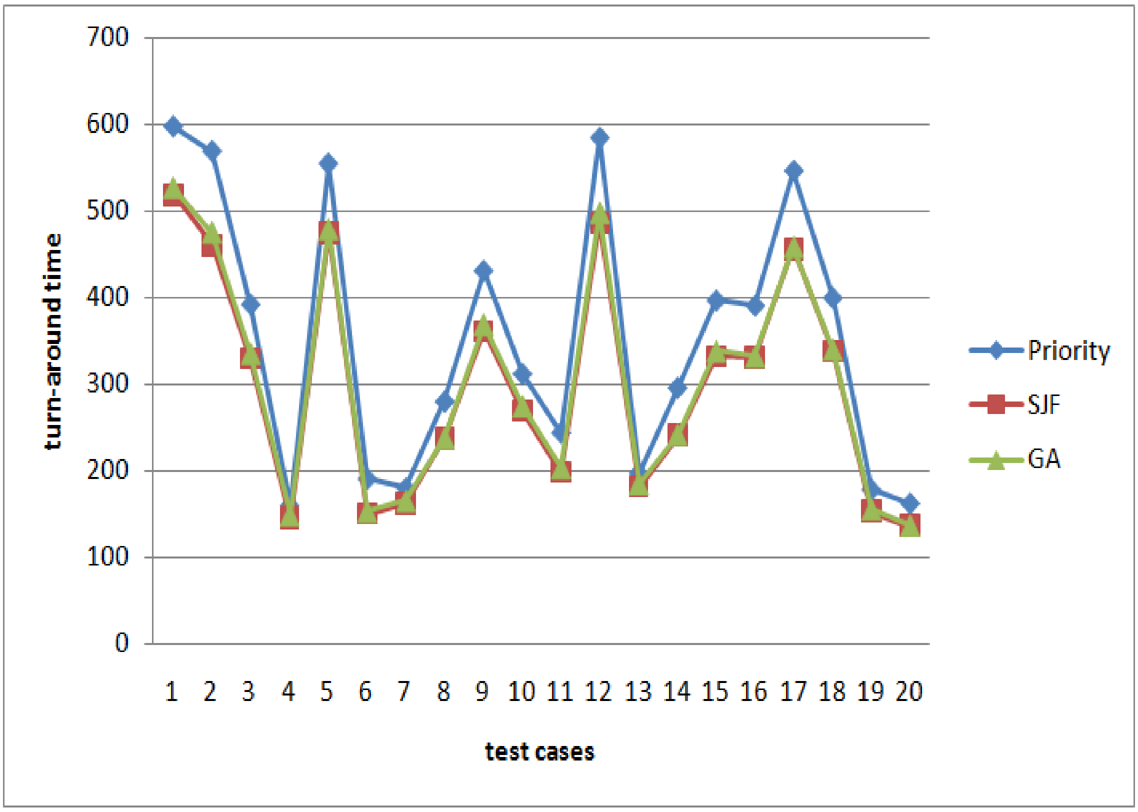

Figure 4.5

In above figure, number of cycle (MAXCYCLE): 10000 and graph demonstrate the total turnaround time of all three algorithms with different number of test cases.

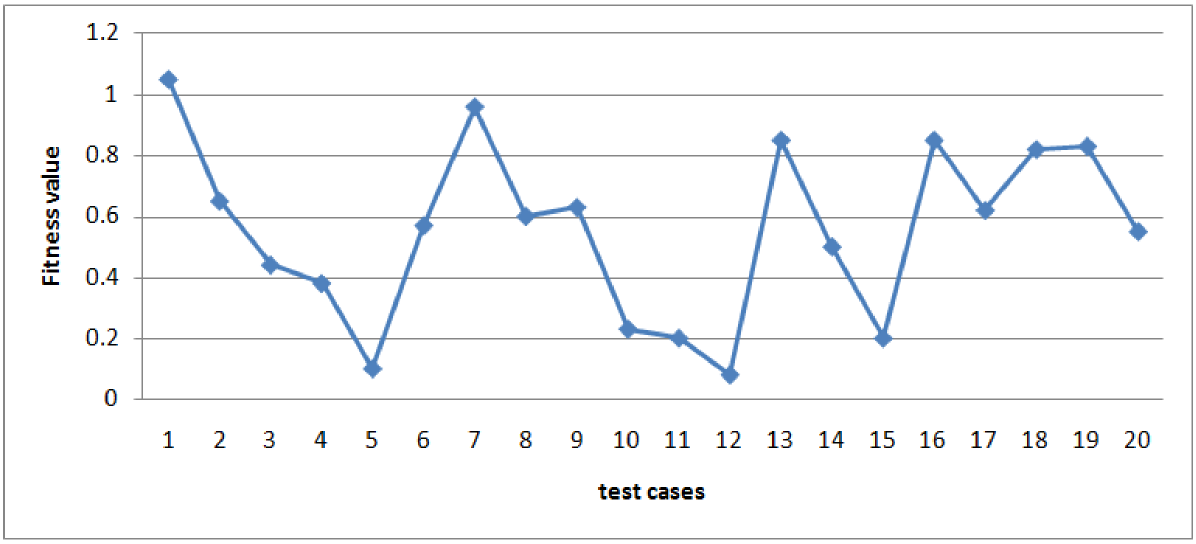

Figure 4.5

In above figure, it shows the fitness value (modulus [sch_ga – required time (RT)]) with different number of test cases.

V. Conclusion

Process scheduling problem are a combinatorial problem in which selection and arrangement of jobs are key factor. Different algorithm and techniques are developed to find above factor, genetic algorithm is one of approximation technique which provide nearby desirable optimal solution. In this paper we proposed a method for process scheduling using a deadline aware approximation algorithm (Genetic Algorithm) to provide efficient process scheduling, where required schedule is nearby or equal to SJF schedule. The results demonstrate that GA approach produces solutions very close to the optimal SJF algorithm and far better than Priority algorithm.

References

[1]. D. E. Goldberg, Genetic Algorithms in Search. Optimization and Machine Learning, Addison Wesley, 1989.

[2]. J. H. Holland, Adaptation in Natural and Artificial Systems. Ann Arbor, MI: Univ. Michigan Press, 1975.

[3]. K. A. De Jong, “Adaptive system design: A genetic approach,” IEEE Trans. Syst., Man, Cybern., vol. SMC-10, pp. 566–574, 1980.

[4]. Linet Ozdamar, “A Genetic Algorithm Approach to a General Category Project Scheduling Problem” IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS, VOL. 29, NO. 1, FEBRUARY 1999.

[5]. Annie S. Wu, Han Yu, Shiyuan Jin, Kuo-Chi Lin, and Guy Schiavone, Member, IEEE, “An Incremental Genetic Algorithm Approach to Multiprocessor Scheduling”, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 15, NO. 9, SEPTEMBER 2004.

[6]. Mrs. Snehal Kamalapur, Mrs. Neeta Deshpande,” Efficient CPU Scheduling: A Genetic Algorithmbased Approach”, 1-4244-0731-1/06/$20.OO ©2006 IEEE.

[7]. Vikas singh, Ritu Tiwari, Deepak Singh, Anupam Shukla, “RGBCA-Genetic Bee Colony Algorithm forTravelling Salesman Problem”, 978-1-4673-0126-8/11/$26.00 c 2011 IEEE.

[8]. Zafril Rizal M Azmi, Kamalrulnizam Abu Bakar, Abdul Hanan Abdullah, Mohd Shahir Shamsir, Wan Nurulsafawati Wan Manan,”Performance Comparison of Priority Rule Scheduling AlgorithmsUsing Different Inter Arrival Time Jobs in Grid Environment” ,International Journal of Grid and Distributed Computing Vol. 4, No. 3, September, 2011.

[9]. Vikas Gaba, Anshu Prashar, “Comparison of processor scheduling algorithms using GeneticApproach” , Volume 2, Issue 8, August 2012.

[10]. MEHDI NESHAT, MEHDI SARGOLZAEI, ADEL NAJARAN, ALI ADELI, “THE NEW METHOD OF ADAPTIVE CPU SCHEDULING USING FONSECA AND FLEMING’S GENETIC ALGORITHM”, Journal of Theoretical and Applied Information Technology 15th March 2012. Vol. 37 No.1.

[11]. Rajveer Kaur, Supriya Kinger, “Analysis of Job Scheduling Algorithms in Cloud Computing”, International Journal of Computer Trends and Technology (IJCTT) – volume 9 number 7 – Mar 2014.

[12]. Devanshu Tiwari, Prof. Damodar Tiwari, “An Efficient Hybrid SJF and Priority based Scheduling of Jobs inCloud Computing”, Volume 2 Issue 4 | December 2014.

[13]. Fatos Xhafa, Ajith Abraham, “Computational models and heuristic methods for Grid scheduling problems”, “Future Generation Computer Systems 26”, 2010, pp.608-621.