Solving Operating System Process Scheduling by Genetic Algorithm Technique

Majed Albogame, Devinder Kaur
Department of Electrical Engineering and Computer Science
College of Engineering, University of Toledo
Toledo, Ohio
majed.albogame@rockets.utoledo.edu, devinder.kaur@utoledo.edu

***Abstract*—The scheduling of operating system process is considered as a hard problem. In this paper, the Genetic Algorithm technique can implement the powerful and efficient scheduling process in the operating system. Accordingly, Genetic Algorithm improves the performance and throughput of the whole system. The results are shown that the Genetic Algorithm is always gives the optimal solution of CPU scheduling process. As a result, Genetic Algorithms plays a critical role in achieving the best short time of CPU.**

***Keywords*—Genetic Algorithm, CPU- Sequencing and Scheduling.**

I. INTRODUCTION

System performance in the operating system can be determined by its scheduling process to be efficient or not [1]. Typically process scheduling in an operating system is hard decision and non-solving problem. The throughput and efficiency of the system will be maximized depending on the average waiting time [2]. If scheduling problem get solved, the overall system will be more flexible and robust.

Genetic algorithm, developed by John Holland in 1975 [3], is a search technique widely applied in optimization problems, with a main objective of evolving a series of solutions and deriving the best of them through some genetic operations. Simply, the technique creates a random population of solutions, and then applies genetic operators such as mutation and crossover to develop the next generation of solutions.

GAs are a potential solution to this problem which can be used to evaluate scheduling process of CPU. The offspring can be produced by using single point crossover between two parents or "chromosomes". The crossover is important factor in GA algorithm aims to improve the population fitness from generation to generation [4].

The mutation is another operation in GA algorithm. Also, it used between offspring chromosomes which are produced by crossover operation. The mutation goal to change an offspring chromosomes to result a more optimal solution in the search range. Experientially, crossover rates can be in the range of 60-80% to satisfy the best solutions. Also, the mutation rates can be in the range of 1-10% produce good results [6]. Figure 1 illustrates GA flowchart and its significant parameters.

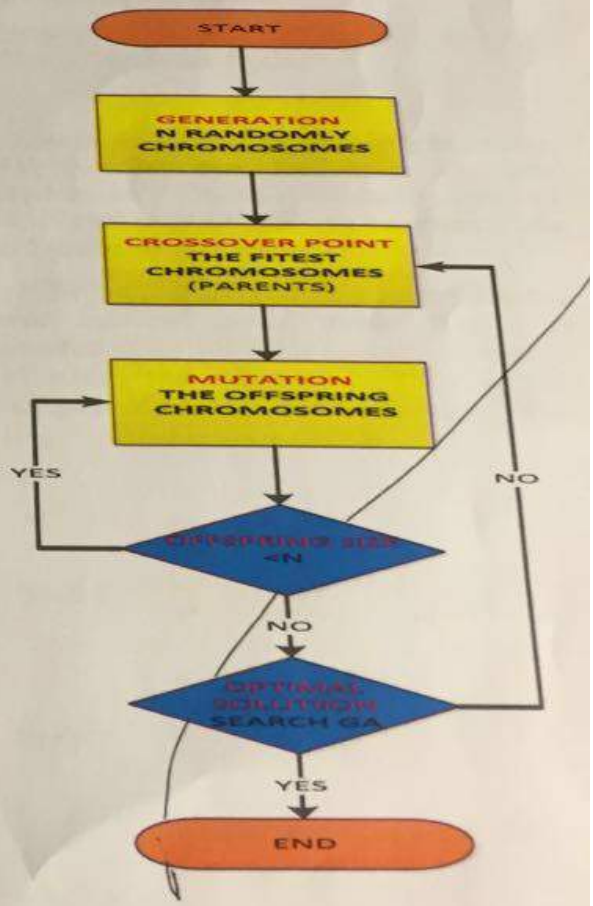

Figure 1. : Flowchart describing GA algorithm.

The rest of the paper is organized as follows. Section II describes the experimental setup. Section III presents the results of the experiment. Finally, in Section IV, conclusions are presented.

II. EXPERIMENTAL SETUP

This section will explain the procedure used to determine the optimal processes sequence [8]. In a single CPU at a time, there are N processes (1, 2, and 3….N) and these processes have their static burst time. The optimal processes sequence is defined as total waiting time of processes should be minimum in CPU [9].

However, there are different algorithms which can use to control CPU scheduling processes such as First Come First Serve (FCFS) algorithm, Sort Job First (SJF) algorithm, and Round Robin (RR) algorithm. But, the Genetic Algorithm can give the optimal solution comparing with them.

Genetic algorithm, a powerful research technique, which nowadays is classified as a kind of evolutionary computation method. In this paper, Genetic algorithm is used to choose the fittest individuals after as parents which are selected by Roulette wheel setting. And to achieve minimum waiting time the fitness function defined here is based on first come first serve algorithm.

The parameters and strategies for Genetic Algorithm in table 1 are used to schedule the processes in CPU.

TABLE I. PARAMETERS USED GA AND CPU SCHEDULING

Parameter / Strategy	Setting
Population size	10
Population Type	Generational
Initialization	Random
Selection	Roulette wheel
Crossover	Two Parents, Modified crossover
Bit Swap Inversion Probability	0.1
Replacement strategy	80 % Best
Stopping Strategy	50 generations
No. of process to be Schedule	5
CPU Scheduling Criteria	Minimum Average Waiting Time

Here is the steps that are used to get the results from the enetic Algorithm depending on the conditions in Table 1 ove:

- Generate random population using Evop (Popcurrent) function.
- Evaluate each individual using fitness function Fit(Popcurrent) by using this formula:

$$Fitness(S_i) = \sum_{i=1}^{N} \frac{Wt_i}{N} \quad (i = 1,\ldots N) \qquad (1)$$

- Use Pickchrom (Popcurrent) function for mating as follows:

$$(P) = \frac{F(S_i)}{\sum_{j=1}^{POPSIZE} F(S_j)} \qquad (2)$$

Where F (S_i) is the fitness of chromosome S_i

- Recombine the two individuals by apply Modified crossover Mcrossover (Popcurrent).
- Make inversion operator Inversion (Popcurrent) to give offspring chromosomes.
- Compute the fitness of offspring and insert the offspring into the new Population.

Actually, there are 5 jobs which are to be consider with their burst time. The number of possible sequences are 5! The total 10 sequence is selected out of 120 for the 5 jobs. Considering the number of jobs as 5 and the crossover point is 2.

In the modified crossover we get proper sequence order individual (parent) because there is no repetition of any job in this sequence. For instance, let assume two individuals (parents) which are marked as fit and use for the next generation [10] and [11].

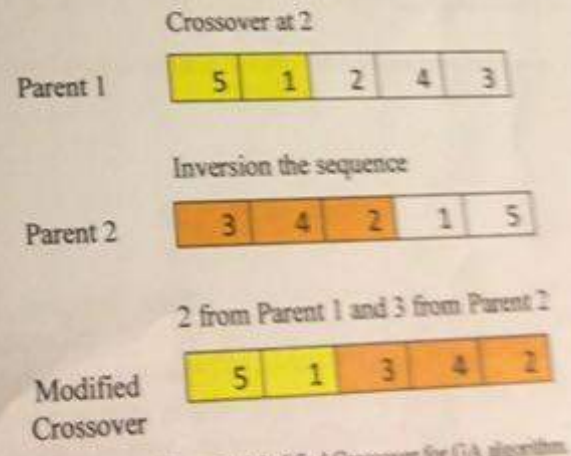

Figure 2: Modified Crossover for GA algorithm

III. RESULTS AND ANALYSIS

In this section the results of the Genetic Algorithm shown in the Table 2. There are four jobs (1, 2, 3, 4, and 5) with their burst time. Also, the twenty sequenced chromosomes are examined by apply the genetic operator's crossover, fitness function and inversion. Moreover, the results with GA are comprised with different technique which are FCFS, SJF, and RR as shown in figure 3 and figure 4.

TABLE II. COMPRISION OF RESULTS

Sr. No.	Burst Time of Process					GA	FCFS	SJF	RR
	P1	P2	P3	P4	P5				
1	5	15	12	25	5	14.8	22.8	14.8	28.4
2	10	50	25	4	8	17	48.8	17	33.4
3	2	10	15	12	4	10.4	16	10.4	19.6
4	4	2	25	3	23	9.6	15	9.6	18
5	8	11	35	4	30	18.4	27.8	18.4	35.6
6	20	25	40	8	36	35.6	48.6	35.6	70
7	22	40	35	2	38	36.4	56	36.4	71.8
8	35	30	36	3	39	41.6	61	41.6	82
9	36	10	20	4	20	21.2	43.6	21.2	41.6
10	2	25	28	5	32	20.2	28.8	20.2	38.8
11	8	2	15	6	35	11.4	14.8	11.4	21.4
12	9	4	17	3	50	11.8	17	11.8	22.4
13	15	8	19	2	40	16.2	24.8	16.2	31.2
14	20	2	20	1	35	14	25.4	14	26.8
15	21	2	21	4	32	16.6	27.2	16.6	31.8
16	16	2	15	8	15	15.4	21.6	15.4	29.8
17	17	2	26	10	16	17.4	27.2	17.4	33.8
18	18	3	18	15	18	22.2	26.4	22.2	43
19	19	4	19	2	19	15.4	26.6	15.4	29.6
20	20	6	20	3	3	10.6	28.2	10.6	20.6
Total Avg. Waiting Time						376.2	607.6	376.2	729.6
Total Mean Avg. Waiting Time						18.81	30.38	18.81	36.48

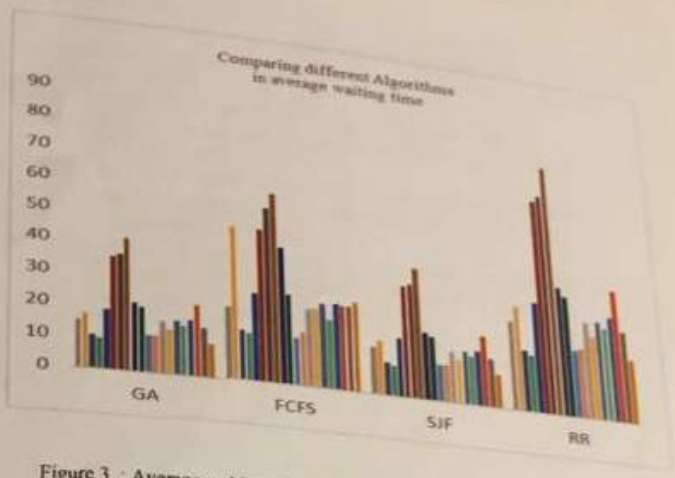

Figure 3. : Average waiting time versus different algorithms in bar form.

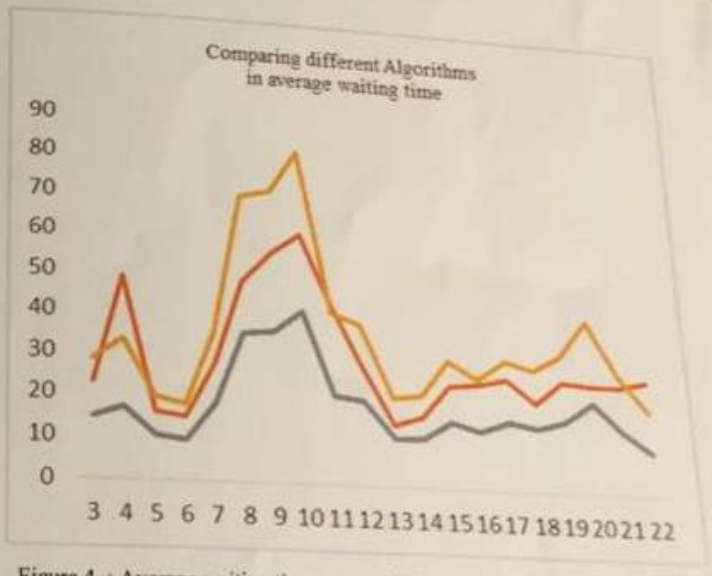

Figure 4. : Average waiting time versus different algorithms in continue form.

Figure 3 and figure 4 focus on the comparison based on the minimum average waiting time for GA, FCFS, SJF, and RR algorithms.The results are shown that the Genetic Algorithm can achieved the less average waiting time and grants the optimal solutions.

IV. CONCLUSIONS

The program results clearly show that the Genetic Algorithms is able to get the highest performance and throughput in operating system process scheduling. Also, the Genetic Algorithm proved that the CPU scheduling process can be solved as easy, optimal and flexible way.

REFERENCES

[1] M.Nikravan,M.H. Kashani,"A Genetic algorithm for process scheduling in distributed operating systems considering load balancing", Proceedings 21st European Conference on Modelling and Simulation Ivan Zelinka, Zuzana Oplatková, Alessandra Orsoni ©ECMS 2007

[2] Blazewicz, J., Domschke, W., and Pesch, E. (1996). The job shop-scheduling problem: Conventional and new solution techniques. European Journal of Operational Research, 93:1-30.

[3] Holland, J.H., 1975. "Adaptations in natural and artificial systems", *Ann Arbor: The Universit0 of Michigan Press.*

[4] David E.Goldberg, Genetic Algorithms in Search Optimization & Machine learning, Second Reprint, Pearson Education Asia pte. Ltd., 2000.

[5] L. N. de Castro, *Fundamentals of Natural Computing*. 2006, pp. 61-88.

[6] A. D. Ali, I. M. M. El Emary, and M. M. A. El-Kareem, "Application of Genetic Algorithm in Solving Linear Equation Systems," *MASAUM Journal of Basic and Applied Sciences (MJBAS)*, vol. 1, no. 2, pp. 179-185,2009.

[7] I. M. Abiodun, L. N. Olawale, and A. P. Adebowale, "The Effectiveness of Genetic Algorithm in Solving Simultaneous Equations," *International Journal of Computer Applications*, vol. 8, no. 8,pp.38-41,2011.

[8] S. Ashour. *Sequencing Theory*. Springer-Verlag, New York, 1972.

[9] K. R. Baker. *Introduction to Sequencing and Scheduling*. John Wiley andSons, Inc., New York, 1974.

[10] L.M.Schmitt, "Fundamental Study Theory of Genetic Algorithms" , International Journal of Modelling and Simulation Theoretical Computer Science 259, 2001, 1 – 61.

[11] L. Davis, "Applying Adaptive Algorithms to Epistactic Domains", in Proceedings of the Int. Joint Conf. on Artificial Intelligence (IJCAI'85), Los Angeles, CA, pp. 162-164.