MNK Publication

International Journal of Latest Research in Science and Technology **ISSN (Online):2278-5299**
Volume 1,Issue 1 : Page No.55 - 59 , May –June 2012
https://www.mnkpublication.com/journal/ijlrst/index.php

EFFICIENT GENETIC OPERATORS BASED ON PERMUTATION ENCODING UNDER OSPSP

[1]Rajiv Kumar, [2]Dr.Pardeep Goel
[1]Research Scholar, Department of Computer Science & Information Technology.
Singhania UniversityJhunjhunu,Rajasthan, India
[2]Associate Professor,M.M. (P.G.) College,Fatehabad

Abstract – Present study in the paper is concerned with the development of new genetic operators to optimize the performance of the system. To improve the production facilities, a set of jobs are executed on the set of machines. For better performance there are large numbers of constraints. Process scheduling theory has been developed to meet all side constraints. Process Schedule is done in such a way that the resulting solution minimizes the given objective function. Many variants of the basic scheduling problem can be formulated by differentiating between machine environments, side constraints and objective functions. Genetic algorithm have been applied to OSPSP. The study shows that proposed operators shows better results

Keywords- Genetic Algorithm,OSPSP,Operators,Scheduling.

1. INTRODUCTION

Process scheduling in an operating system can be stated as, ***"Selection of Process for Execution and allocation of the CPU to the selected process"***. Scheduling is considered as the fundamental function of OS. Efficient scheduling always increases the performance of the operating system. Scheduling is important because it influences user service and efficiency of processor. Therefore, the throughput and efficiency of the OS increases by use of efficient process scheduling.

2. SCHEDULING PROBLEMS

In OS the problem is to allocate available resources to the given process within desirable time period to perform a set of works. Realistic scheduling problem is very difficult to solve in OS. This is because it has many constraints. Some of the constraints cannot be defined mathematically.

In general scheduling problems can be production and computational fields. These are most important fields as related to scheduling. Different tasks individually compete for resources like manpower, money, processors (machines), energy and tools. These resources very differently in nature. Moreover, a organize set of tasks, reflecting relations among them can be defined in several ways. In addition, various criteria which measure the quality of the performance of a set of tasks can be taken into account.

[1] They characterized scheduling problems by three sets:

(i) set $\mathcal{J} = \{J1, J2, J3 \ldots, Jn\}$ *of n jobs,*

(ii) set $\mathcal{P}$ = {P1, *P2,….,Pm } of m processors {machines)* and

(iii) set $\mathcal{R}$ = *{R1, R2… Rs} of s* types of *additional resources* $\mathcal{R}$,

Process *Scheduling* means to assign *Processors (P)* and *Resources*($\mathcal{R}$) to *Tasks* ($\mathcal{T}$) in order to complete all tasks under the imposed constraints.

3. TYPES OF OPERATING SYSTEM SCHEDULERS

Operating systems generally have three types of distinct scheduler, ***"Long Term Scheduler"***, **"*Medium Term Scheduler"*** and ***"Short Term Scheduler"***. These are based on the relative frequency with which these functions are performed. In OS the Scheduler module is responsible for selecting the next jobs to be admitted into the system and the next process to run.

3.1 Long Term Scheduler:

[2] The long term or admission scheduler decides which jobs or processes are to be admitted to the ready queue in the Main Memory It means that, when an attempt is made to execute a program, its admission to the set of currently executing processes is either authorized or delayed by the long term scheduler. Thus, this scheduler dictates what processes are to run on a system, In current operating systems, this is used to make sure that real time system CPU have more time to finish the particular task. So there is requirement for efficient real time scheduling for the modern system with GUI interface. Basically the queue is resided in the hard disk or in the virtual memory. Long term scheduling is also important in large scale systems like batch processing systems, computer clusters, and supercomputers and renders farms. In these regard dedicated job scheduler is required for this function

3.2 Medium Term Scheduler:

[2], The medium term scheduler removes processes from main memory. It places them on secondary memory like disk drive or other source of secondary memory or vice versa. In this process swapping of processes is takes place. The

Publication History

Manuscript Received	:	*5 June 2012*
Manuscript Accepted	:	*25 June 2012*
Revision Received	:	*28 June 2012*
Manuscript Published	:	*30 June 2012*

medium term scheduler decides to swap out a process which has not been active for some time, or a process which has a low priority or a process which is taking up a huge amount of conventional memory in order to free up main memory, so that it can used for other processes.

3.3 Short Term Scheduler:

[2],*CPU scheduler* is the short term scheduler, It decides which of the ready processes are to be executed next following a clock interrupt, an I/O interrupt, an operating system call or another form of signal. The short term scheduler takes scheduling decision very quickly as compared to the long term scheduler. This scheduler can be preemptive. It implies that it is capable to remove processes forcibly from a CPU when it decides to allocate that CPU to another process, or non preemptive i.e when scheduler are not able to take off the control of CPU from the currently executed process.

3.4 IMPORTANCE OF SCHEDULING

Scheduling is a decision making practice. It is used in different types of operating system. Its aim is to optimize one or more objectives with the allocation of resources to tasks over given time periods. The resources and tasks in an organization can take a lot of different forms. The resources may be machines in a workshop, crews at a construction site, processing units in a computing environment, and runways at an airport and so on. The tasks may be operations in a production process, take offs and landings at an airport, executions of computer programs, stages in a construction project. Each task can have a definite priority level, an earliest likely starting time and a due date. The objectives can also take many different forms and one objective may be the minimization of the completion time of the last job and another may be the minimization of the number of jobs completed after their respective due dates. Scheduling plays an important role in most manufacturing and service systems as well as in most information processing environments.

Scheduling derives its importance from the two different considerations:

(i) Ineffective scheduling results in deprived utilization of available resources. A noticeable

symptom is the idleness of facilities, human resources and apparatus waiting for orders to be

processed. As a result of this cost of production increases.

(ii) Poor scheduling in the OS will always result in the increases of Average waiting time,

Turnaround Time and poor service response time. With ineffective scheduling the

Performance of the OS always reduced.

4. GENETIC ALGORITHM

Genetic Algorithms (GAs) are adaptive methods which may be used to solve search and optimization problems. They are based on the genetic processes of biological organisms. Over many generations, natural populations evolve according to the principles of natural selection and survival of the Fittest", First clearly stated by Charles Darwin in The Origin of Species. By mimicking this process, genetic algorithms are able to evolve" solutions to real world problems, if they have been suitably encoded. For example, GAs can be used to design bridge structures, for maximum strength/weight ratio, or to determine the least wasteful layout for cutting shapes from cloth. They can also be used for online process control, such as in a chemical plant, or load balancing on a multi-processor computer system. The basic principles of GAs were first laid down rigorously by Holland [3], and are well described in many texts (e.g. [4], [5],[6]). GAs simulates those processes in natural populations which are essential to evolution. Exactly which biological processes are essential for evolution, and which processes have little or no role to play is still a matter for research.

Due to the adaptability of the GA it can be used to solve complex problem. Scheduling problem is considered as the NP- Hard problem. GA is suitable to apply on scheduling problem for find out optimal solution.

5. METHODOLOGY FOR OSPSP

In the present work GA have been proposed as a Long Term Scheduler (LTS).GA is a robust search technique. It has the capability to cope with problem on which it is applied. So GA can enhance the performance of the OS. Survey of literature shows that GA has been applied to scheduling problem of different domains. They have applied GA to the scheduling problem with prevailing data mining of parameter setting. The present data mining is used to the optimal parameter setting of GA under the given scheduling problem. Thus, it is proposed to use parameter setting of GA under the permutation encoding. The best parameter setting under OSPSP with permutation encoding has given better results.

During the research work, it was found that permutation encoding to represent the candidate solution gives optimal parameter setting for OSPSP. Thus, there was a need to analyze the parameter setting of GA based on permutation encoding under the scheduling problems. To develop new algorithm, it became essential to analyze the parameter setting for permutation encoding.

From the survey of literature, it was observed that permutation encoded genetic operators are very limited. There is a need to design new genetic operator. The new genetic operator should be based on permutation encoding. Research was carried out to find new genetic operators. The purposed genetic operators have enhanced the performance of the process scheduling. The proposed BQPSGA is shown in fig.1.

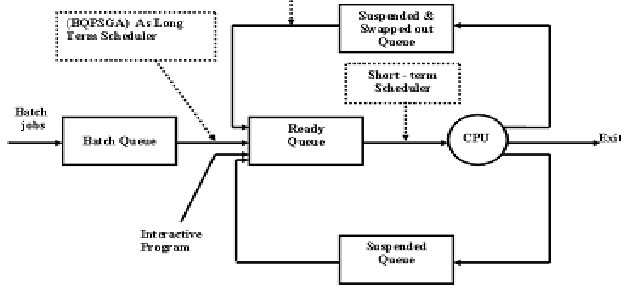

Fig.1.Proposed BQPSGA for OSPSP

5.1 Selection Operator

The performance of the GA is greatly depends upon the Selection operator. Selection operator chooses the fittest individual solution for reproduction. In the present work, "**Modified Random Selection (MRS)"** operator is proposed.

5.2 Crossover Operator

Rajiv et.al. [7] has been proposed ten new crossover operators. These operators are based on permutation encoding. The list of the proposed crossover operators are:

1. Modified Maximal Preservative Right Side Placement Crossover (**MMPX)** Operator
2. Sorted Partially Matched With Same Position Crossover(**SPMX-1**) Operator
3. Sorted Partially Matched Right Side Placement Crossover (**SPMX-2**) Operator
4. Random Order with Random Sequence Placement Crossover (**RORPX**) Operator
5. Random Select with Left shift Placement Crossover (**RLSX**) Operator
6. Random Select with Right Shift Placement Crossover (**RRSX**) Operator
7. Sub Schedule Match Crossover (**SSMX-1**) Operator
8. Sub Schedule Match Crossover with Reverse Placement (**SSMX-2**) Operator
9. Sub Schedule Match Inverted Crossover (**SSMIX-1**) Operator
10. Sub Schedule Match Inverted Crossover (**SSMIX-2**) Operator

5.3 Mutation Operator

Rajiv et.al.[7] has proposed "Selective Swap Mutation (SSM)". The main assumption used here is that sometime due to after much generation, the best genes in the chromosome reach at the extreme position in the chromosome. The extreme positions are the first and last. When the extreme positions of genes are exchanged, then there is tendency to backtrack or to find out the optimal solution.

6 Experimental Setup

To solve the scheduling problem various new operators have been developed by the author. GAPS simulator have been developed [7].GAPSS provide the environment in which different types of operators have been studied. This simulator is designed in C Language.

7.Simulation Result and conclusion

The simulation results show that the proposed genetic operators to solve the OSPSP are efficient. The results are shown in Table No.1.,Table No.2 and Fig. 2,3,4,5 They shows better result as compared to the existing operators. BQPSGA is a new algorithm which can be used as long term scheduler in the operating system.GA cannot be used as short term scheduler because it takes lot of time to produce the result. In case of batch system GA will be applicable.

References

[1] Blazewicz,J.,Ecker,K.,Pesch,E.,Schmidt,G.andWeg larz,J.(2001) scheduling Computer and Manufacturing Processes, Springer Verlag, Berlin, 2001 (2nd edition).

[2] Stallings, W. (2004). Operating Systems Internals and Design Principles (fourth edition). Prentice Hall. ISBN 0-13-031999-6.

[3] Holland,J.H.(1975). Adaptation in Neural and Artificial System, MIT Press, 1975

[4] Goldberg, D. E. (1987). Simple genetic algorithms and the minimal, deceptive problem.

[5] Goldberg, D. E., Korb, B., & Deb, K. (1989). Messy genetic algorithms: Motivation,analysis, and first results. Complex Systems, 3 (5), 493–530.

[6] Davis,L.(1985).Job shop scheduling with genetic algorithms. In J. J. Grcfcnstcttc, editor, Proceedings of the International Conference on Genetic Algorithms and their Applications, pages 136 140. San Mateo: Morgan Kaufmann, 1985

[7] Rajiv Kumar, Sanjeev gill , Ashwani Kaushik , "An Impact of crossover operators on the performance of genetic Algorithm under operating system process scheduling problem" , International Conference on Communication System and Network Technology (IEEE Conference) ,Organized by SMVDU Jamu(J&K) on 03-05 June, 2011, pp704-708,DOI 10.1109/CSNT.2011.150

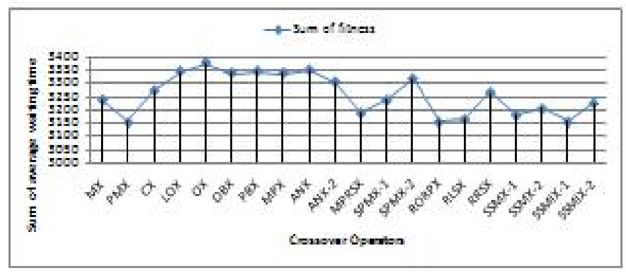

Fig. 2 Comparison of Proposed Crossover Operators with the Available crossover operators w.r.t Total average waiting time for All Data set

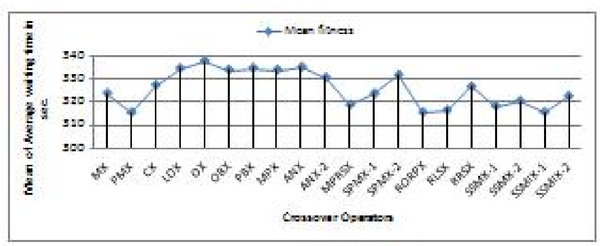

Fig.3 Comparison of Proposed Crossover Operators with the Available crossover operators w.r.t Mean of Total average waiting time for All Data set

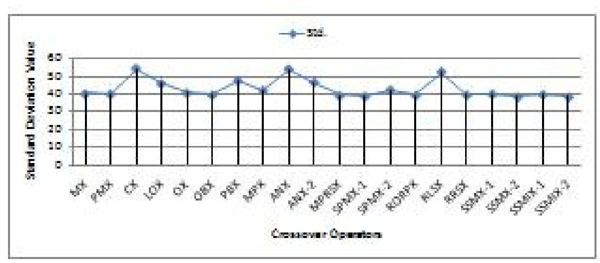

Fig.4 Comparison of Proposed Crossover Operators with the Available crossover operators with average waiting time as fitness function w.r.t Standard deviation for All Data sets

Table 1 Comparison of Proposed Crossover Operators with Available crossover Operators with average waiting Time as a fitness function

Data Set No.	Available Crossover Operators										Proposed Crossover Operators									
	MX	**PMX**	**CX**	**LOX**	**OX**	**OBX**	**PBX**	**MPX**	**ANX-1**	**ANX-2**	**MPRSX**	**SPMX-1**	**SPMX-2**	**RORPX**	**RLSX**	**RRSX**	**SSMX-1**	**SSMX-2**	**SSMIX-1**	**SSMIX-2**
1	293.733	279.933	264.066	295.6	282.066	279.333	286.533	289.266	286.733	289.333	284.933	297.533	287.066	278.8	282.6	349.133	281.2	284.333	279.533	282.333
2	289.933	282.133	313.4	286.666	291.133	292.666	291.266	288.266	293.266	291.866	284.466	348.733	285.066	284.066	229.866	286.533	282.4	290.4	282	286.4
3	261.8	253.333	342.133	264.866	300.533	299.333	294.533	290.466	263	258.4	256	291.4	300.666	256.066	287.733	255.666	257.733	260	253.866	259.733
4	347.333	343.266	312.466	347.866	354.6	368.866	364.866	345.066	348.733	356.866	345.066	329	349.933	344.266	325.533	352.8	343.133	345.6	343.266	347.933
5	301	301.133	361.8	321.4	316.266	308.4	303.266	306.866	316.733	303.8	298.133	375.333	302.066	298.666	318.2	301.2	299.666	303.266	305.333	306.266
6	344.8	341.2	394.4	386.666	382.8	350.8	345	388	386.2	384.066	341.466	368.2	379.6	345.133	388.8	348.733	340.2	347.6	342.4	345.866
7	300	290.133	429.6	323.6	321.533	323.866	296.933	325.533	295	310.533	307.2	281.266	307.2	290.4	260.466	298.066	303.266	302.533	289.133	325.733
8	340.4	324.533	290.6	336.2	357.2	341.866	349.933	331.666	337.933	338	327.866	285.533	327.6	323.4	325.466	328.666	325.333	326.133	323.466	327.466
9	377.2	370.8	293.733	379	394.333	393.533	402.666	375.6	414	390.333	371	290.8	398.666	366.666	367.866	369.866	372.933	369.4	368.866	369.733
10	382.266	368.133	272.533	405.2	378	381.866	412.6	399.133	410.266	383.066	372.4	369.8	380.266	367.733	378.8	376	376.266	375.8	367.866	374.8
Sum	3238.46	**3154.**6	3274.73	3347.06	3378.46	3340.53	3347.6	3339.86	3351.86	3306.26	3188.53	3237.6	3318.13	**3155.**2	3165.33	3266.66	3182.13	3205.06	3155.73	3226.26
Mean	323.846	**315.46**	327.473	334.706	337.846	334.053	334.76	333.986	335.186	330.626	318.853	323.76	331.813	315.52	316.533	326.666	318.213	320.506	315.573	322.626
S.D.	40.1684	40.0476	54.0188	45.7307	40.6840	39.5136	47.3406	41.9158	53.6587	46.5010	39.0369	38.7210	42.2887	39.2924	52.2108	39.5767	39.9963	38.4989	39.6671	38.4618

Table 2. Comparison of Proposed Crossover Operators with Available crossover Operators with Mean Turnaround time as a fitness function

Data Set No.	Available Crossover Operators										Proposed Crossover Operators									
	MX	PMX	CX	LOX	OX	OBX	PBX	MPX	ANX-1	ANX-2	MPRSX	SPMX-1	SPMX-2	RORPX	RLSX	RRSX	SSMX-1	SSMX-2	SSMIX-1	SSMIX-2
1	347.3	333.5	343.	349.2	335.6667	332.9333	340.1333	342.8667	340.3333	342.9333	338.5	355.6	340.6	332.4	336.9	336.6	334.8	337.9	333.1	335.9
2	341.3333	333.5333	371.5333	338.0667	342.5333	344.0667	342.6667	339.6667	344.6667	343.2667	335.86	448.4	336.4667	335.4667	274.4	337.933	333.8	341.8	333.4	337.8
3	341.5333	333.0667	441.8	344.6	380.2667	379.0667	374.2667	370.2	342.7333	338.1333	335.7333	357.0667	380.4	335.8	367.4667	335.4	337.46	339.7333	333.6	339.46
4	417.3333	413.2667	378.1333	417.8667	424.6	438.8667	434.8667	415.0667	418.7333	426.8667	415.0667	404.6	419.9333	414.2667	392.1333	422.8	413.13	415.6	413.2667	417.93
5	359.1333	359.2667	437.4	379.5333	374.4	366.5333	361.4	365	374.8667	361.9333	356.2667	462.7333	360.2	356.8	376.3333	359.3333	357.8	361.4	363.4667	364.4
6	444.4667	440.8667	481.8	486.3333	482.4667	450.4667	444.6667	487.6667	485.8667	483.7333	441.1333	463.7333	479.2667	444.8	525.7333	448.4	439.87	447.2667	442.0667	445.5
7	365.6667	355.8	525.1334	389.2667	387.2	389.5333	362.6	391.2	360.6667	376.2	372.8667	334.8667	372.8667	356.0667	322.6667	363.7333	368.93	368.2	354.8	391.4
8	416	400.1333	344.2	411.8	432.8	417.4667	425.5333	407.2667	413.5333	413.6	403.4667	336.9333	403.2	399	401.0667	404.2667	400.93	401.7333	399.0667	403.0667
9	464.6	459.0667	345.1333	466.4	481.7333	480.9333	490.0667	463	501.4	477.7333	458.4	370.5333	486.0667	454.0667	455.2667	457.2667	460.33	456.8	456.2667	457.1333
10	477.8	463.666	352.266	500.733	473.533	477.4	508.133	494.666	505.8	478.6	467.933	465.333	475.8	463.2667	474.3333	471.5333	471.8	471.3333	463.4	470.3333
Sum	3975.2	3892.2	4021.2	4083.8	4115.2	4077.267	4084.333	4076.6	4088.6	4043	3925.267	3999.867	4054.867	**3891.93**	3926.33	3937.33	3918.67	3941.8	3892.467	3963
Mean	397.52	389.22	402.12	408.38	411.52	407.726	408.433	407.66	408.86	404.3	392.526	399.986	405.486	**389.193**	392.633	393.733	391.87	394.18	389.2467	396.3
S.D.	52.8950	52.8676	65.2346	59.4282	55.6695	53.3341	60.8992	57.1587	67.2292	59.8812	51.7131	55.3035	57.5463	52.3046	75.4694	53.6345	52.90	51.45575	52.35461	51.05265