

Genetic algorithm Approach for Optimal CPU Scheduling

¹Preeti Sindhvani, ²Vaishali Wadhwa

^{1,2}N.C.College of Engineering, Israna, Panipat, Haryana, India

Abstract

Scheduling in many different areas is a 'NP-complete' problem. i.e. algorithms that implement scheduling require exponential time to reach a solution. These problems require innovative solutions. Genetic algorithms can be used for finding innovative solutions for scheduling. This Paper contains how we can use genetic algorithm in finding an innovative solution for CPU Scheduling. Using an efficient encoding, and designing appropriate crossover and mutation operators for our problem may achieve this. The algorithm requires a population of individuals. Each individual is an encoded version of a proposed solution. The algorithm consists of the evaluation of individuals, selection of individuals, which will contribute to the next generation, recombination of the parents by means of crossover, mutation and other operators to produce a new generation.

Keywords

CPU scheduling, NP Hard, genetic algorithm, crossover, mutation, selection.

I. Introduction

Scheduling in many different areas falls into the category of 'NP-complete' problems. i.e. current algorithms require exponential time to reach a solution 'NP-complete' so, it require innovative solutions[2]. Moreover, it is a critical factor that effect the operating system efficiency In process scheduling we allocate processes to processor in such a order so that throughput and efficiency of the resulting system is maximized. This paper proposes the genetic algorithm based technique to find out the optimal job sequence in a single-processor environment. Genetic algorithms were first proposed by the Holland in the 1960s [1]. Genetics algorithm based on the principle of genetics and evolution. The algorithm starts with a search space or population which is consists of all feasible solution to the optimization problem. A member of population is called an individual or chromosomes. A fitness value is associated with each individual. A chromosome is subdivided into genes. A gene is the GA's representation of a single factor for a control factor. The values taken by genes are called alleles. Several pair of individual (parents) in the population mate to produce offspring by applying the genetic operator. Crossover. Selection of parents is done by repeated use of a choice function[2]. Then we replace old individuals in the population with these new offspring such that the number of individual in the new population is the same as old population. Each serving string undergoes inversion with a specified probability. The research will specifically try to find a genetic algorithm that makes automatic iterative scheduling practical for modern but relatively low cost computing equipment. This may be achieved by using an efficient encoding, and designing appropriate crossover and mutation operators for our problem.

Now, our problem is that we have N no. of processes numbered as (1, 2, 3...N).and we are working on a single processor environment. Our aim is to allocate these processes to CPU in such a order so that the resulting waiting time is minimized. And we assumed that all processes is having equal priority. and all processes are ready and waiting for the CPU allocation.

II. Genetic Algorithm

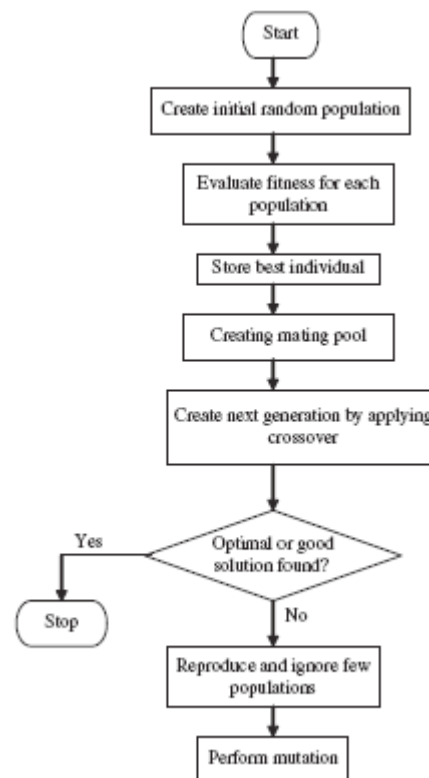


Fig. 1: Showing the steps involved in genetic algorithm
The algorithm used in the paper

1. Initialize Solution space. Here we used the random generated solution space.
2. Encoding the solution space (by using real no. Encoding)
3. then we evaluate the fitness using a fitness function.
4. While (best individual is not found)
5. Selection (by using the Roulette wheel Selection method)
6. Crossover (by using the modified ordered crossover operator)
7. Mutation (Inversion)
8. Fitness Evaluation;
9. Replacement of the old parent with new offspring (by using weak parent replacement)
10. Return the best solution found.

III. Key elements

A. Individual

An individual is a single solution [3]. Individual groups together two forms of solutions as given below:

1. The chromosome, which is the raw 'genetic' information (genotype) that the GA deals.
2. The phenotype, which is the expressive of the chromosome in the terms of the model.

B. Search space (population)

A population is a collection of individuals.
Or the set of all feasible solution.

C. Encoding

Encoding is a process of representing individual genes. In permutation encoding

Every solution is a string of numbers, which represents the number in sequence.

Sometimes corrections have to be done after genetic operation is completed. E.g.

Chromosome A 1 5 3 2 6 4 7 9 8

Chromosome B 8 5 6 7 2 3 1 4 9

Permutation encoding is only useful for ordering problems.

D. Fitness function

In order to find the optimal solution a fitness function must be designed for the problem under consideration. For a particular solution the fitness function returns a single numerical value, which is proportional to the ability of the solution [1].

And our fitness function should calculate the average waiting time of the given solution. The solution, which is having the minimum value as, calculated from the fitness function will be the one that is having the minimum waiting time and will be the fittest.

The fitness function of a Solution S_m is given by

Fitness(S_m) =

$$\frac{\sum_{i=1}^N W_i}{N}$$

($i=1, 2, 3, \dots, N$). W_i is the waiting time of process $S_m(i)$, N is the total no. of processes.

E. Selection

Selection is the process of choosing two parents from the population for crossing[4]. After deciding on an encoding, the next step is to decide how to perform selection i.e., how to choose individuals in the population that will create offspring for the next generation.

F. Roulette Wheel selection

Each individual is assigned a slice of the roulette wheel, the size of the slice being proportional to the individual's fitness. The wheel is spin N times, where N is the number of individuals in the population. On each spin, the individual under the wheel's marker is selected to be in the pool of parents for the next generation [1].

This method is implemented as follows:

1. Sum the total fitness value of the individuals in the population. Let it be f .
2. Repeat N times:
 - i. Choose a random integer 'p' between 0 and f .

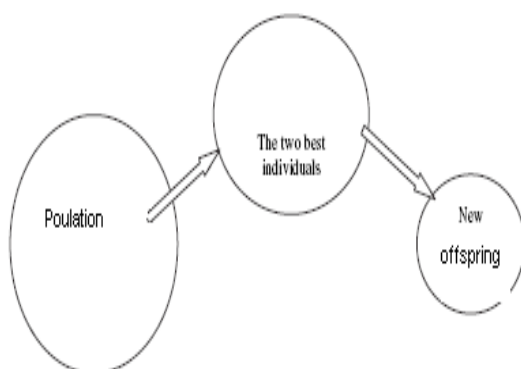


Fig. 2: showing the process of selection

- ii. Loop through population, summing the fitness values, until the sum is greater than or equal to 'p'. The individual whose expected value puts the sum over this limit is the one selected.

G. Crossover

Crossover is the process of taking two parent solutions and producing a child from them.

We used modified crossover method in our algorithm.

H. Ordered Crossover

Ordered two-point crossover is used when the problem is of order based, for example in U-shaped assembly line balancing etc. Given two parent chromosomes, two random crossover points are selected partitioning them into a left, middle and Right portion. The ordered two-point crossover behaves in the following way: c inherits its left and right section from parent 1, and its middle section is determined

Parent 1: 4 2 | 1 3 | 6 5

Parent 2: 2 3 | 1 4 | 5 6

Child 1: 4 2 | 3 1 | 6 5

Child 2: 2 3 | 4 1 | 5 6

by the genes in the middle section of parent 1 in the order in which the values

Appear in parent 2. A similar process is applied to determine child 2.

I. Modified Ordered Crossover

In modified ordered crossover the two crossover points are the processes having the longest and shortest burst time.

J. Mutation

After crossover, the strings are subjected to mutation. Mutation prevents the algorithm to be trapped in a local minimum. Mutation plays the role of recovering the lost genetic materials as well as for randomly disturbing genetic information. It is an insurance policy against the irreversible loss of genetic material. Mutation has traditionally considered as a simple search operator. If crossover is supposed to exploit the current solution to find better ones, mutation is supposed to help for the exploration of the whole search space. Mutation is viewed as a background operator to maintain genetic diversity in the population. It introduces new genetic structures in the population by randomly modifying some of its building blocks. Mutation helps escape from local minima's trap and maintains diversity in the population. It also keeps the gene pool well stocked, and thus ensuring ergodicity. A search space is said to be ergodic if there is a non-zero probability of generating any solution from any population state.

There are many different forms of mutation for the different kinds of representation. For binary representation, a simple mutation can consist in inverting the value of each gene with a small probability. The probability is usually taken about $1/L$, where L is the length of the chromosome. It is also possible to implement kind of hill-climbing mutation operators that do mutation only if it improves the quality of the solution. Such an operator can accelerate the search. But care should be taken, because it might also reduce the diversity in the population and makes the algorithm converge toward some local optima.

K. Inversion

Inversion operator is a primary natural mechanism to recode a problem[5]. In inversion operator, two points are selected along

the length of the chromosome, the

Chromosome is cut at those points and the end points of the section cut, gets reversed

(Switched, swapped). To make it clear, we consider a chromosome of length 8, where two inverse points are selected randomly (the points are 3 and 5 denoted by character as shown in Fig.)

Parent 4 2 7 6 8 1 3 5

Child 4 2 8 6 7 1 3 5

L. Replacement

Two parents are drawn from a fixed size population, they breed two children, but not all four can return to the population, so two must be replaced i.e., once off springs are produced, a method must determine which of the current members of the population, if any, should be replaced by the new solutions. The technique used to decide which individual stay in a population and which are replaced in on a par with the selection in influencing convergence. Basically, there are two kinds of methods for maintaining the population.generational updates and steady state updates.

M. Weak Parent Replacement

In weak parent replacement, a weaker parent is replaced by a strong child [6]. With the four individuals only the fittest two, parent or child, return to population. This process improves the overall fitness of the population when paired with a selection technique that selects both fit and weak parents for crossing, but if weak individuals and discriminated against in selection the opportunity will never raise to replace them.

N. Termination criteria

The genetic process will end if there is no change to the population's best fitness for a specified number of generations

IV. Pseudo-code

```
(1) Begin /* MCGA Algorithm*/
(2) Generate random population using random1 () function
(3) Evaluate each individual using fitness fitness() function.
(4) WHILE NOT finished Do
    BEGIN /* Produce New generation */
(5) Select two individual from the old population for mating using
    Selection () function
(6) Recombine the two individuals by apply Modified ordered
    crossover operator using mocrossover () function.
(7) Apply inversion operator using inversion () function to give
    offspring
(8) Compute the fitness of offspring and Insert the two fittest
    offspring into the new Population using replace()
    function.
```

END

(9) IF termination criteria has been reached

THEN

Finished: = TRUE

END

END

V. Experimental Description

There are 6 jobs, which are to be considered. The number of possible sequences is 6! The total 10 sequences are selected out

of 720 for the 6 jobs.

Considering the number of jobs as 6 and the crossover point is 2 and 4. let us consider following two individuals, which are marked as fit to generate next generation.

3 1 2 4 5 6 and 6 4 3 5 1 2

After cross over Child 1: 3 4 1 2 5 6

Child 2: 6 3 4 5 1 2

And let the inversion point be 2 and 5

After inversion Child 1: 3 5 1 2 4 6

Child 2: 6 1 4 5 3 2

After that we calculate the fitness of parent1, parent2, child1, child2 and two fittest individual are returned to the solution space.

And above step is repeated until convergence criteria meet.

VI. Results

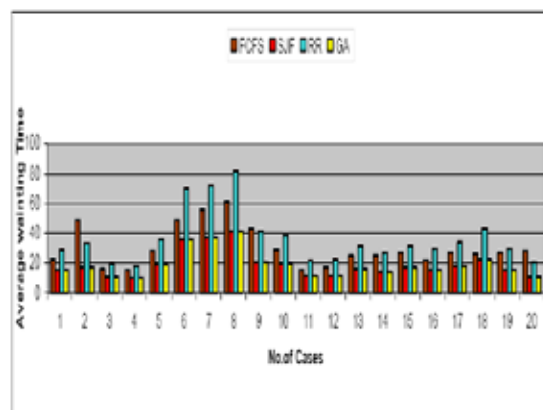


Fig. 3: Showing the result

VII. Conclusion

Genetic algorithms for the CPU Scheduling. The Experiment results clearly show that the proposed approach is able to find optimized solution. The experiment carried out is efficient to find best sequence. The work can be extended so that technique can be implemented for dynamic process scheduling and sequencing. The performance can also increase by apply diploidy operator on this problem.

The description of the genetic algorithm included many genetic terms. In order to better understand how genetic algorithms can be applied to combinatorial optimization problems, the following equivalence will be useful. Combinatorial Optimization Genetic Algorithm, Encoded Solution, Chromosome Solution, Decoded Chromosome Set of Solutions, Population Objective function.

References

- [1] Introduction to genetic algorithm by S.N.Shivanandnam and S.N.Deepa.
- [2] Genetic Algorithm approach to Operating System process scheduling Problem Dr.Rakesh Kumar, Reader Department of Computer Science and Application, Kurkshetra university, Haryana, India
- [3] 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
- [4] Blazewicz, J., Domschke, W., and Pesch, E. (1996). The job shop scheduling problem: Conventional and new solution techniques. European

Journal of Operational Research

- [5] Holland, J.H., 1975. "Adaptations in natural and artificial systems", Ann Arbor: The University of Michigan Press.
- [6] David E. Goldberg, Genetic Algorithms in Search Optimization & Machine Learning, Second Reprint, Pearson Education Asia Pte.Ltd., 2000.



Preeti Sindhwani did her B.Tech. from N.C.College of Engineering, Israna(Panipat). Now she is pursuing M.Tech. from N.C.College of Engineering, Israna(Panipat). She is also Lecturer in N.C.College of Engineering, Israna(Panipat). Her research interest includes approximation and optimization based algorithms.



Vaishali Wadhwa did her B.Tech. from A.P.J,Gurgaon and she did her M.Tech. from N.C.College of Engineering, Israna(Panipat). Now she is pursuing Ph.D. from Thapar university.. She is also Assistant Professor in N.C.College of Engineering, Israna(Panipat). Her research interest includes approximation and optimization based algorithms.