Genetic Algorithm – An Approach To Solve Out The Scheduling Problems In Multi-Processor System

Gaytri [1],
Asstt. Prof. Ramesh Yadav [2]

[1]M.Tech* (P) B.I.T.S. Bhiwani. M.D.U.
[2]Asstt.Prof. B.I.T.S. Bhiwani. M.D.U.

Abstract

The scheduling problem in a multilayer multiprocessor environment finds its relevance in many industrial and computing applications. In this paper, we present a genetic algorithm for the solution. Extensive computational experiments demonstrated that genetic algorithm performs well.

In literature, several heuristic methods have been developed that obtain suboptimal solutions in less than the polynomial time. Recently, Genetic algorithms have received much awareness as they are robust and guarantee for an effective solution. Genetic algorithm is wildly used to solve Flexible Task Scheduling Problem. The genetic algorithm we proposed uses many different strategies to get a better result. During the phase of create initial population, the genetic algorithm takes into account the number of operations in each job. And the intelligent mutation strategy is used which makes every individual and gene have different probability to mutate.

In this paper, the object of scheduling algorithm is to get a sequence of the operations on machines to minimize the make span.

1. Introduction

1.1 Scheduling Of Tasks

For performing multiple task over a time we must assign them in proper sequence or phase so that they utilize resources in best suited manner i.e. Known as task scheduling.

Many parallel applications consist of multiple computational components. While the execution of some of these components or tasks depends on the completion of other tasks, others can be executed at the same time, which increases parallelism of the problem.

The task scheduling problem is the problem of assigning the tasks in the system in a manner that will optimize the overall performance of the application, while assuring the correctness of the result.

The task scheduling problem can be modeled as a weighted directed acyclic graph (DAG). A vertex represents a task, and its weight the size of the task computation. An arc represents the communication among two tasks, and its weight represents the communication cost. The directed edge shows the dependency between two tasks.

In computer science, scheduling is the method by which threads, processes or data flows are given access to system resources (e.g. Processor time, communications bandwidth). This is usually done to load balance a system effectively or achieve a target quality of service. The need for a scheduling algorithm arises from the requirement for most modern systems to perform multitasking (execute more than one process at a time) and multiplexing (transmit multiple flows simultaneously).

2. Task Scheduling In Multiprocessor

Task Scheduling in Multiprocessor is a term that can be stated as finding a schedule for a general task graph to be executed on a multiprocessor system so that the schedule length can be minimized. Task scheduling in multiprocessor systems also known as multiprocessor scheduling. Multiprocessor scheduling problems can be classified into many different categories based on characteristics of the program and tasks to be scheduled, the multiprocessor system, and the availability of information. Multiprocessor scheduling problems may be divided in two categories: Static and dynamic task scheduling. A static or deterministic task scheduling is one in which precedence constraints and the relationships among the task are known well in advance While non-deterministic or dynamic scheduling is one in which these information is not known in advance or not known till run time. A major factor in the efficient utilization of multiprocessor system is the proper assignment and scheduling of computational tasks among processors.

2.1 GENETIC ALGORITHMS

Genetic Algorithms are good at taking large, potentially huge search spaces and navigating them, looking for optimal combinations of things, solutions you might not otherwise find in a lifetime.

Simple Genetic Algorithm

```
{
initialize population;
evaluate population;
while Termination Criteria Not Satisfied
{
select parents for reproduction;
perform recombination and mutation;
evaluate population;
}
```

Chromosome generation

The details to generate a chromosome can be seen in
following steps:
1:Select randomly a task from the entire entry tasks. Set
this task as the first task in SQ.
2: Repeat step 3 for (v-1) times.
3: Randomly select a task who is not in SQ and whose
predecessors all have been in SQ, and add this task to SQ.
4: For SP part, randomly generate an integer number
between 1 and m for each task in SQ and add it to SP.

3.Problem Description & Objective

To solve out the multi processer scheduling problem which is defined as the assignment of a given set of tasks to a set of processors in such a way that scheduling length of task is minimum &

all precedence tasks run effectively (minimum execution time).

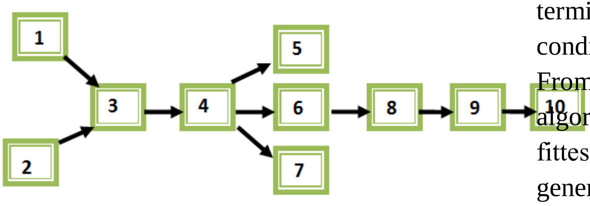


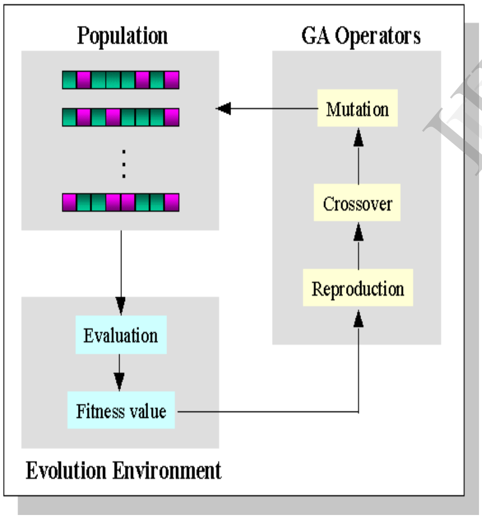
Fig. 1 Directed Acyclic Graph of 10 nodes and their dependencies on each other

4. Methodology

In our work, we implemented Genetic algorithm for solution of multiprocessor task scheduling problem.

Detail block diagram (Fig.) represents the sequence of

operations to get the desired results.



Typically, a genetic algorithm consists of the following steps:

GA1: Initialization –initialize the population.

GA2: Evaluation –evaluate each chromosome using fitness function.

GA3: Genetic operations –Select the parent and apply genetic operators on them to produce new chromosomes.

GA4: Repeat steps GA2 and GA3 until termination

condition reached.

From the above steps, we can see that genetic algorithms utilize the concept of survival of the fittest; passing "good" chromosomes to the next generation, and combining different strings to explore new search points.This is a searching process based on the laws of natural selection and genetics. Usually, a simple GA consists of three operations: Selection, Genetic Operation, and Replacement The population comprises a group of chromosomes from which candidates can be selected for the solution of a problem. Initially, a population is generated randomly. The fitness values of the all chromosomes are evaluated by calculating the objective function in a decoded form (phenotype). A particular group of chromosomes (parents) is selected from the population to generate the offspring by the defined genetic operations. The fitness of the offspring is evaluated in a similar fashion to their parents. The chromosomes in the current population are then replaced by their offspring, based on a certain replacement strategy. Such a GA cycle is repeated until a desired termination criterion is reached (for example, a predefined number of generations is produced). If all goes well throughout this process of simulated evolution, the best chromosome in the final population can become a highly evolved solution to the problem.

5. Result & Discussion

In our work, we implemented an algorithm for solution of multiprocessor task scheduling problem. We assume that the number of tasks considered in the problem is 10 and processors are 3, but these can be varied. There can be maximum 6 number of predecessors of a node shown in Fig. 1. For performance evaluation of our algorithm we take different parameters values for different parameters. As two important

GA control parameters, crossover probability (pc) and mutation probability (pm) affect the performance of GAs drastically. A set of good GA parameters help in improving the ability of a GA to search for near global optimal solutions.

We also examines the effects of GA numerical parameters on its performance in terms of average fitness found for this specific problem domain. It also describes the observed behavior of genetic algorithm in case of task scheduling. How the average fitness increases or decreases when the parameters of GA are varied? Some results are taken here after varying some parameters of genetic algorithm. These parameters are:

1. Population size (POP)
2. Number of generations (GEN)
3. Crossover probability (Cp)
4. Mutation probability (Mp)

6. Conclusion

One of the most challenging problems in computing is the problem of scheduling of tasks to be executed on a multiprocessor system is. Genetic algorithms are well adapted to multiprocessor scheduling problems. As the resources are increased available to the GA, it is able to find better solutions. A genetic algorithm based on principles of evolution found in nature for finding an optimal solution. Genetic Algorithm use random choices to guide themselves to the problem space and they are used for different kind of task scheduling problems. It continuously tries to improve the average fitness of a population by construction of new population .Quality of solution depends heavily on the selection of some key parameters like fitness function ,population size , crossover probability and mutation probability. Task scheduling in multiprocessor system using genetic algorithm is an efficient way due to the characteristics of Genetic Algorithm as : quality of solution ,robustness and guarantee for good solution effect of mutation probability on the performance on Genetic Algorithm.

References:

[1] [1] Neelu Sahu, Sampada Satav "Task Scheduling Using Compact Genetic Algorithm for Heterogeneous System" [2012]

[2] Rachhpal Singh " Modified Genetic Algorithm Approach to Optimize Task Scheduling on Heterogeneous Multiprocessor Parallel System using Node Duplication"[2012]

[3]Mostafa r. Mohamed, Medhat h. A. Awadalla"Hybrid Algorithm for Multiprocessor Task Scheduling" IJCSI International Journal of Computer Science Issues, Vol. 8, No. 2, May 2011

[4] Sachi Gupta, Gaurav Agarwal, "Task Scheduling in Multiprocessor System Using Genetic Algorithm", Second International Conference on Machine Learning and Computing, IEEE 2010.

[5] Adel Manaa and Chengbin Chu, "Scheduling multiprocessor tasks to minimize the makespan on two dedicated processors", European Journal of Industrial Engineering, pp. 265 – 279, Volume 4, 2010.

[6] Intisar A.Majied Al-Said, Nedhal Al-Saiyd, Firas Turki Attia, "Multiprocessor scheduling based on genetic algorithms", 2009.

[7] Amir Masoud Rahmani and Mojtaba Rezvani, "A Novel Genetic Algorithm for Static Task Scheduling in Distributed Systems", 2009.

[8] Ali Pedram, "A method for scheduling multi processing systems with genetic algorithm", International Journal of Engineering and Technology Vol. 1, No. 2, June, 2009.

[9] S.N.Sivanandam, S.N.Deepa, “Introduction to Genetic Algorithms”, Springer-Verlag Berlin Heidelberg, 2008.

[10] Amir Masoud Rahamani, Mohamad Ali Vahedi “A novel Task Scheduling in Multiprocessor Systems with Genetic Algorithm by using Elitism stepping method”, 2008.

[11] Yajun Li, Yuhang Yang, Maode Ma, Rongbo Zhu, “A Problem-Specific Genetic Algorithm for Multiprocessor Real-time Task Scheduling”, The 3rd Intetnational Conference on Innovative Computing Information and Control (ICICIC'08), IEEE 2008.

[12] Peyman Almasi Nejad, Ahmad Farahi, Davood Karim Zadegan Moghadam, Reza Asgary Moghadam, “An Intelligent Method for Multi Processor Scheduling using Genetic Algorithms”, International Conference on MultiMedia and Information Technology, IEEE 2008.

[13] Dr. Franz Rathlauf, “Representations for Genetic and Evolutionary Algorithms”, 2nd edition, @ Springer. 2006.

[14] Tang, K.S., K.F. Man, S. Kwong and Q. He. "Genetic algorithms and their applications", IEEE Signal Processing Magazine, vol.13, 2004.