Hybrid Genetic Algorithm Based Task Scheduling in Heterogeneous Grid

Uma B. Gurav and L.D. Netak

Abstract-- **In this paper, the problem of task scheduling of dynamically arrived tasks to the processors in heterogeneous grid computing environment is solved, Dynamic task scheduling , is solved using an advanced evolutionary algorithm called hybrid genetic algorithm. The execution of the task in a shortest amount of time and speed of the processor are the key factors in the evaluation of the fitness of the processor. Hybrid genetic algorithm is recent concept of genetic algorithm which is a combination of genetic algorithm as well as local search. The paper presents a novel solution based on , hybrid genetic algorithm, which is efficient, simple & faster compared to genetic algorithm. This novel representation makes it possible to solve the problem with a small population and in a few generations. It allows use of an efficient local search operator within the evolutionary algorithm. The proposed algorithm is proven very efficient in many other scheduling problems and applications. A comparative analysis with the results from previous work & simulation result proves that this proposed hybrid genetic algorithm can produce good quality solutions in a very short space of time.**

Index Terms-- **hybrid genetic algorithm, crossover, mutation, fitness scaling, fitness function, no. of generation , size of population**

I. INTRODUCTION

THE problem of scheduling of tasks onto heterogeneous processors in dynamic and heterogeneous grid computing environment (GCE) is very complex job, we can incorporate ARTIFICIAL INTELLIGENCE in GCE likewise resource management, grid maintenance, information security. Self – adaptive Path scheduling of tasks is one of the intelligence that can be incorporated in GCE with the help of hybrid genetic algorithms. [18, 10]

In GCE the tasks should take a path, once the user submits a job to the resource, which may be dynamic in nature. The path changes at every instance of time as processors are added or removed due to failure by scheduler Effective dynamic route establishment requires certain criteria fulfillment such as Computation time required to fulfill requests of the job, determination of current load of the system and prediction of job execution time etc. In this paper, the importance is given on only two factors speed of processor, and Expected computation time.

Due to the complexity of Grid systems and large scale distributed applications, many researchers have considered different versions and modes of scheduling.

The main consideration is given in this paper on a version of the problem, which doesn't take into account possible restrictions on job interdependencies, data transmission and economic and cost policies on processors. The focus is on, the scheduling version needed to achieve high performance applications and, from the Grid users perspective, to offer QoS (Quality Of Service) of the Grid system. This type of scheduling arises in applications that can be solved by splitting them into many independent jobs, submitting them to the Grid and combining the partial results to obtain the final solution. Moreover, in Grid systems there is a need for allocating independent user applications to Grid resources. Thus, this paper considers the scenario in which tasks submitted to the Grid are independent and are not preemptive (they cannot change the resource they have been assigned to once their execution is started, unless the resource is dropped from the Grid.) [18,20]

II. HYBRID GENETIC ALGORITHM

Genetic algorithms are search algorithm based on mechanics of natural selection and natural genetics. Genetic algorithm, works with individuals, each representing a solution to the problem being tackled. A fitness function is defined in order to get the fittest of the solution space. This optimization technique is used to get the best solution out of the solution search space; An initial population is created and then evolved by means of genetic operators, such as crossover and mutation, to form a new population (the next generation) that is hoped to be fitter than the last one. The crossover operator is applied to pairs of individuals in order to interchange their genetic material. By applying this operator to the fittest individual, good properties should propagate down the generations. The mutation operator makes random changes in the genetic material by mutating a single bit or gene, thus allowing the genetic algorithm to explore new corners of the search space. The evolution process is repeated a number of iterations or until some other criterion is met, depending on the population size , and the best solution is sought from the final population of individuals.

In previous works, Scheduling in a Grid computing environment using Genetic Algorithms, the problem of resource allocation is handled using genetic algorithm which leads to more no. of generations and large population size, Using hybrid G.A the no. of generation as well as population size can be reduced.

Hybrid G.A. use local search methods such as tabu search, random walk model etc, to find local optimums i.e. a point with the best fitness value among its neighbor points. The hybrid G.A is faster and more accurate than a simple genetic algorithm for some reasons.

This work is supported by DR. Babasaheb Ambedkar Technological University, Lonere, INDIA.

Uma Gurav is with the Department of Computer Engineering , Dr.D.Y.Patil College ,Maharashtara, Mumbai University, India
(e-mail: umabgurav@yahoo.co.in)
L. D. Netak is with the Department of Computer Engineering, Dr. Babasaheb Ambedkar Technological University, Lonere, India
(e-mail: ldnetak@yahoo.com)

First, local search methods can serve the genetic operators with solutions those are better in comparison to randomly generated solutions. Second, genetic algorithms are not good hill-climbers and the combination of them with local search methods alleviates this problem [8]. Third, using hybrid G.A the no. of generations and size of population is reduced.

A.PROBLEM FORMULATION.

Formal definition of an instance of the problem is , as follows. It consists of following parameters:

- A number of independent tasks that must be routed. Any task has to be processed entirely in unique resource.
- A number of heterogeneous processors candidates to participate in the planning.
- The existing workload (in millions of instructions) of each job.
- The computing capacity of each processor (in MIPS).
- The time to complete the task from historical information of the processor. This parameter measures the previous workload of a processor.

III. HYBRID GENETIC ALGORITHM TO SOLVE THE DYNAMIC TASK SCHEDULING PROBLEM OF TASKS

In this section, proposed hybrid genetic algorithm for solving the resource allocation problem is described,, There are set of operations which can be used are selection, crossover, mutation, local search . The novel solution which is used in this paper, has advantages such as,

A. *Chromosome representation*

- The number of genes for a chromosome required in hybrid genetic algorithm will be less in contrast to genetic algorithm‘s chromosomes.
- The size of the population (SOP) of GCE will be independent from the size of the environment since SOP is reduced due to local search.

These characteristics make genetic operations such as crossover, mutation, and selection simpler.
Hence chromosomes with invalid information of the path are detained are not carried over in next generation.

For example, If user submits a task and later on cancels or exits it, that means it is a invalid task, but since it has arrived in grid, scheduler will collects it's information, Schedules it . But because of local search filtering mechanism the invalid tasks are not carried over in next Generation. Normally, genes are carried over in next generation either in a random manner or either mutated or crossovered,

Hence because of this filtering mechanism at the initial stage (i.e. local search) the invalid chromosomes are not carried over and no. of generations required for calculations are reduced.

In this hybrid genetic algorithm based scheduling problem, the objective is upon arrival of a Task, it should be able to be allocated to a processor. There is some criteria for resource selection mechanism, i.e. with the help of proposed algorithm dynamic route is established between the scheduler and processor. The application will start to run on any set of processors, during the task execution scheduler periodically collects information about the execution times, speed of processor, and also it has to maintain record of which processor is idle, this statistics is used to automatically estimate the resource requirements of the task, depending on this information our proposed algorithm establishes a dynamic route.

The proposed algorithm, works with a population where each individual is possible route between the scheduler –processor pair of the network. The coding of a route is an array of integers, where each integer identifies a resource traversed by a route.

The algorithm first generates an initial population of size P randomly by means of scheduler which generates initial population of processors available, The scheduler starts by pointing at scheduler which is nothing but source node say S. The pointed node is added to the route and marked as visited; the information about visited processor is collected. Suppose if the task is decomposable then it can be divided into subtask, and the subtask is assigned to the visited node. Then a node say J is randomly selected using any local search algorithm,

(the initial focus is on local search in hybrid G.A) , Using any local search algorithm the search procedure becomes a fast , since genetic algorithms are not good hill climbers , It is now established from that it is a very hard for a 'pure ' GA to 'fine tune' the search in a complex spaces . Researchers and practitioners have shown that a combination of global and local search is almost always beneficial.

Randomly selected node J is added to the route, and marked as visited, now the scheduler points at node J. Collects all the information about it and if the task is decomposable then subtask is assigned to that processor, depending on it's h/w, s/w configuration .The process is repeated until all the processors are visited, and all the dynamically arrived tasks have completed their execution either in a parallel manner or in a distributed manner. When another tasks arrives the algorithm starts it's execution again.

Here, a chromosome represents path and processor is represented by gene.

B. *Genetic operators*

I. *Crossover operator*

This operator is applied to the pair of routes say X and Y, which have at least one node in common, apart from source and destination nodes. Since not every pair of routes which are established dynamically, can be crossed, this crossover operator is not applied to randomly chosen pairs as conventional GA'S do.

Instead this scheduling algorithm examines first scheduler information and all possible pair information, beginning with those pairs which include individuals with higher fitness value, First, common nodes are searched in the parent route nodes . If any common, one of them is chosen randomly and the parent nodes are divided into two halves by that node . . The children are made one by binding first half of the chromosome of A to the second half of the chromosome B and the other half by binding first half of the chromosome of B to the second half of the chromosome of A. If any of the resulting children's are valid routes (it should not contain any cycle) and it is different from parents, then it is added to children's population. This procedure continues, until population is exhausted.

But since ,arrival of tasks are considered in a dynamic manner the population size will not be fixed always, it will keep on changing as tasks comes and goes or processors are added and removed, but here certain limit on size of population is imposed and after reaching that particular point it should iterate again. (figure 1)

II. Mutation Operator

The mutation of a route is performed as follows, a node from a route is selected randomly, and a new route is generated randomly from the selected node (mutation node) to the destination node, using the information that scheduler has. (figure 2)

C. *FITNESS SCALING MECHANISM*

After applying crossover to the population, individuals that are to form next generation population have to be selected using fitness scaling mechanism. Selection of the survivors is done by ranking mechanism, that is , the algorithm selects the P individuals with best fitness values.

IV. FITNESS FUNCTION EVALUATION AND IMPLEMENTATION

Theoretically, the precise execution time of a task can not be predicted in advance because of dynamic nature of grid. So, in proposed model, the prediction of possible execution time of job is done from historical experiences and use this prediction to schedule the task onto node-processor. Here the most difficult part is to deal with dynamic load of the system that affects the execution time of the job to be submitted. In the proposed model dynamic load is not considered.

The prediction of the execution time of a job on a node from historical usage of information as follows:

$$T_{p,e}(t+1) = (1 - S_p(t))\ T_{p,e}(t) + S_p(t)\ T_{p,act}(t)$$

Where, $T_{p,e}$ **$(t+1)$** is the predicted execution time e on a processor p of a new job , t denotes the no. of times that the job has been executed on processor p . Tp, act(t) is the actual execution time of the job , Sp is the speed of the processor . From the previous information of n no. of generation, the prediction is done in (n+1) iterations.

And fitness is calculated as,

$$F(n_i) = \frac{\sum_{i=1}^{P} n_i\ T_{p,e}(t)}{N}$$

Scheduling N tasks on p processors on the grid. To solve this scheduling problem , the prediction model to calculate the execution time on different processor is used . The fitness function is evaluated based on the condition, the processor which can execute the task in shortest amount of time will be assigned a task.

A. Selection

Theoretically, the node which can complete this task in the shortest time, has the biggest probability of getting the job. In the roulette wheel selection, Time for execution of all nodes is sorted in ascending order Tp,e ,the biggest proportion is allocated to the node which can finish the job in a shortest time. Actual execution time will increase as the system runs. Selection is made depending on the fitness function value once the population size exceeds predefined value defined in the algorithm.

B. Hybrid Genetic Algorithm based task scheduling algorithm

```
Initialize the population parents;
- T^i p,e , Sp, T(p, act) for i = 1….p; for N no. of tasks.

Repeat until population gets exhausted or no new tasks
are there in the search space

Do while (tasks arrives dynamically in grid)
{
Local_search (parents, probability)
Compute every T^i p, e for i = 1 …… p;
Sort in ascending order of T^i p, e for i = 1 …… p;
Assign using selection mechanism.
mating_pool: = select (parent1, parent2);
offsprings: = crossover (mating_pool);
Mutate (offsprings) ;
Parents: =select (parents, offsprings) ;
If (job is finished )
{
update the predicted   completion time,
update other resource information SpTp;
}
}
End;
```

V. SIMULATION RESULTS

In analyzing the performance of hybrid genetic algorithm, the hybrid genetic algorithm discussed in the previous section is implemented and tested on randomly arrived task values. Based on execution time of a task on a processor the fitness function value is calculated, in this no. of generation varies from 20-60. The Hybrid GA used following parameters: population size: 0-200

Crossover probability: 1.0
Mutation probability : 0.005
Max. No. of generations : 80

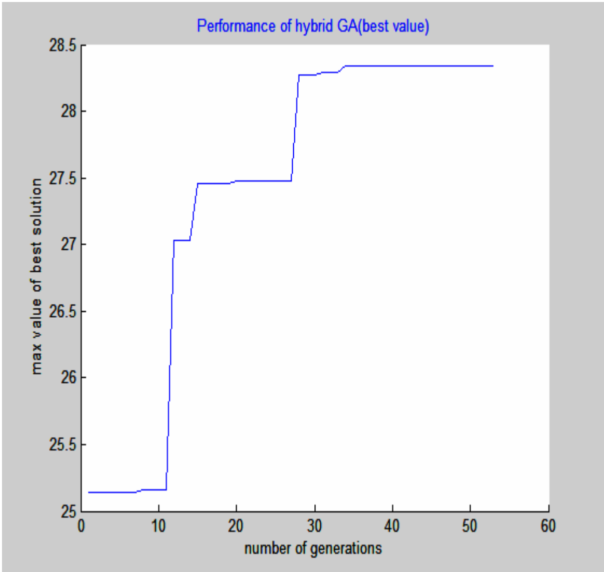

Fig. 1. Performance of hybrid genetic algorithm with respect to number of generations and max value of best solution

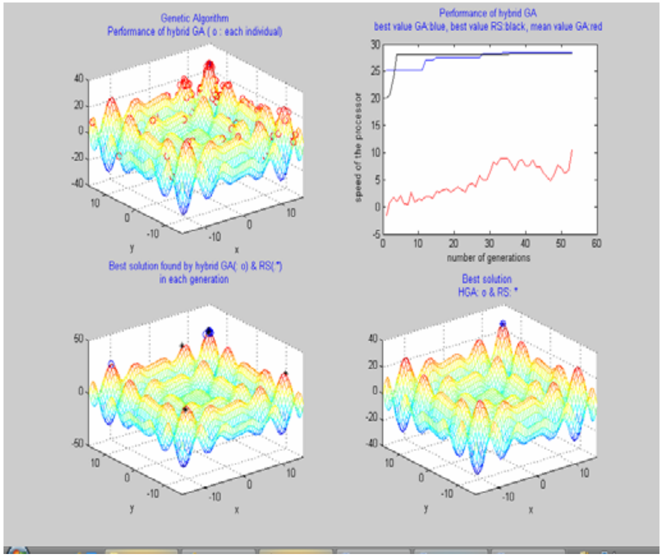

Fig. 2.
1. Performance of hybrid genetic algorithm (with each individual task – red balls)
2. Performance of hybrid genetic algorithm compared to with local search and random search considering speed of processor
3. Best solution found by hybrid genetic algorithm along with random search in each generations over 40 generations.
4. Best solution found by HGA and Random Search over 40 generations .

It is obvious from the Fig. 1 & Fig. 2, that the optimal or near optimal solution can be found using the proposed model Referring to above figure 1, it can be easily seen that the algorithm finds best solutions after 40 generations while using relatively small population. Hence relatively small no. of generations can easily find the best solution faster compared to random search.

VI. CONCLUSIONS

In this paper, a novel representation for the task scheduling Problem in GCE using Hybrid genetic algorithm, based on local search operator – random walk model for reducing no. of generations is proposed..

The experimental results illustrates that in the scheduling problem, the route is found in a few generations with population of arrived tasks .

The results and demonstration proves that the solution found using a hybrid G.A is optimal.

VII. FUTURE WORK

Scheduling of tasks onto processors, considering dynamic load of processor, as well as pre-emption of tasks, job –migration will be the future work.

VIII.REFERENCES

[1] Hisao Ishibuchi*, *Member*, *IEEE*, Tadashi Yoshida*, and Tadahiko Murata**, *Member*, *IEEE*, Balance between Genetic Search and Local Search in Memetic Algorithms for Multiobjective Permutation Flowshop Scheduling

[2] Neda Shahidi, Hadi Esmaeilzadeh, Marziye Abdollahi, Caro Lucas ,Memetic Algorithm Based Path Planning for a Mobile Robot, International journal of information technology volume 1 NUMBER 4 2004 ISSN:1305-239X

[3] Jing Tang, Meng Hiot Lim, Yew Soon Ong and Meng Joo Er, parallel genetic algorithm with selective local search for large scale quadratic assignment problems.

[4] Goldberg, D. E., Genetic Algorithms in Search, Optimization and Machine Learning, MA: Addison-Wesley:

[5] Digalakis, J. G. and K. G. Margaritas, A performance comparison of parallel genetic and memetic algorithms
Using MPI, IntRep03, University of Macedonia, Parallel Distributed Processing Laboratory, Thessalonica, Greece, 2000.

[6] Hart W. E.: Adaptive Global Optimization with Local Search. Ph. D. Thesis, University of California, San Diego (1994).

[7] Shahidi N., Esmaeilzadeh H. Abdollahi M., Lucas C.: Self-adaptive genetic Algorithm: An Adaptive Conjugate gradient approach. IEEE Conference of Cybernetic and Intelligent Systems (CIS'2004), in press.

[8] Gosia Wrzesinska Jason Maassen Henri E. Bal, Self-adaptive applications on the Grid , 2005 ACM

[9] Helene N. Lim Choi Keung, Justin R.D. Dyson, Stephen A. Jarvis, Graham R. Nudd , Performance Modeling of a Self-adaptive and Self-optimizing Resource Monitoring System for Dynamic Grid Environments.

[10] Yang Gaoa,* , Hongqiang Rongb, Joshua Zhexue Huangc , Adaptive grid job scheduling with genetic algorithms , Future Generation Computer Systems 21 (2005) 151–161

[11] Krasnogor N.: A genetic Algorithm with Self-adaptive Local Search; TSP as a Case Study. In Proc. of the 2000 International Genetic and Evolutionary Computation Conference (GECCO 2000).

[12] Land, M. W. S.: Evolutionary Algorithms with Local Search for Combinatorial Optimization. Ph. D. Thesis, University of California, San Diego, 1998.

[13] ems. Jone WTang, J., M. H. Lim and Y. S. Ong, A parallel hybrid GA for combinatorial optimization using
grid technology, Proc. of the Congress on Evolutionary Computation, Canberra, Australia, vol.3,pp.1895-1902, 2003.
, Inc., 2000.

[14] Sathish S. Vadhiyar1,_ and Jack J. Dongarra2,3Self Adaptively in Grid Computing Exper. 2004; 00:1{26}

[15] zhiming liu,Jiliu Zhou, Su Lai
New Adaptive Genetic Algorithm Based On Ranking, Proceedings of the Second International Conference on Machine Learning and cybernetics, X.'an, 2-5 November 2003

[16] Vincenzo Di Martino , Scheduling in a Grid computing environment using Genetic Algorithms ,Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS.02)

[17] Yew-Soon Ong, Meng-Hiot Lim, Ning Zhu, and Kok-Wai Wong , Classification of Adaptive genetic Algorithms:
A Comparative Study , IEEE transactions on systems Transactions on systems , Man, And Cybernetics—PART B: cybernetics , Vol. 36, NO. 1, February 2006 141

[18] E. K. Burke and A. J. Smith.. In Proceedings of the International Conference on Neural Information Processing and Intelligent Information Systems, volume 1, pages 469–472. Springer, 1997., A genetic Algorithm to Schedule Planned Grid Maintenance

[19] E. K. Burke and A. J. Smith. A memetic algorithm for the maintenance scheduling problem. In Proceedings of the International Conference on Neural Information Processing and Intelligent Information Systems, volume 1, pages 469–472. Springer, 1997.

[20] Seonho Kim and Jon B. Weissman ,
A GA-based Approach for Scheduling Decomposable Data Grid Applications

IX. BIOGRAPHIES

Uma Gurav is working as a Lecturer at Dr. D.Y. Patil College of engineering, Mumbai University, India. She has received B.E, Degree in Computer Science and Engineering from Mumbai University and currently pursuing M.TECH at Dr. Babasaheb Technological University, Lonere, India. Her main research Interest includes Genetic Algorithms , Search and optimization techniques and distributed systems..

Laxman D. Netak is working as a assistant professor with Dr. Babasaheb Ambedkar Technological University, Lonere, India.

His main research interest includes grid computing, computer networks, optical fiber networks and distributed systems.