Efficient Job Scheduling on Computational Grid with Differential Evolution Algorithm

S.Selvi, Dr. D.Manimegalai and Dr.A.Suruliandi

***Abstract*—Grid computing refers to the combination of computer resources from multiple administrative domains to reach common goal. Grids offer a way of using the information technology resources optimally inside an organization. Grid environments facilitate distributed computation. Hence the scheduling of grid jobs should be considered as an important issue. This paper introduces a novel approach based on Differential Evolution algorithm for scheduling jobs on computational grid. The proposed approach generates an optimal schedule so as to complete the jobs within a minimum period of time and utilizing the resources efficiently.**

***Index Terms*—Differential Evolution Algorithm, Grid computing, Job scheduling, Makespan, Optimization**

I. Introduction

Grid computing is a form of distributed computing that involves coordinating and sharing computing, application, data and storage or network resources across dynamic and geographically dispersed organization [1].Grid technologies promise to change the way organizations tackle complex computational problems. Grid computing is an evolving area of computing where standards and technology are still being developed to enable this new paradigm.

Users can share grid resources by submitting computing tasks to grid system. The resources of computational grid are dynamic and belong to different administrative domains. The participation of resources can be active or inactive within the grid. Hence, it is impossible for anyone to manually assign jobs to computing resources in grids. Therefore grid job scheduling is one of the challenging issues in grid computing. Grid scheduling system selects the resources and allocates the user submitted jobs to appropriate resources in such a way that the user and application requirements are met.

There are many research efforts aiming at job scheduling on the grid. Scheduling m jobs to n resources with an objective to minimize the total execution time has been shown to be NP-complete [2]. Therefore the use of heuristics is the defacto approach in order to cope in practice with its difficulty. Krauter *et al.* provided a useful survey on grid resource management systems, in which most of the grid schedulers such as AppLes, Condor, Globus, Legion, Netsolve, Ninf and Nimrod use simple batch scheduling heuristics [3]. Jarvis *et al.* proposed the scheduling algorithm using metaheuristics and compared FCFS with genetic algorithm to minimize the makespan and it was found that metaheuristics generate good quality schedules than batch scheduling heuristics [4]. Braun et al. studied the comparison of the performance of batch queuing heuristics, tabu search, genetic algorithm and simulated annealing to minimize the makespan [5]. The results revealed that genetic algorithm achieved the best results compared to batch queuing heuristics. Hongbo Liu *et al.* proposed a fuzzy particle swarm optimization (PSO) algorithm for scheduling jobs on computational grid with the minimization of makespan as the main criterion [6]. They empirically showed that their method outperforms the genetic algorithm and simulated annealing approach. The results revealed that the PSO algorithm has an advantage of high speed of convergence and the ability to obtain faster and feasible schedules.

In this paper, we address a job scheduling problem on computational grid, in which minimization of execution time is considered as the objective. To tackle this problem, Differential Evolution algorithm is proposed to search for the optimal schedule which in turn gives the solution to complete the batch of jobs in minimum period of time.

The rest of the paper is organized as follows. Section 2 presents the problem statement related to job scheduling. In Section 3, background of DE algorithm is described and the proposed algorithm is outlined. The computational results are reported in Section 4 and the conclusions are presented in Section 5.

II. Problem Statement

Scheduling is the process of mapping the jobs to specific time intervals of the grid resources. The grid job scheduling problem consists of scheduling m jobs with given processing time on n resources. Let J_j be the independent user jobs, j = {1, 2, 3…m}. Let R_i be the heterogeneous resources, i = {1, 2, 3…n}. The speed of each resource is expressed in number of cycles per unit time (CPUT). The length of each job is expressed in number of cycles. The information related to job length and speed of the resource is assumed to be known based on user supplied information, experimental data and application profiling or other techniques [7].

The objective of the proposed job scheduling algorithm is to minimize the makespan. Makespan is a measure of the

Manuscript received June 14, 2010

S. Selvi, Department of Electronics and Communication Engineering, Dr.Sivanthi Aditanar College of Engineering, Tiruchendur, Anna University Tirunelveli, India (email: mathini31@yahoo.co.in).

Dr.D.MANIMEGALAI, Department of Information Technology, National Engineering College, Kovilpatti, Anna University Tirunelveli, India (email: megalai_nec@yahoo.co.in).

Dr.A.Suruliandi, Department of Computer Science and Engineering, Manonmaniam Sundaranar University, Tirunelveli, India (email: suruliandi@yahoo.com).

throughput of the heterogeneous computing system. Let $C_{i,j}$ ($i \in \{1,2,...n\}, j \in \{1,2,...m\}$) be the completion time that the resource R_i finishes the job J_j, $\sum C_i$ represents the time that the resource R_i finishes all the jobs scheduled for itself. Makespan is defined as $C_{max} = \max \{\sum C_i\}$ [6].

To address the problem, we start with the following assumptions.

1) Any job Jj has to be processed in resource Ri until completion.
2) Jobs come in batch mode.
3) All jobs and grid resources are submitted at a time while start processing each batch.

III. Scheduling using Differential Evolution (DE) Algorithm

A. Previous work using DE algorithm

Differential Evolution is a novel population based evolutionary algorithm, which has been proposed for optimizing complex problems over a continuous domain. DE searches for the global optima by utilizing differences between contemporary population members, which allows the search behavior of each individual to self-tune. So far, DE has attracted much attention and wide applications in a variety of fields [8] [9].

Onwubolu et al. addressed the flow-shop scheduling problem using DE algorithm [10].In their work, the algorithm was implemented by mapping Job/Machine sequence to real numbers for DE operations. Since this approach is not feasible in the case of grid scheduling, Talukder et al. proposed a workflow execution planning approach using Multi objective Differential Evolution to generate trade-off schedule by considering the completion time of tasks and the total execution cost of jobs, in which they dealt with exact scheduling sequences [11]. Our approach makes use of integer values in order to map the resource/job sequence.

B. Differential Evolution algorithm

The differential evolution algorithm (DE) introduced by Storn and Price is a novel parallel direct search method, which utilizes NP parameter vectors as a population for each generation G. DE is a kind of evolutionary optimization algorithm. There are several variants of DE available [12]. This paper makes use of the DE/rand/1/bin scheme.

It starts with the random initialization of the initial population of NP individuals. Each individual has an n dimensional vector. The ith individual at generation't' can be represented as $X_i^t = [x_{i,1}^t, x_{i,2}^t, ..., x_{i,n}^t]$.

According to the mutation operator, a mutant vector is generated by adding the weighted difference between two randomly selected target population individuals to a third individual as follows.

$$V_i^t = X_a^t + F * \left(X_b^t - X_c^t\right) \qquad (1)$$

Where $a, b, c \in (1,2,...NP)$ are randomly chosen and mutually exclusive. $F \in [0, 1]$ is the scaling factor which affects the differential variation between two individuals. After the mutation phase, the cross over operator is applied to obtain the trail vector $U_i^{t+1} = [u_{i,1}^{t+1}, u_{i,2}^{t+1}, ... u_{i,n}^{t+1}]$ by the following equation:

$$u_{i,j}^{t+1} = \begin{cases} v_{i,j}^t, if \quad rand_j \le CR \quad or \quad j = randn_i \\ x_{i,j}^t \qquad\qquad , otherwise \end{cases} \qquad (2)$$

Where $rand_j$ is the jth independent random number uniformly distributed in the range of [0, 1].Also $randn_i$ refers to a randomly chosen index from the set {1, 2…n}.CR is a user defined cross over factor in the range [0, 1].

Following the crossover operation, to decide whether or not the trail vector U_i^{t+1} should be a member of the population of the next generation, it is compared with the target individual X_i^t.

Finally the selection is based on the survival of the fitness as follows.

$$X_i^{t+1} = \begin{cases} U_i^{t+1}, \quad if \quad fit\left(U_i^{t+1}\right) < fit \left(X_i^t\right) \\ X_i^t \ , otherwise \end{cases} \qquad (3)$$

C. The proposed Job Scheduling Algorithm

1) General scheme of DE based grid job scheduling algorithm

The pseudo code for DE based grid job scheduling algorithm is illustrated in Algorithm 1. Table I depicts the explanation of abbreviated parameters used in Algorithm 1.

Algorithm 1 *DE based Grid Job Scheduling Algorithm*

```
Define RT, JT, ESR, JL, F, CR, NP, MaxIter, STR, ETR
  Create the initial population of random individuals
  Check the feasibility of initial population vectors
for 1 to MaxIter
  Calculate the makespan of each individual
for i = 1 to NP
  Select random integer randn_i ∈ (0, 1, 2... JT)
  Select mutually exclusive random
  individuals X_a, X_b and X_c
  Calculate mutant vector V according to
  equation (1) starting from the position
  randni of each individual.
  Select the random value randj ∈ [0, 1]
  Calculate the trail vector U_i according to
  equation (2)
  Check the feasibility of trail vector Ui
end for
  Calculate the makespan of trail vector set
  for i = 1 to NP
  if makespan of U_i is less than X_i then
  Select U_i
  else
  Retain X_i
end if
end for
  Record the solution with minimum makespan
end for
```

TABLE I PARAMETERS USED IN ALGORITHM 1

RT	Total Resources	CR	Crossover Factor
JT	Total Jobs	NP	Population Size
ESR	Execution speed of Resource	MaxIter	Maximum number of Iteration
JL	Job length	STR	Start time of resource engaged in grid
F	Scaling factor	ETR	End time of resource engaged in grid

2) *Solution Representation*

In the proposed scheduling algorithm, the solution is represented as an array of length equal to the number of jobs. The value corresponding to each position i in the array represent the resource to which job i was allocated. The job-to-resource representation for the resource job pair(3,13) is illustrated in Fig. 1.

2	1	2	3	1	2	3	1	2	3	2	1	1

Fig.1. Job-to-resource representation for the grid job scheduling problem

Grid resource 1	J2	J5	J8	J12	J13
Grid resource 2	J1	J3	J6	J9	J11
Grid resource 3	J4	J7	J10		

Fig. 2. Mapping of jobs with grid resource

The first element of the array denotes the first job (J1) in a batch which is allocated to the Grid resource 2; the second element of the array denotes the second job (J2) which is assigned to the Grid resource 1, and so on
(see Fig. 2)

IV. SIMULATION ON DE BASED GRID JOB SCHEDULING ALGORITHM

A. *Experimental setup*

The performance of the proposed scheduling algorithm was tested for the resource job pairs of small scale problem (3,13) and large scale problems such as (5,100), (8,60) and (10,50). The numerical simulations are carried out with the dataset used and tested in the paper[6].

The DE based grid job scheduling algorithm is coded in MATLAB R2008b and experiments are executed on a Pentium IV 2.67 GHz PC with 512 MB memory.

B. *Parameter setting*

The selection of control parameters is the key function of the Differential Evolution algorithm. The Scaling Factor F controls the amplification of differential variations. CR is a real valued cross over factor which controls the probability of selection of trail vector. Both F and CR affect the convergence rate and robustness of the search process. Their optimal values are dependent both on objective function characteristics and on the population size NP. Suitable values for F and CR were found by implementing extensive experiments with different sets of parameters [10].

The nature of convergence of DE algorithm had been observed for all kind of problems by few tests. After that, the number of iterations had been fixed as 400. During experimentation, it was found that the algorithm converges within less number of iterations for small scale problem when it was tested with population size as 25 times of number of jobs. But the execution time was found to be high for large scale problem with such population. Hence, the population size was set as 10 times that of the number of jobs employed which produces the optimal schedule. In the proposed algorithm, the parameters are set as specified in Table II.

C. *Test Result and Comparison*

The experiment for each problem was run 10 times with different initial random population. Each run had a fixed number of 400 iterations. The average makespan value and the standard deviation of 10 different runs were recorded. The time taken for the algorithm to produce the scheduling solution was also monitored. The performance of the proposed algorithm is compared with the results of Fuzzy Discrete PSO Scheduling algorithm proposed in the latest paper, which has been accepted as one of the best in the literature [6].

Table III shows the comparison between the DE algorithm developed in the present study discussed in this paper and the Fuzzy Discrete PSO Scheduling algorithm developed in a previous study for grid scheduling problem [6].

TABLE II PARAMETER SETTINGS FOR DIFFERENTIAL EVOLUTION ALGORITHM

Parameter	Value
Population Size (NP)	10 * No. of jobs
Cross over factor (CR)	0.011
Scaling Factor (F)	0.01 – 0.015
Number of Iterations (MaxIter)	400

1) *Results for the Resource job pair (3, 13)*

The scaling factor was set as 0.011. The average makespan is found to be 46.05 and the performance for the pair (3, 13) is illustrated in Fig.3. The algorithm takes 22.44 seconds to produce the scheduling solution which is found to be a good result when compared with PSO algorithm.

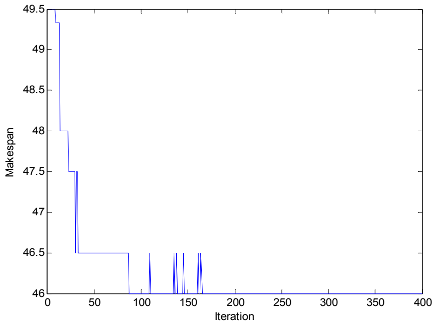

Fig. 3 Performance for the resource job pair (3,13)

2) *Results for the Resource job pair (5,100)*

The performance of the scheduling algorithm for the pair (5,100) based on DE approach is competitive when compared with PSO algorithm. The scaling factor was set as 0.01 to study the nature of convergence. The makespan response for the population of 1000 is shown in Fig.4.

As the completion time of DE algorithm for the pair (5,100) is almost 3.5 times that of the pair (8, 60), the algorithm was studied by conducting more number of runs and the reason was found to be the size of the population. As the number of job increases, the population size is also multiplicative. Hence, the algorithm was run with the population size of 500. The experiment yields the best result, which is illustrated in Table IV and in Fig.5.

TABLE III Comparison between Differential Evolution (DE) and Particle swarm optimization (PSO) [6].

Parameter	Resource job pair							
	(3,13)		(5,100)		(8,60)		(10,50)	
Average Makespan	DE	PSO	DE	PSO	DE	PSO	DE	PSO
	46.0500	46.2667	86.3645	84.0544	42.4819	41.9489	38.3900	37.6668
Standard deviation	± 0.1581	± 0.2854	0	± 0.5030	0	± 0.6944	± 0.2000	± 0.6068
Completion time- seconds	22.4400	106.2030	1550.3271	1485.6000	430.0000	1521.0000	285.2600	1585.7000

TABLE IV Performance of DE Algorithm for (5,100) with NP=500

Problem	Average Makespan	Standard Deviation	Completion Time(sec)
(5,100)	86.48	0.073	570

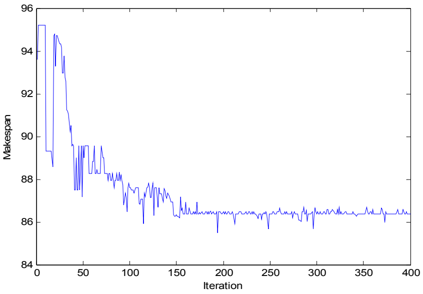

Fig.4 .Performance for the resource job pair (5,100)

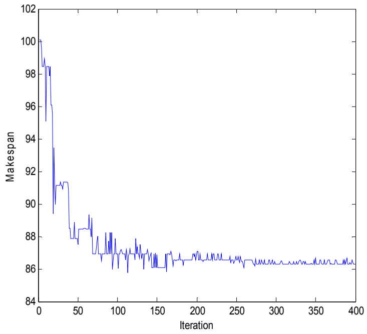

Fig. 5 Performance for (5,100) with NP=500

3) *Results for the Resource job pair (8, 60)*

The empirical results of the scheduling problem for the pair (8,60) are illustrated in Table III and in Fig.6. The scaling factor was set as 0.009. The average makespan of DE algorithm for the pair (8, 60) is slightly higher than the PSO approach but the standard deviation and the completion time outperform the PSO algorithm.

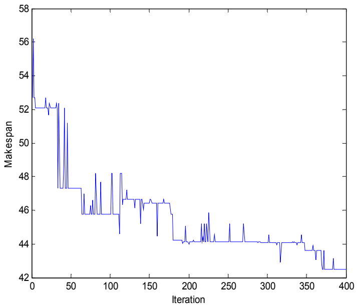

Fig.6.Performance for the resource job pair (8,60)

4) *Results for the Resource job pair (10, 50)*

The grid scheduling problem was run with the scaling factor of 0.015.The experimental result is illustrated in Fig.7. Although the PSO approach yields less average makespan than DE algorithm, the DE algorithm spends much less time to complete the scheduling process with less standard deviation.

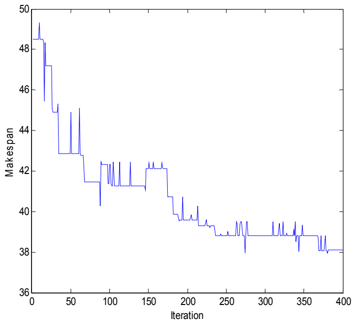

Fig.7.Performance for the resource job pair (10,50)

5) *Effect of Resource Utilization*

The performance of scheduling algorithm depends on the factor that how effectively the resources have been utilized.

The Resource Utilization for the problems (3,13), (5,100), (8,60) and (10,50) is illustrated in Fig.8, Fig.9, Fig.10 and Fig.11 respectively.

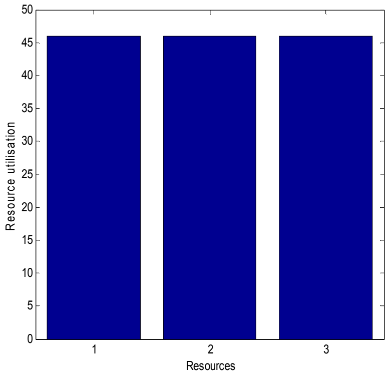

Fig.8. Resource Utilization for the problem (3,13)

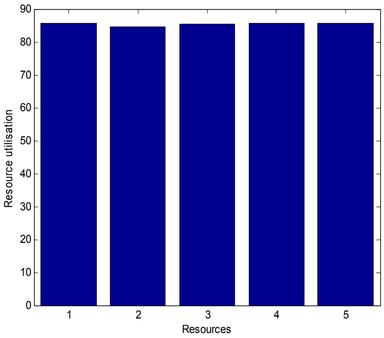

Fig.9. Resource Utilization for the problem (5,100)

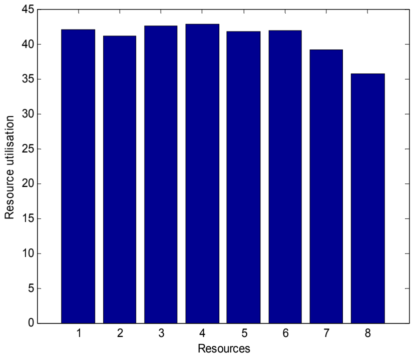

Fig.10. Resource Utilization for the problem (8,16)

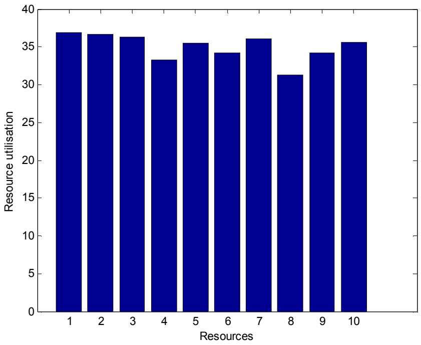

Fig.11. Resource Utilization for the problem (10,50)

V. Conclusion

This paper presents a novel job scheduling approach based on DE algorithm to solve grid scheduling problem to minimize the completion time. The proposed scheduling algorithm is very simple, as it involves small number of parameters for devising the algorithm. As the status of resource is dynamic within the grid environment, it is necessary to produce the faster and feasible schedules. The experimental result shows that DE based grid scheduling approach is capable of generating the solution within a minimum period of time. Simulation results and comparisons based on a set of problem demonstrated the efficiency and effectiveness of our proposed approach. In our future work, it is proposed to develop adaptive DE based algorithms and generalize the DE-based algorithm to multi-objective complex scheduling problems and stochastic scheduling problems.

Acknowledgment

We would like to thank the Professors H.Liu, A.Abraham, A.E.Hassanien for providing the datasets to implement and test the algorithm proposed in this paper.

References

[1] I.Foster ,C.Kesselmann,(Eds.), The Grid: Blueprint for a New Computing Infrastructure, Morgan Kaufmann Publishers, USA, 1999.

[2] O.H.Ibarra and C.E.Kim, Heuristic algorithms for scheduling independent tasks on nonidentical processors, J.ACM 24, 2 (Apr. 1977), pp. 280-289.

[3] K.Krauter, R.Buyya, M.Maheswaran, A taxonomy and survey of grid resource management systems for distributed computing, Software-Practice and Experience, 32:135-164, DOI: 10,1002/spe.432,2002

[4] S.A.Jarvis, D.P.Spooner, H.N.Lim Choi Keung,G.R.Nudd, J.Cao, S.Saini, Performance prediction and its use in parallel and distributed computing systems. In the Proceedings of the IEEE/ACM International Workshop on Performance Modelling,Evaluation and optimization of Parallel and distributed systems,Nice,France.2003

[5] T.D.Braun, H.J.Siegel, N.Beck, D.A.Hensgen, R.F.Freund, A comparison of eleven static heuristics for mapping a class of independent tasks on heterogeneous distributed systems,Journal of Parralel and Distributed Computing,2001,pp.810- 837

[6] H.Liu, A.Abraham, A.E.Hassanien, Scheduling jobs on computational grids using a fuzzy particle swarm optimization algorithm, Future Generation ComputerSystems(2009),doi:10, 1016/j.future.2009.05.022

[7] S.K.Garg, R.Buyya, H.J.Siegel, Time and cost trade-off management for scheduling parallel applications on Utility Grids, Future Generation Computer Systems (2009), doi:10.1016/j.future.2009.07.003

[8] F.P.Chang, C.Hwang,Designofdigital PID controllers for continuous time plants with integral performance criteria,Journal of the chineese Institute of Chemical Engineers, 35(6), 2004, pp.683-696.

[9] Y.P.Chang, C.J.Wu, Optimal multiobjective planning of large-scale passive harmonic filters using hybrid differential evolution method considering parameter and loading uncertainty, IEEE transactions on Power Delivery, 20(1), 2005, pp.408-416.

[10] G.Onwubolu, D.Davendra, Discrete Optmization Scheduling flow shops using differential evolution algorithm, European Journal of Operational Research 171 2006, pp. 674-692

[11] A.K.M.K.A.Talukder, M.Kirley, R.Buyya, Multiobjective Differential Evolution for scheduling workflow applications on global Grids, Concurrencyand Computation:Practice and Experience, published online in Wiley Interscience, DOI: 10, 1002/cpe.1417 (2009)

[12]] K.Price, R.Storn, Differential evolution (DE) for Continuous function optimization, 2007.http://www.icsi.berkeley.edu/%7Estorn/code.html

S.Selvi was born at Tirunelveli, Tamilnadu, India in 1975. She is an assistant professor of Electronics and Communication Engineering, Dr.Sivanthi Aditanar College of Engineering, Tiruchendur, India. She received her B.E. and M.E in Electronics and Communication Engineering and Computer Science Engineering respectively from Manonmaniam Sundaranar University, Tirunelveli ,India. Her research interests include Digital Signal Processing, Distributed Computing and Grid Computing

Dr.D.Manimegalai had her BE & ME from Government College of Technology, Coimbatore and PhD from Manonmanium Sundaranar University, Tirunelveli. She worked in PSNA College of Engineering, Dindigal from 1985 – 1987. Since 1987, she has been working in National Engineering College in various positions. She has modest number of research publications including journals such as AMSE, Elsevier and Pattern Recognition letter and in National and International Conferences. She is also recognized supervisor for guiding Ph.D students by various universities. Currently 8 scholars are pursuing research under her supervision. She had attended number of Seminars, Workshops, Faculty Development Programmes and Conferences. Also she had organized workshops and conferences. Recently she had organized many AICTE, DST, DRDO and CSIR Sponsored National Seminars, workshops and conferences. .Her Current area of research interests includes Medical Image Processing and Data Mining and Image Retrieval. She is a life member of Computer Society of India, System Society of India and Indian Society for Technical Education and fellow member in Institution of Engineers.

Dr.A.Suruliandi was born at Theni, Tamilnadu, India in 1966. He is an associate professor in the Computer Science and Engineering Department at Manonmaniam Sundaranar University, Tirunelveli ,India. He received his B.E Degree from Bharathiyar University in 1987 , M.E and Ph.D. degree from Manonmaniam Sundaranar University, Tirunelveli ,India in 2000 and 2009 respectively. He has academic experience of 22 Years. He has published 16 research papers in National and 17 research papers in International conferences. His research interests include Digital Image Processing, Pattern Recognition and Remote Sensing