A Hyper Heuristics Technique for Data Partitioning and Scheduling to Heterogeneous Systems using Genetic Algorithm and Improved Particle Swarm Optimization

Sundar Ganesh, R.Sivakumar, N.Rajkumar

ABSTRACT---Development of the load partitioning for multiple round load distribution and effective scheduling of partitioned load to heterogeneous processor is primary goal of distributed and parallel system. In this paper, we propose hyper heuristics scheduling algorithm for load partitioning using genetic and improved particle swarm optimization techniques.A communication model is used to predict the optimal activation order, optimal number of processor and optimal number of rounds of the load. Heuristics Based Scheduling Algorithm is proposed using Hyper Heuristic Scheduling which is used to find the candidate solution (low level heuristic) to form Scheduling Solutions (heuristics algorithms) for large scale system with diversity operator as sequence dependent and sequence independent scheduling. For this solution, processing time of the entire processing load will be reduced. Hybrid Real Code genetic algorithm(HRGA) computes optimal activation order with cross over and mutation operator without considering the processor latency and different types of variation in the perturbation parameters. In order to optimize this issue, we utilizeImproved Particle swarm optimization (IPSO) determine the load fraction for generating activation order in terms of dynamically predicting fitness value of the processor with certain number. The Simulation analysis demonstrates the proposed model performance in terms of mean, standard deviation, computational complexity and Average Execution Time comparing against hybrid real coded genetic algorithm.

Keywords: Data partitioning, Processor Scheduling, Improved Particle Swarm Optimization, Real Coded Genetic Algorithm, Hyper Heuristic

1. INTRODUCTION

Data Partitioning is the main feature to obtain effective scheduling on available processor in the parallel and distributed system. To reach the minimum processing time on the load execution by several strategies using data partitioning. To improve the performance of load balancing on the multiple processors is scheduling [2].With new developments in soft computing algorithms, many loads scheduling technique has been utilized in the recent years. Divisible load scheduling [3], Bandwidth centric scheduling strategies [4] was discussed as it acts as effective approach. The load balancing techniques and scheduling techniques used in parallel and distributed system are still NP hard [4][9].To determine the polynomial time for the load partitioning and scheduling, affine model of communication [5][8] is considered which include the communication latency towards formulizing the effective data partitioning and processor scheduling tool.

A Method for efficient distribution of work load between different processing elements in heterogenous systems is presented. [6].An efficient technique to find the optimal number round of load distribution to create the optimal activation order for execution on optimal number of processor[7][10]. The optimal activation order needs to be computed based on the optimal load fraction and processing time.

The load distribution among different processors using improved PSO techniques has presented[9]. Hyper Heuristic Scheduling which is used to find the candidate solution (low level heuristic) to form Scheduling Solutions (heuristics algorithms) for large scale system with diversity operator as sequence dependent and sequence independent scheduling is considered as rule based scheduling[11]. Heuristics based Scheduling with heuristic selection will reduce the processing time of the load in scheduling and execution[12].Initially Hybrid Real Code genetic algorithm is been processed on the activation order to computes optimal activation order with cross over and mutation operator without considering different types of variation in the perturbation parameters. The Proposed model integrates the several Heuristics. By the technique of Improved PSO, load distribution strategies are utilised to predict dynamically the fitness value of theprocessor.

2. RELATED WORK

There exists many predominal approaches related to the objective of the proposed research, the similar approaches utilized in the existing is as follows

Revised Manuscript Received on August 19, 2019.

Sundar Ganesh C S, Assistant Professor, Department of EEE, Karpagam College of Engineering, Coimbatore,T.N, India(E-mail: sundarganesh.cs@kce.ac.in)

Dr.R.Sivakumar, Professor, Department of Mechatronics Engineering, Akshaya College of Engineering, Coimbatore,T.N, India(E-mail: sivakumarr@acetcbe.edu.in)

Dr.N.Rajkumar, Associate Professor, Department of Computer Science and Engineering, Akshaya College of Engineering, Coimbatore, T.N, India (E-mail: rajkumarn@acetcbe.edu.in)

2.1.Real Code Genetic Algorithm for Data Partitioning and Scheduling

A hybrid real-coded GA is used to determine the population for the load distribution. The population generation search creates the set of participating processors, and in every round the load fractions assigned to them, such that the minimum processing time, for a given activation order. Utilizing the load fractions obtained for each activation order, can further taken to predict a better activation order on the next round of the load distribution to the processor. Real-coded GA uses hybrid crossover and mutation operators.

3. PROPOSED MODEL

In this model, effective data partitioning and scheduling will be discussed utilizing the HRGA and IPSO.

3.1. Processor Scheduling Problem

The processor scheduling problem occurs from a single tasking in a single processor to multitasking in multi processors. The Optimal Load Activation order for n rounds and No of participating processor can be defined as follows

Load Activation Order isT= {$T_{1,}$ T2....T_N} (1)

No of Participating Processor P= {P1, P2, P3...Pn} (2)

The makespan that indicates the completion time of the last task is the most important measurement in scheduling problem.

3.2. Scheduling Objective Condition

The objective of the data partitioning or scheduling is to calculate the size of the fractions of load assigned to each of the processor in each rounds, such that the minimum processing time of the entire processing load. The Scheduling objective has to consider both constrainst such as with latency or without latency.

Load Fraction is represented by A_o

Scheduling for processor depends on Load Fraction and Processing rate

Scheduling of P = A_o*P_R

Where

Processing rate is reciprocal of the speed which is represented as P_{R1}

3.3. Load Distribution Condition

The load distribution depends on the Finish time of initial round of the load and its processing time of processor to the load fraction assigned by load activation order. The primary condition for load distribution in the divisible load theory is processor should stop computing at the same time instant to achieve the minimum processing time.Load distribution is carried out using the affine model of communication is shown in Figure 1

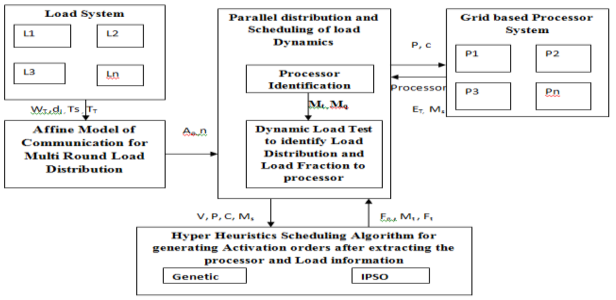

Load System
L1
L2
L3
Ln
W_T,d_I, Ts, T_T
Affine Model of Communication for Multi Round Load Distribution
A_o,n
Parallel distribution and Scheduling of load Dynamics
Processor Identification
M_t, M_q
Dynamic Load Test to identify Load Distribution and Load Fraction to processor
P, c
Processor
E_T, M_s
Grid based Processor System
P1
P2
P3
Pn
V, P, C, M_s
F_{ol}, M_t, F_t
Hyper Heuristics Scheduling Algorithm for generating Activation orders after extracting the processor and Load information
Genetic
IPSO

Figure 1: Architecture Diagram of the proposed model

3.3.1. Affine Communication model

Affine communication is used to predict the optimal activation order, optimal number of process and optimal number of rounds.In this model, processor is categorized into multiple sub processors with defined computation speed. Each processor is assigned with dynamic values of communication links with some processing speed. Latency of the processor for different communication links is included. The Load distribution is carried out using load fractions; load fraction grouping can be single round or multiple rounds. Multiple load fractions are combined as the activation order to processor.

The optimal activation order for the processor can be obtained from the load fractions for any activation order for a specified total processing load and for a specified number of rounds in affine model. The affine model to compute normal activation order will be based on the following Initial Round of Load & Total Load has to be defined

Strategy 1: Activation Order

If ((Sub processor 1 ==0) &&(Sub Processor 2==0))Assign the 1st and 2nd round loads to sub processor 1 and assign 1st and 2nd rounds of load to sub processor 2Compute the processing time of each processor Else Wait for the execution of the process in each sub processor

Strategy 2: Activation Order

If ((Sub processor 1 ==0) && (Sub Processor 2==0)) Assign the 1^s round of loads distribution to sub processor 1 and assign $1^{st\ and}$ 2nd rounds of load to sub processor 2Compute the processing time of each processor Else Wait for the execution of the process in each sub processor

Strategy 3: Activation Order

If ((Sub processor 1 ==0) && (Sub Processor 2==0))Assign the $1^{st\ and}$ 2nd round of loads distribution to sub processor 1 and assign 1st rounds of load to sub processor 2Compute the processing time of each processor ElseWait for the execution of the process in each sub processor

Strategy 4: Activation Order

If ((Sub processor 1 ==0) && (Sub Processor 2==0))Assign the 2ndround of loads distribution to sub processor 1 and assign $1^{st\ and}$ 2nd rounds of load to sub processor 2Compute the processing time of each processor Else Wait for the execution of the process in each sub processor

Strategy 5: Activation Order

If ((Sub processor 1 ==0) && (Sub Processor 2==0)Assign the 1st and 2rdround of loads distribution to sub processor 1 and assign 2nd rounds of load to sub processor 2Compute the processing time of each processor ElseWait for the execution of the process in each sub processor

Calculate the best activation order based minimum processing time using above computed strategies which is best activation order to the optimal participating resource and optimal rounds of load distribution. This condition can be applied to large scale system but it leads to computation complexity. In order to solve the computation difficulties, hyper heuristic scheduling optimization model will be utilized.

3.4.Hyper Heuristic Scheduling mechanism

Hyper Heuristics Algorithm is used to select the Optimal Heuristics for generating the optimal activation order, Processor and round of the load distribution. The hyper heuristics is generated using real coded genetic and Improved particle Swarm optimization.

3.4.1.Parameter Notations

The parameter notation and its description is mentioned in the table 1

Table 1: Hyper Heuristics Scheduling parameters

Notation	Description
W_T	Total Work Load
d_l	deadline of the each task
T_s	Task size
T_T	Task Type
A_o	Load Activation order
n	rounds of load distribution
M_s	Processor Size
B_w	Bandwidth
V	Velocity of the Task or Particle
P	Task or Task Partition
C	Task or Task Partition
Fn	Fitness function
F_t	Flow Time
M_t	Make span
E_T	Executed Task or particle or chromosome
M_t	Resource Characteristics or Resource Type
M_q	Resource Utilization or Resource queue

3.4.1. Scheduling using Real Coded Genetic Algorithm

Integration of two or more heuristic algorithms at each iteration generates the complementary advantages of these algorithms to find a best result than a single heuristic algorithm does. Real coded Genetic Algorithm and Hybrid real coded genetic algorithm generates the constant size population of the individual processor by using genetic operator like reproduction, cross over and mutations.

Each optimal activation order is composed of multirounds of load fractions (population). Population is generated randomly or using any divisible load theory. A load fractions assigned based on the fitness value of the processor or sub processor for dynamical load condition.

Reproduction operator in genetic operation transfers the load flow based on selected strategies to generate the heuristics. The Strategies is used to select the processor which have idle timeafter completion of the execution of the load in the round 1 and waiting for the load assignment to it from the next round will be assumed as idle time of the processor . It can be utilized for processing of the load if added to candidate pool of sub processor with processing speed.The available processor can be calculated using Taylor Equation, which is as follows

$$p^x = 1 + \frac{x}{n(n-l)} + \frac{x^2}{n(n-2)} + \frac{x^3}{n(n-3)} + \cdots, 0 < x < n-1 \tag{3}$$

The available processor is mapped effectively. The finish time computation may help the scheduler to map the processor for load assignment through any activation order. The formula to compute the flow time is as follows

$$\mathrm{T}^n = \sum_{k=0}^{n} a^k E^T \tag{4}$$

The utilization of Hybrid cross over operator and mutation operator will generate constrainst for available candidate pool to each load distribution. The genetic algorithm uses four cross over and mutation uses affine communication model for evaluation function modelling.Cross over generates next possible processor to the load fraction. Mutation uses the load fraction diversity from available processors.

Process –The Resource Scheduling

1. Initialize the processor to the activation order
2. Randomly select the activation order from the candidate pool to the load fraction
3. While Fitness criteria is not reached
   Update the processor of the solution through indexing & ranking using Reproduction operator

3.4.2.Scheduling optimization using Improved Particle Swam optimization

IPSO will be modelled as auto optimization strategy where genetic algorithm is as consider as resource scheduling algorithm. IPSO relies on the prediction of the Processor to load fraction through the choice of optimal activation order. Hyperparameter is selection is continuous optimization problem, hence we propose a improved optimization to the Load fraction assignment to the processor. The IPSO technique initializes the load fraction of the particle which acts as task, solution space composed of processor has to be optimized by updating the particle evolution. Each Particle is selected based local best and global best criteria's through calculation of velocity which reflects the operating speed of the task in the selected processor. The fitness value is modelled and termination condition for each looping. Inertia weight W shall be positive constant or positive linear/nonlinear function of time plays the important role of balancing global search and local search.

Process:

P

$\mathbf{P_{best}}$is the local Best position of all particles
$\mathbf{g_{best}}$is the global best position shared by all particles
Particles represented in the vector
No. of Particles – No .of Load
Search Space – No. of Processor
Each task (particle) treated as point in N dimensional Space
Each particle organizes it's the best solution based on fitness function is treated as p best
Any particle organized in the particular space based on fitness function is treated a g best

Particle is iterated to obtain the pbest and g best locations by random weight assignment at each iteration . Position Modification

$$V_i^{k+1} = wV_i^k + c_1\, rand_1\, (\dots)\ x\ (particlebest_i - s_i^k) + c_2\, rand_2\, (\dots)\ x\ (globalbest - s_i^k) \qquad (5)$$

Where
v_i^k: velocity of fraction of load i at the iteration k
w: weighting function of the PSO, and
c_j : weighting factor of the PSO distributed random number between 0 and 1
s_i^k : present position of load fraction i at iteration kparticle best$_i$: particle best of agent I of the PSOglobal best: global best of the group of the PSO

$$w = wMax-[(wMax-wMin)\ x\ iter]/maxIte \qquad (6)$$

wherewMax= initial weight,
wMin = final weight,
maxIter = maximum iteration number,
iter = current iteration number.

$$s_i^{k+1} = s_i^k + V_i^{k+}1 \qquad (7)$$

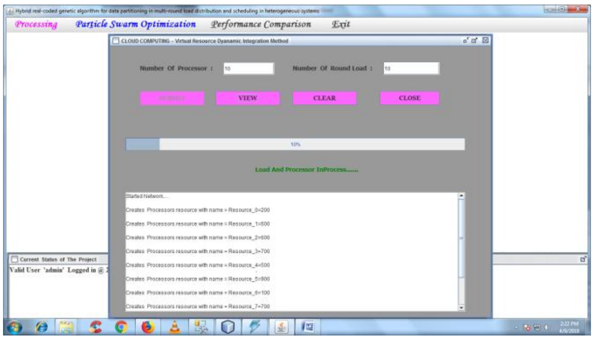
A large inertia weight facilitates a global search while a small inertia weight facilitates a local search.

4. IMPLEMENTATION AND RESULTS:

In this paper, the implementation is done by particle swarm optimization and it is compared with the hybrid PSO algorithm.

In this project 10 processors are used and 10 rounds are taken for Distributed scheduling.

The loads are separated and it is assigned a job id. For every load, it is assigned with start time and end time. Fig2. Shows the load distribution.


Hybrid real-coded genetic algorithm for data partitioning in multi-round load distribution and scheduling in heterogeneous systems
Processing
Particle Swarm Optimization
Performance Comparison
Exit
CLOUD COMPUTING – Virtual Resource Dyanamic Integration Method
VIEW
CLEAR
CLOSE
10%
Current Status of The Project

Number Of Processor : 10
Number Of Round Load : 10

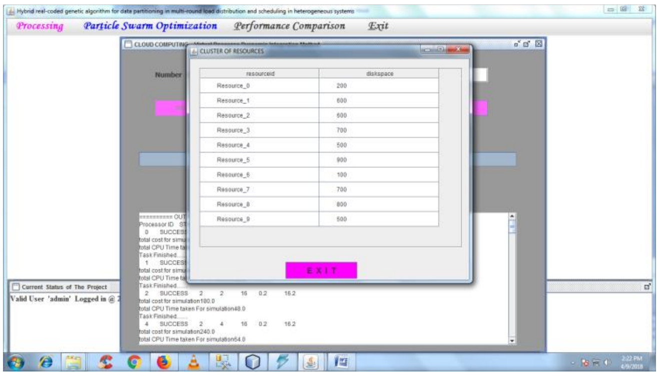
Load And Processor InProcess......

```
Started Network....
Creates  Processors resource with name = Resource_0=200
Creates  Processors resource with name = Resource_1=600
Creates  Processors resource with name = Resource_2=600
Creates  Processors resource with name = Resource_3=700
Creates  Processors resource with name = Resource_4=500
Creates  Processors resource with name = Resource_5=900
Creates  Processors resource with name = Resource_6=100
Creates  Processors resource with name = Resource_7=700
```

Valid User 'admin' Logged in @

Fig.2 Load Distribution

The processor is distributed into a cluster of resources is shown in Fig.3


Hybrid real-coded genetic algorithm for data partitioning in multi-round load distribution and scheduling in heterogeneous systems
Processing
Particle Swarm Optimization
Performance Comparison
Exit
CLOUD COMPUTING
CLUSTER OF RESOURCES
Number
EXIT
Current Status of The Project

resourceid	diskspace
Resource_0	200
Resource_1	600
Resource_2	600
Resource_3	700
Resource_4	500
Resource_5	900
Resource_6	100
Resource_7	700
Resource_8	800
Resource_9	500

```
========== OUT
Processor ID   ST
  0    SUCCES
total cost for simu
total CPU Time ta
Task Finished.....
  1    SUCCES
total cost for simu
total CPU Time ta
Task Finished.....
  2    SUCCESS    2     2     16    0.2     16.2
total cost for simulation180.0
total CPU Time taken For simulation48.0
Task Finished.....
  4    SUCCESS    2     4     16    0.2     16.2
total cost for simulation240.0
total CPU Time taken For simulation64.0
```

Valid User 'admin' Logged in @

Fig.3 Cluster of Resources

After the load and resources are scheduled by the affine model. The Processors are scheduled to different resources. The resources are clustered to some cluster. It is assigned is shown in Table 1

Sl.No	Cluster name	Resource Name
1	Cluster 1	Resource 1
2	Cluster 2	Resource 2
3	Cluster 3	Resource 5
4	Cluster 5	Resource 4
5	Cluster 5	Resource 6
6	Cluster 6	Resource 8
7	Cluster 6	Resource 9
8	Cluster 8	Resource 3
9	Cluster 9	Resource 7

Table 1 Cluster and Resource associations

The hybrid real coded GA and IPSO is applied and the performance values are compared. The performance comparisonof execution time of hybrid real coded GA and IPSO is shown in Fig.4

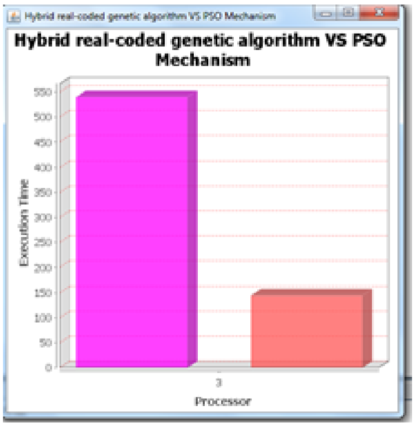


Fig4. Performance comparison chart of Execution timevs Processor

Retrieval Number: B11360882S819/2019©BEIESP
DOI:10.35940/ijrte.B1136.0882S819

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

The execution time is very high in genetic algorithm whereas in particle swarm optimization is very low. Hence the IPSO is a better algorithm for round distributed load scheduling problem. The performance comparison of throughput value is shown in Fig.5

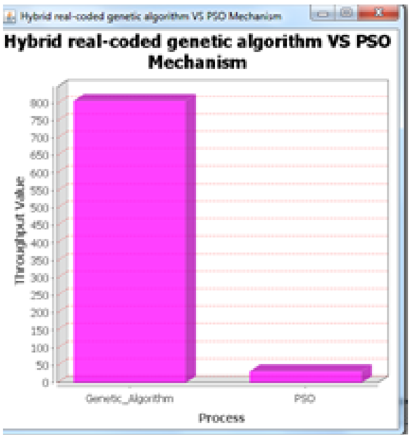


Fig.5 Comparison Chart of Through Put value

Through putvalue is very large in the hybrid genetic algorithm when compared to the particle swarm algorithm. The transfer unit value is shown in Fig.6

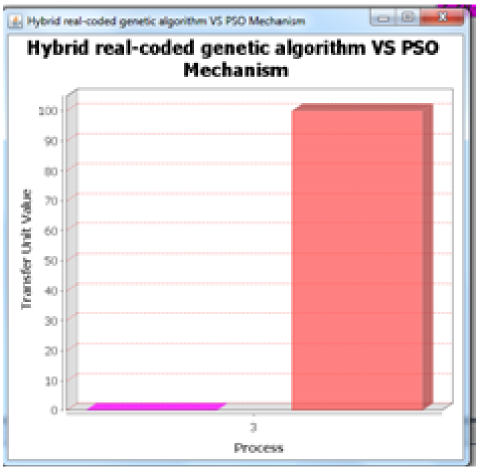


Fig.6 Performance comparison of transfer unit value

The Transfer unit value is very high in particle swarm optimization when compared to hybrid real coded genetic algorithm.

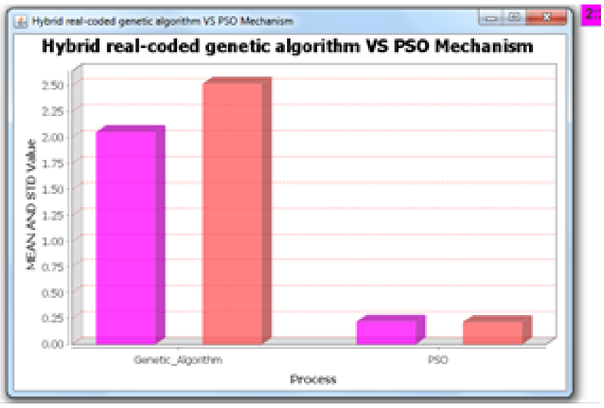


Fig.7 Performance Comparison of Mean and Standard Deviation

The mean and std value are greatly improved in PSO when compared to hybrid real coded genetic algorithm.

5. CONCLUSION

In this paper, improved particle swarm optimization techniques and hybrid real coded genetic algorithm is applied to multi round scheduling problem. First the load partitioning is done and it is scheduled. Affine communication model is used to find the optimal number of processor and optimal number of rounds of the load. The resources are allocated to a set of clusters. The improved Particle swarm optimization is proposed to determine the load fraction for predicting the fitness value of the processor with certain number. The Simulation analysis is demonstrated for the performance in terms of mean, standard deviation, Average Execution Time comparing against hybrid real coded genetic algorithm. The comparison shows improved particle swarm optimization is a better algorithm than hybrid real coded genetic algorithm in terms of execution unit, transfer value, mean and standard deviation.

REFERENCES:

1. Suresh, S., Huang, H., & Kim, H. J. (2014). Hybrid real-coded genetic algorithm for data partitioning in multi-round load distribution and scheduling in heterogeneous systems. *Applied Soft Computing Journal*, *24*, 500-510.
2. Alexey Lastovetsky, Ravi Reddy," Distributed Data Partitioning for Heterogeneous Processors Based on Partial Estimation of Their Functional Performance Models",European Conference on Parallel Processing, Netherland, August 25-28, 2009,pp.91-101
3. HamidrezaKhaleghzadeh, Ravi Reddy Manumachu, Alexey Lastovetsky," A Novel Data-Partitioning Algorithm for Performance Optimization of Data-Parallel Applications on Heterogeneous HPC Platforms",IEEE Transactions on *Parallel and Distributed Systems,Vol.29,Issue 10,2018 pp.2176-2190*
4. O.Beaumount,L.Carter,J.Ferrante,"Bandwidth-centric allocation of independent tasks on heterogeneous platforms" IEEE Proceedings 16th International Parallel and Distributed Processing Symposium**,** 15-19 April 2001.
5. Salah-Salim Boutammine, Daniel Millot, Christian Parrot," Dynamically Scheduling Divisible Load for Grid Computing, International Conference on High Performance Computing and Communications,26-28 September 2007, pp.263-272
6. ZimingZhong, Vladimir Rychkov, Alexey L. Lastovetsky, "Data Partitioning in MultiCore and Multi GPU Platforms using Functional Performance models", IEEE Transactions on Computers, Vol.64,Issue 9, 2015, pp.2506-2518.
7. Nagendra Singh and Yogendra Kumar," Multi-objective Economic Load Dispatch Problem Solved by New PSO", Advances in Electrical Engineering Volume 2015,pp.1-6
8. Leandro dos SantosCoelho, VivianaCoccoMariani, "Improved differential evolution algorithms for handling economic dispatch optimization with generator constraints", Energy Conservation and Management,Vol48, Issue 5, 2007,pp.1631-1639

Retrieval Number: B11360882S819/2019©BEIESP
DOI:10.35940/ijrte.B1136.0882S819

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

9. AbolfazlZaraki, MohdFauzi Othman,” Implementing Particle Swarm Optimization to Solve Economic Load Dispatch Problem”, IEEE International Conference of Soft Computing and Pattern Recognition, December 4-7,2009.
10. Y.A. Adekunle , Z.O. Ogunwobi , A. Sarumi Jerry , B.T. Efuwape , SeunEbiesuwa , and Jean-Paul Ainam,” A Comparative Study of Scheduling Algorithms for Multiprogramming in Real-Time Systems”, International Journal of Innovation and Scientific Research ISSN 2351-8014 Vol. 12 No. 1 Nov. 2014, pp. 180-185
11. Dr. R. Rama Kishore, ArpnaSaxena,” An efficient Multi parametric CPU scheduling algorithm for single processor systems”, International Journal of Computer Science & Engineering Technology, Vol. 5 No. 05 May 2014, pp.532-540
12. NeetuGoe,”A Comparative study of CPU scheduling algorithms”, nternational Journal of Graphics & Image Processing ,Vol 2,issue 4,2012, pp.245-251
13. Qasim Mohammed Hussein, ArmaneesaNaamanHasoon,” Dynamic process scheduling using genetic algorithm, Annual Conference on New Trends in Information & Communications Technology Applications, 7 - 9 March 2017, pp.111-115
14. Andrew J Page, Thomas M Keane, Thomas J Naughton,” Multi-heuristic dynamic task allocation using genetic algorithms in a heterogeneous distributed system”, Journal of Parallel and Distributed Computing, Vol.70, Issue 7,2010, pp.758-766.
15. AnuTaneja, Amit Kumar,” Efficient CPU Scheduling using Genetic Algorithm Approach”, International Journal of Computer Science and Technology, Vol. 3, Issue 3, 2012, pp.787-789.