IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-5, Issue-10 E-ISSN: 2347-2693

Parallel Job Scheduling Using Grey Wolf Optimization Algorithm for Heterogeneous Multi-Cluster Environment

Sapinderjit Kaur[1*], Kirandeep. Kaur[2], Amit.Chhabra[3]

[1*]Dept. of Computer Engineering and Technology, Guru Nanak Dev University, Amritsar, India
[2]Dept. of Computer Engineering and Technology, Guru Nanak Dev University, Amritsar, India
[3]Dept. of Computer Engineering and Technology, Guru Nanak Dev University, Amritsar, India

*Corresponding Author: er.sapinder.bal93@gmail.com, Tel.: +91-8284814571

Available online at: www.ijcseonline.org

Received: 22/Aug/2017, Revised: 08/Sep/2017, Accepted: 25/Sep/2017, Published: 30/Oct/2017

***Abstract*—:** Multi-cluster environment consists of computational nodes that allow computational problems with resource requirement more than those available resources in a cluster to be treated. Scheduling jobs in heterogeneous multi-cluster environments where each cluster has varied number of processors and each computational node has a varying speed is NP hard. Thus, we always search for sub-optimal solution for scheduling jobs. Various meta-heuristics have been proposed for scheduling jobs. The literature shows that the Genetic algorithm has been employed for parallel jobs scheduling in heterogeneous multi cluster environment. But it suffers from certain limitations like slow convergence speed, local optima problem. In this research work, a Grey Wolf Optimization algorithm (GWO) has been introduced in order to minimize makespan, flowtime and mean waiting time. The proposed methodology has shown quite significant improvement over available ones.

***Keywords*—** *Heterogeneous multi-cluster environment, Scheduling, Co-allocation, Grey wolf Optimization Algorithm(GWOA))*

I. INTRODUCTION

1.1 Heterogeneous multi-cluster environment:

Multi-cluster environments consist of multiple clusters of computers which act collaboratively and thus allowing computational problems with resource requirement more than those available resources in a cluster to be treated. In heterogeneous multi cluster environment, each cluster has a varied number of computational nodes and computational nodes have a varying speed. However, the degree of complexity of the scheduling process is greatly increased by the heterogeneity of computational nodes as well as co-allocation process, which distributes the tasks of parallel jobs across multiple clusters.

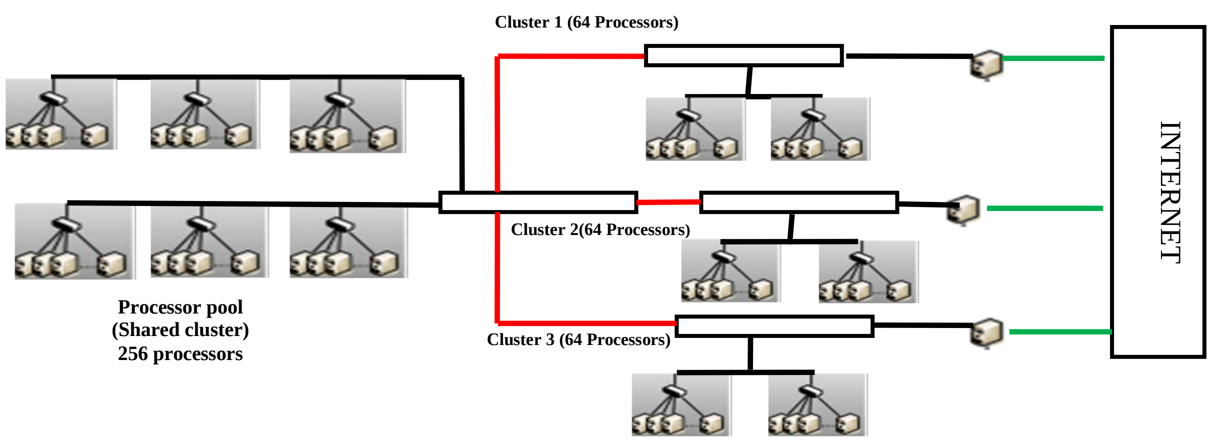

Fig 1: Heterogeneous Multi-cluster environment

1.2 Scheduling

A major issue while allocating jobs in multi cluster environment is how to effectively distribute the jobs among resources in order to improve performance so that some tasks do not suffer unbounded delays. This problem is called job scheduling. The complexity of a general scheduling problem is NP-Hard. The job scheduling problem in heterogeneous environment becomes more challenging task as it is important to achieve not only the promising potentials of tremendous distributed resources, but also effective and efficient scheduling techniques. Therefore, a lot of algorithms have been introduced for scheduling jobs in a computational Grid. All of them aim to minimize the job completion time (makespan). Different Grid scheduling approaches have been investigated and applied to different Grid scenarios. However, job scheduling in a computational Grid is multi-criteria in nature. Thus, it is important to develop algorithms with an enhanced scheduling technique, including additional criteria, along with minimum completion time of jobs.

- Job Selection
- Processor allocation

1.3 Co-allocation

While allocating jobs in heterogeneous multi-cluster environment, the scheduler checks the processor requirement of the selected job. If the processor requirement of the selected job is more than the currently available processors, then the parallel job is decomposed and allocated to multiple clusters. This type of allocation where single task is distributed across cluster boundaries is referred to as co-allocation.

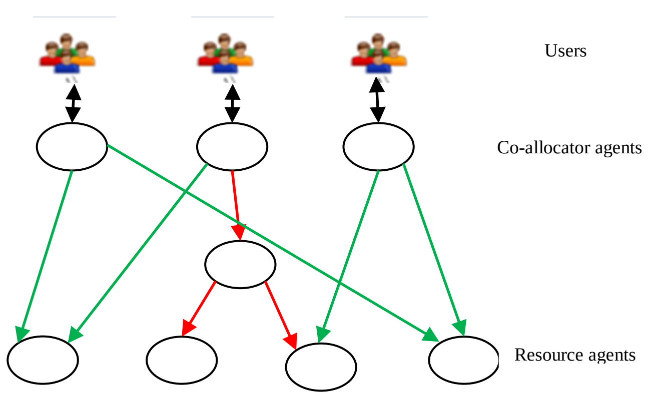

Fig 2: Co-allocation of jobs

Motivation

Different types of works done in the field of scheduling using metaheuristic algorithms have been studied in this work. Various researchers have proposed some very interesting ideas regarding this topic. Grey Wolf Optimization Algorithm is a relatively new technique in which much work hasn't been done. The other metaheuristic techniques have various limitations like slow convergence, not having global maxima, getting trapped in local minima etc. But GWOA on the other hand has proven to possess a rather fast convergence speed. Also it is not largely affected by the size of the problem and can handle more number of objective functions. Also most of the research has been usually done with sequential jobs. Moreover, scheduling of jobs is an NP hard problem. So, there always exists a scope of finding optimal solutions for scheduling jobs. So in this work we have been inspired to work with GWOA using real parallel workloads in a multi-cluster environment.

Rest of the paper is organized as follows, Section I contains the introduction, Section II contain the related work of job scheduling algorithms for heterogeneous multi-cluster environment., Section III addresses gaps in existing literature and proposed methodology. Section IV contains results and discussion for existing and proposed methodology, and Section V concludes research work with future directions.

II. Related Work

Eloi Gabaldon et al. (2016) proposed that reducing energy consumption in large-scale computing facilities has become a major concern in recent years. Most of the techniques have focused on determining the computing requirements based on load predictions and thus turning unnecessary nodes on and off. Nevertheless, once the available resources have been configured, new opportunities arise for reducing energy consumption by providing optimal matching of parallel applications to the available computing nodes. The authors implemented a multi-objective genetic algorithm which is based on a weighted blacklist able to generate scheduling decisions which optimizes the energy consumption and the makespan globally.

Chao-Tung Yang et al. (2010) developed a multi cluster system which worked with and without storage devices, and the system performance was evaluated. They dispatched jobs with the strategy to make the best use of system resources. The authors introduced new scheduling system based on multi clusters in diskless environments. Well-Balanced Allocation Strategy (WBAS) in which the scheduler dispatches jobs to appropriate resources across multi-clusters. The strategy focused on dispatching jobs to computational nodes with similar performance capacities, thus equalizing execution times among all the nodes the jobs require.

Sid Ahmed MAKHLOUF (2011) proposed that computational grids have the potential for solving large-scale problems using heterogeneous and geographically distributed resources. However, a number of the major technical hurdles must be overcome before this potential can be realized. The authors found that One problem that effective co-allocation of job is critical to effective utilization of computational grids and gives a certain Quality of Service for grid users is the s. Due to the lack of centralized control and the dynamic nature of resource availability, any successful co-allocation mechanism should be highly distributed and robust to the changes in the Grid environment.

T. Vigneswari et al. (2014) proposed job scheduling plays an important role for the efficient utilization of grid resources available across different domains and geographical zones. Scheduling of jobs is challenging and NP complete. Evolutionary Swarm Intelligence algorithms have been extensively used to address the NP problem in grid scheduling. Artificial Bee Colony (ABC) was proposed for optimization problems based on foraging behavior of bees. The proposed model utilizes a novel Heterogeneous Earliest Finish Time (HEFT) Heuristic Algorithm along with Min-Min algorithm to identify the initial food source. Simulation results show the performance improvement of the proposed algorithm over other swarm intelligence techniques.

Héctor Blanco et al. **(2011)** presented a new scheduling strategy that allocates multiple jobs from the system queue simultaneously on a heterogeneous multi cluster, by applying co-allocation when it is necessary. This strategy was composed by a job selection function and a linear programming model to find out the best allocation for multiple jobs. The proposed scheduling technique was shown to reduce the execution times of the parallel jobs and the overall response times

Joanna Kołodziej et al. (2010) have tackled the particular independent batch scheduling within the computational grid like a bi-objective global reduction issue with makespan and power usage as primary requirements and also applied Dynamic voltage Scaling Frequency technique towards the administration of accumulative energy used by the particular grid resources as well as create three genetic algorithm just as power conscious grid schedulers that have been empirically examined within three grid capacity circumstances within fixed and variable modes. The experiment results proved the performance of proposed method within the minimization of power usage by entire system as well as in variable load balancing belonging to the resources within grid clusters that are enough to manage required Quality grade.

Mohammad Shojafar et al. (2013) have introduced hybrid technique known as FUGE which is dependent on Fuzzy principle and genetic algorithm which is designed to execute optimum balancing of loads taking in to account execution time as well as cost. It alter the conventional genetic algorithm and also fuzzy technique in order to create new fuzzy based GA to be able to enhance the efficiency such as makespan. this algorithm allocates tasks to sources through taking in to account Virtual machine Computing rate, storage, bandwidth of VM and job size.

Javid Taheri et al. (2014) works on matchmaking scheduling phase and provide two algorithms to reduce the make-span for executing all jobs and transfer time for all data-fields. It use two collaborating algorithm for schedule job & replicate data-fields to connected nodes and storage nodes respectively. This proposed work can be easily extended to both data and job oriented system and for large system, it can address the bulk scheduling mode.

Hedieh Sajedi et al. (2014) introduced a new algorithm is established for scheduling in grid computing. This new algorithm provides a genetic algorithm, which uses the DLBS optimization algorithm (COA). This method reduces the completion time of machines and avoids trapping in a local minimum effectively. It has the capability of globally optimization. Two parameters execution time and average running time are used to analysis the proposed algorithm and compared with various existing algorithms.

Frédéric Pinel et al. (2010) presented the sensitivity analysis of a Cellular Genetic Algorithm (CGA) with local search is used to design a new and faster heuristic for the problem of mapping independent tasks to a distributed system. This approach improves the Min-Min heuristic search. It mainly minimizes the makespan time and reduces running time.

Kenli Li et al. (2014) introduced heuristic energy aware stochastic tasks scheduling algorithm to fix the issue regarding scheduling of independent tasks with standard distribution, timeline as well as energy usage budget restrictions. It formulates first power conscious scheduling as a linear programming that increases the certain assurance probabilities using timeline and power usage constraints.

Abawajy J.H. et al. (2008) proposed adaptive scheduling approach (AS) for HPC with much significance on space-sharing scheduling strategy. The requirement for tending to this issue emerges because of the deficiency of processing power of an individual framework for some global scale issues. Another two-level adaptive space-sharing scheduling strategy (ASSS) was proposed for non-devoted heterogeneous HPC and its performance has been examined to compare it with existing strategies (AAP and MAP Policy). Generalization was made that proposed approach outperforms as compared to an existing approaches because it considers heterogeneity in addition to node variability while scheduling.

Mateusz Guzek et al. (2010) presented energy conscious static algorithm i.e genetic cellular algorithm depending on task clustering methods which minimizes power consumption as a result of exchange of data within distributed system. In this, genetic operators operate on multiple tasks rather than implementing these on particular task. This is accomplished through assigning and organizing the jobs that comprises programs within the running issues, and reducing inter-processor communications. Experiment results showed that this algorithm is very persuasive with regard to application execution time, inter processor conversation and power exchanges dissipation.

Chunling Cheng et al.(2015) introduced power conscious task scheduling algorithm depending on vacation queuing model to minimize power consumption while maintaining the desired performance in parallel computing system. next depending on the busy time as well as busy cycle within the constant condition and also analyze requirements involved with tasks visit time and power usage of compute nodes within heterogeneous computing systems.

J. Barbosa and A.P. Monteiro et al. (2010) proposed the scheduling algorithm for homogeneous clusters to develop static schedules. The proposed work manages the scheduling of jobs, where jobs are represented by a directed acyclic graph (DAG) on clusters and compared to HEFT. Proposed work minimizes the waiting time along with the maximization of the cluster utilization.

Seyedali Mirjalili et al. (2014) proposed a new meta-heuristic called Grey Wolf Optimizer (GWO) which is evolved from the hunting behaviour of grey wolves. It involves four types of grey wolves referred to as alpha, beta, delta, and omega. Alpha wolves are at the top of hierarchy. The three main steps of hunting, searching for prey, encircling prey, and attacking prey, were implemented by the authors. The algorithm was then benchmarked on 29 well-known test functions, and the results were verified by a comparative study with Particle Swarm Optimization (PSO), Gravitational Search Algorithm (GSA), Differential Evolution (DE), Evolutionary Programming (EP), and Evolution Strategy (ES). The results showed that the GWO algorithm is able to provide very optimal results in comparison to these well known meta-heuristics.

III. METHODOLOGY

Following are the various gaps in earlier work on parallel job scheduling for heterogeneous multi-cluster environment.

1. The meta-heuristic techniques suffer from pre-mature convergence issues while evaluating optimistic values.
2. Many meta-heuristic techniques sometime stuck in local optima only.
3. The use of Grey Wolf algorithm for parallel scheduling is ignored by most of the existing researchers.

A. Grey wolf Optimization

Grey Wolf Optimization algorithm (GWOA) is basically a swarm-intelligence based method that mimics the leadership hierarchy and hunting behavior of grey wolves in nature. Grey wolves are considered to be apex predators; it means that in the food chain, they are at the top. Grey wolves usually prefer living in a pack. On an average group size is usually 5–12. In the GWO hierarchy, alpha wolves (α) are considered to be the most dominating member of the group (best candidate solution) and they are the decision makers. Other subordinates to α are beta (β) and delta (δ) wolves and they help in controlling the majority of wolves in the hierarchy that are referred to as omega wolves (ω).The ω wolves have lowest rank in the social hierarchy of grey wolves. The mathematical model for hunting behaviour of grey wolves comprises of the following:

(i) Tracking, chasing, and approaching the prey.
(ii) Pursuing, encircling, and harassing the prey until it stops moving.
(iii) Attacking the prey (Exploitation).

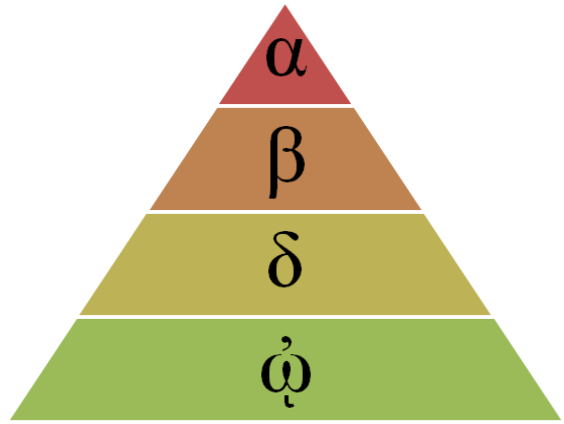

Fig 3: Grey wolf social hierarchy

Encircling Prey: Grey wolves encircle the prey while hunting

Hunting: Hunting of the prey is generally guided by α and β, and δ participate occasionally. The best candidate schedules, that is, α, β, andδ, have better understanding about the potential location of prey. The other search schedules, that is, (ω) update their positions on the basis of the position of three best search schedules.

Attacking the Prey: The grey wolves finish the hunt by attacking the prey when it stops moving While approaching the prey, we also keep on decreasing the fluctuation range . 'A' is a random value in the interval $[-a,a]$ where a is linearly decreased from 2 to 0 over the entire course of iterations. When random values of 'A' are in the range $[-1, 1]$, the next position of the search schedule can be in any position between its current position and the position of the prey (clusters).The value $|A| < 1$ forces the wolves to attack the prey. After the attack, they try to search for the prey in the next iteration again, wherein they again find the next best scheduleα among all wolves. This process keeps on repeating till the termination condition is met.

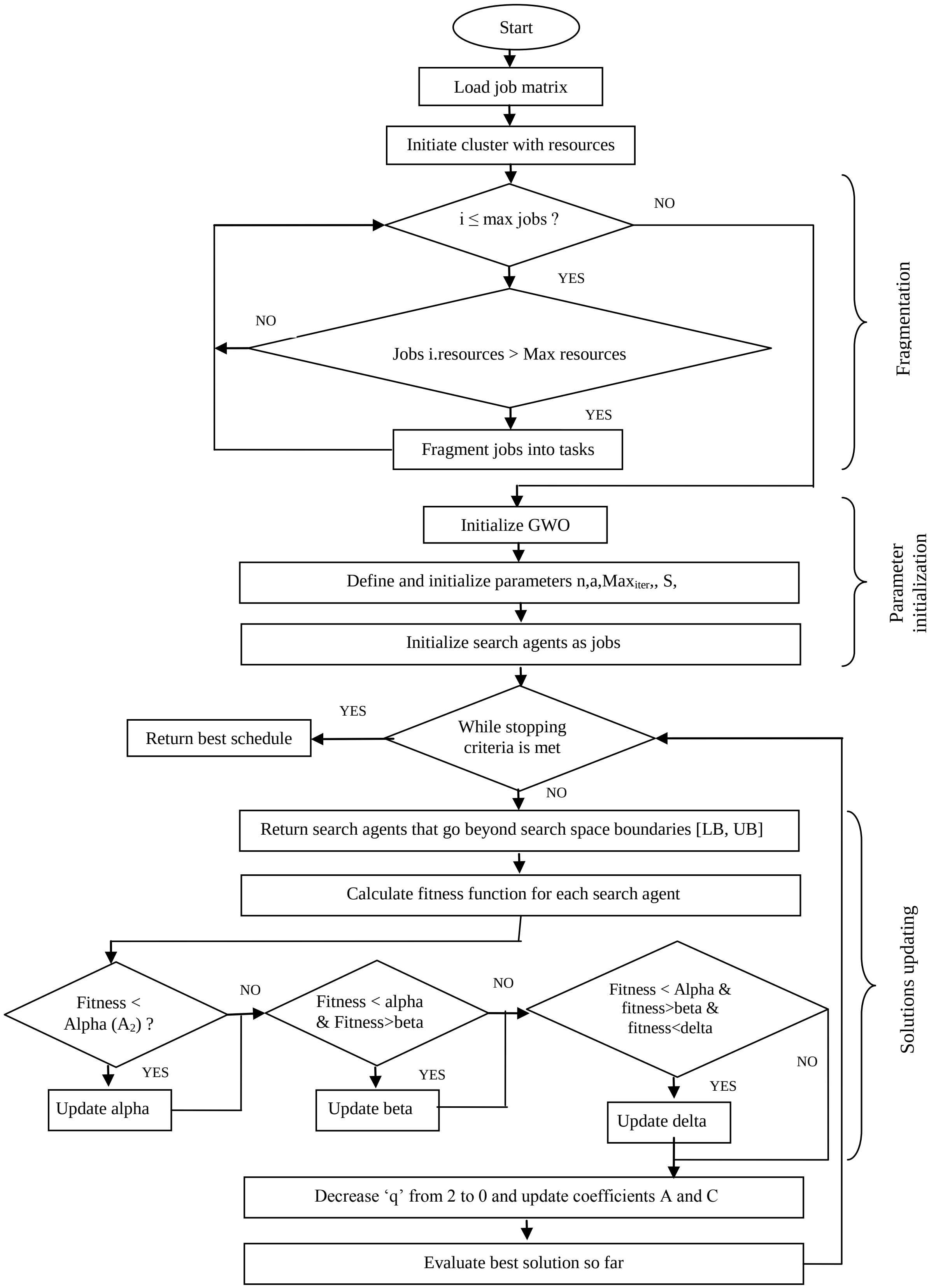

Fig 4: Flowchart for Proposed Technique

IV. RESULTS AND DISCUSSION

A heterogeneous multi-cluster environment has been created. 5 clusters have been taken into consideration while performing experiments. Each cluster has a varying computational speed. Three configurations of resources (computational nodes) have been taken i.e. 96, 112 and 128. The evaluation is done using real workload traces (Parallel Workload Archives). 100, 300 and 500 are the sets of jobs being used. Each metric is taken as average of 10 values with the same input and same resource configuration so as to remove the randomness in the output metrics.

Table 1: Overview: heterogeneous multi-cluster environment

Cluster Number	Resources	Speed
1	32	1
2	32	2
3	32	3
4	16	4
5	16	5

Table 1 depicts the heterogeneous multi-cluster environment created in the simulator where each cluster has a varying speed as well as resources.

Makespan: Makespan refers to the total length of the schedule i.e. the finishing time of the last task. İt is the most popular optimization criterion and indicates the productivity of a system. Lesser the value of makespan, more efficient is the scheduler.

Flowtime: The sum of finalization times of all tasks is called flowtime.

Mean Flow Time: The mean flow time of a schedule gives a way of measuring of the average time period which a job require inside a computer system as well as the average number of incomplete jobs in the system. It can also be described as the sum of the completion times of all jobs within the system.

Mean waiting time: The amount of time jobs spent in the ready queue waiting their turn to acquire CPU.

Initially experiments were performed for Genetic Algorithm followed by Grey Wolf Algorithm. Further a comparison was performed of these two techniques in order to evaluate the parameters i.e, Makespan, Flowtime, Mean waiting time. Bi-objective fitness function has also been taken into consideration which is defined by α parameter, which determines how the makespan or flowtime affect the decision process. The value taken for α =0.6. If α=0, it indicates that the flowtime completely dominates the fitness function, while α=1 denotes that only the makespan is considered for evaluating the solution. Any intermediate values of α represents a combination of both these parameters.

4.2.1 GA Tuning

The parameters were tuned by executing Genetic algorithm for different values of mutation rate (MR) and cross-over rate (CR) and number of iterations.

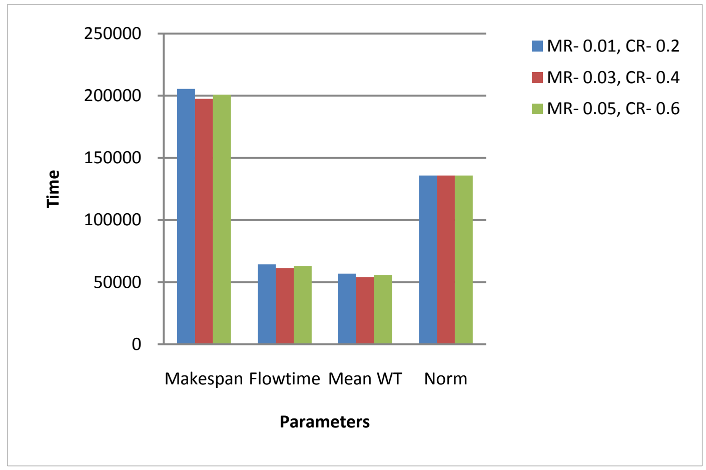

Figure 5: GA Tuning

4.2.4 Comparison GA and GWO

The parameter values of both the existing and proposed algorithm were compared with each other and the changes were noted. Initially, the case with 100 jobs was compared taking all 3 types of processors into consideration.

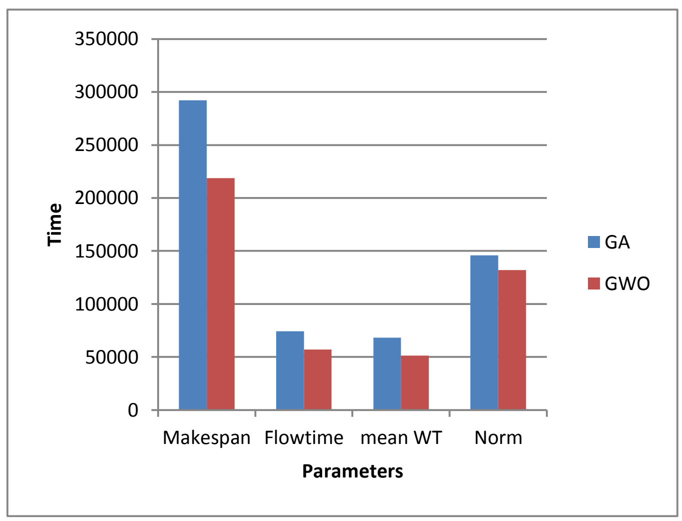

Figure 6: Comparison of GA and GWO for (100, 96) set

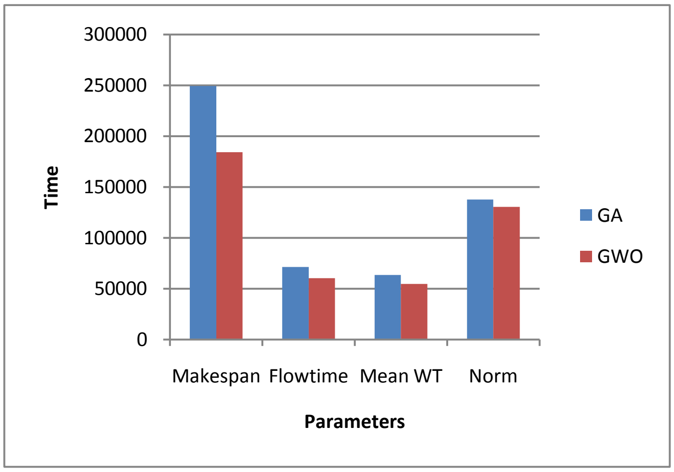

Figure 7: Comparison of GA and GWO for (100, 112) set

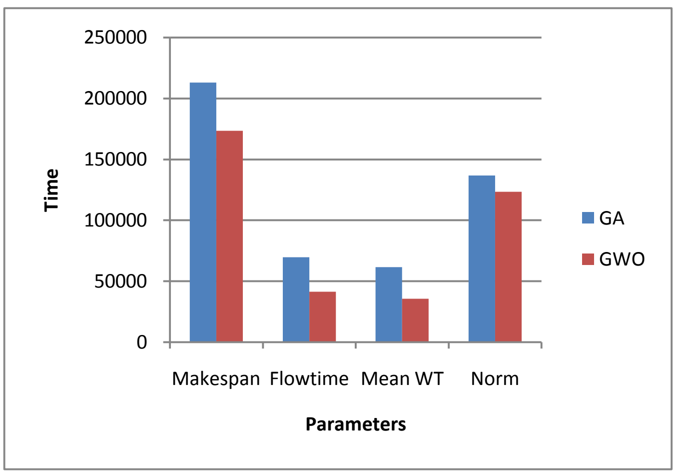

Figure 8: Comparison of GA and GWO for (100, 128) set

Fig 6 indicates the comparison of the results of the existing and proposed algorithm on 96 processors and 100 jobs. The results show that there is a very clear improvement in the values of our proposed algorithm. Fig 7 and Fig 8 follow the same trend the only difference being the number of processor.

The results for 300 jobs were evaluated for varied set of processors i.e. 96,112 and 128 processors which are shown below.

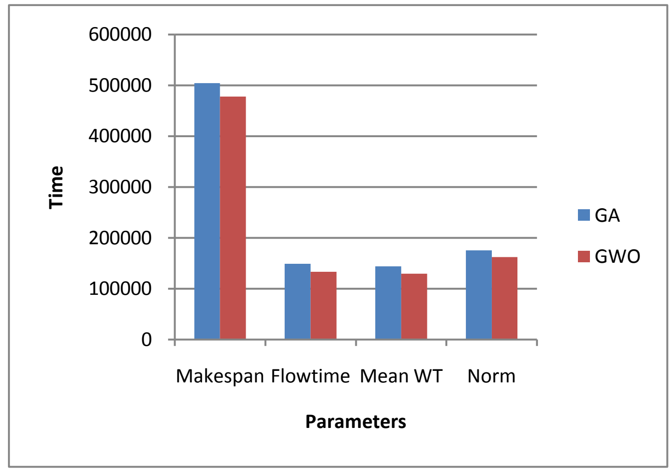

Figure 9: Comparison of GA and GWO for (300, 96) set

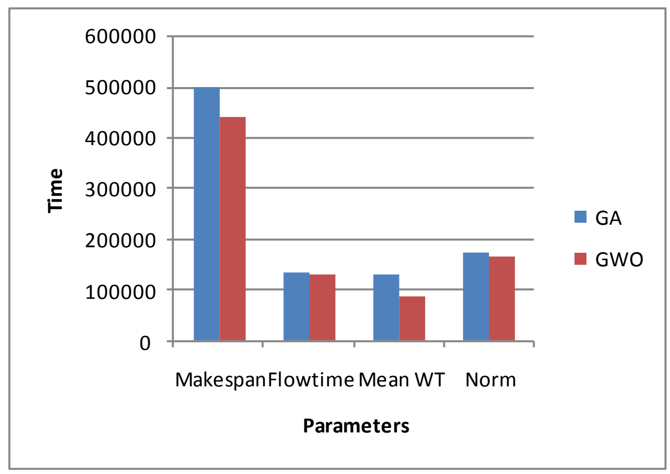

Figure 10: Comparison of GA and GWO for (300, 112) set

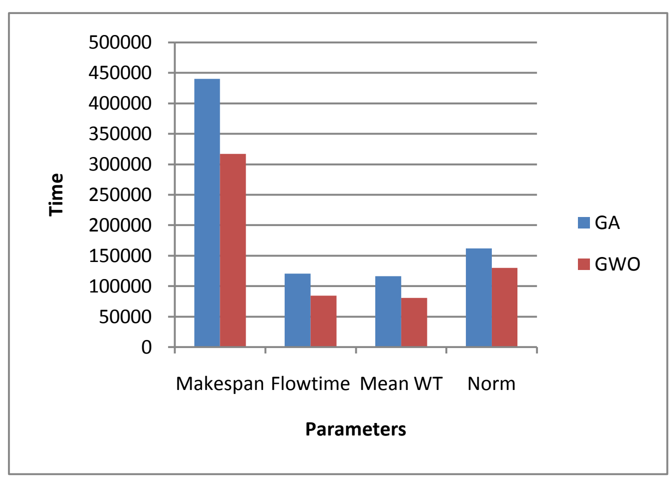

Figure 11: Comparison of GA and GWO for (300, 128) set

Fig 9 indicates the comparison of the results of the existing and proposed algorithm on 96 processors and 300 jobs. The results show that there is a very clear improvement in the values of our proposed algorithm. Fig 10 and Fig 11 follow the same trend the only difference being the number of processors. Now, the results for 500 jobs will be evaluated for varied set of processors i.e. 96, 112 and 128 processors.

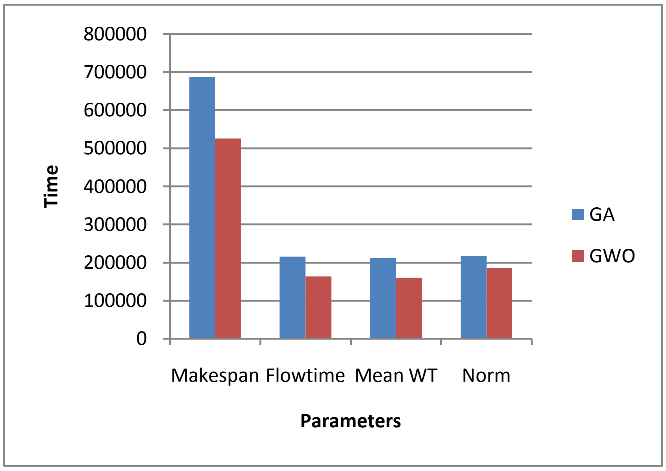

Figure 12: Comparison of GA and GWO for (500, 96) set

Fig 12 indicates the comparison of the results of the existing and proposed algorithm on 96 processors and 500 jobs. The results show that there is a very clear improvement in the values of our proposed algorithm.

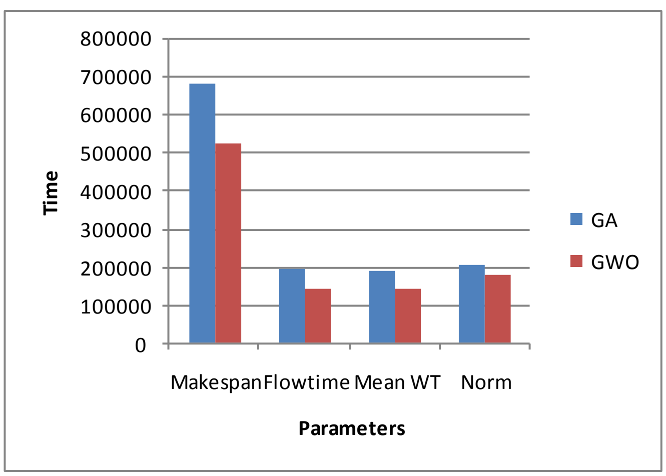

Figure 13: Comparison of GA and GWO for (500, 112) set

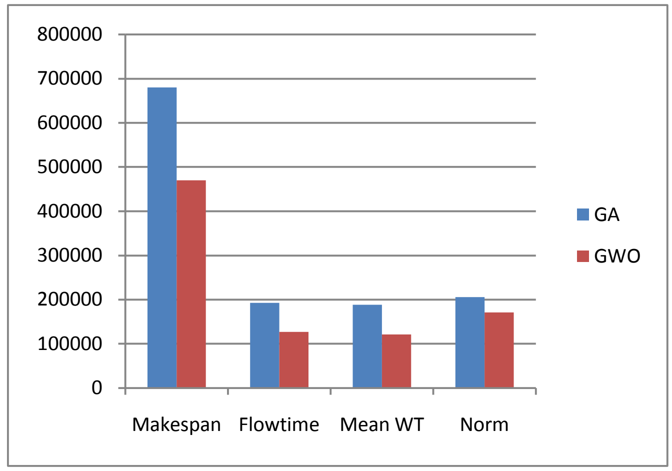

Figure 14: Comparison of GA and GWO for (500, 128) set

Fig 13 indicates the comparison of the results of the existing and proposed algorithm on 112 processors and 500 jobs. The results show that there is a very clear improvement in the values of our proposed algorithm. Fig 14 follows the same trend as Fig 13 with the only difference being number of processors.

It is clear from this graph set that as we keep on increasing the number of processors, the values for parameters shows an improvement.

In the end it is safe to say that the proposed algorithm has performed better than the existing technique in all the cases. We have successfully shown that the makespan, flowtime, mean waiting time and bi-objective normalization function are effectively optimized by GWO algorithm better than the Genetic Algorithm for every configuration of processors (96, 112, 128) and job set (100, 300, 500).whenever workload is increased, the values of the parameters increase but even then the values of GWO parameters are less in comparison to GA. When jobs are kept fixed and the number of processors is varied, even then GWO shows the optimized results in comparison to GA.

V. CONCLUSION and Future Scope

The simultaneous optimization of parameters is quiet difficult if there is a correlation between them. So, bi-objective optimization has been considered which is referred to as normalization function which works on the two parameters i.e. makespan and flow time and it is defined by α parameter, which determines how the makespan or flowtime affect the decision process. The value taken for α=0.6. If α=0, it indicates that the flowtime completely dominates the fitness function, while α=1 denotes that only the makespan is considered for evaluating the solution. Any intermediate values of α represents a combination of both these parameters.

In this proposed work scheduling GWO performs better than Genetic algorithm. The comparison between the proposed techniques with the existing technique using parameters such as: Flow Time, waiting time and makespan is shown in this work. This comparison has shown that the proposed work results are much better than the existing results. The results of existing technique and proposed technique have been compared taking two strategies. For the first time, jobs are kept constant and the number of processors is varied and then number of processors is kept constant and number of jobs is varied.

It can be observed that for 96 processors the improvement in makespan, flowtime, mean waiting time and bi-objective fitness function is: 18%, 26%, 10.5% and 5% respectively. So we conclude that if we keep on increasing the number of resources GWO shows more improvement in comparison to GA

The proposed technique has not taken inter-process communication into consideration. So, in near future this can be considered while scheduling of jobs in heterogeneous multi-cluster systems. Also, In near future hybridization of Gray wolf with other techniques will also be considered to further optimize the results.

REFERENCES

[1] Abawajy, Jemal H., and Sivarama P. Dandamudi. "*Parallel Job Scheduling on Multicluster Computing Systems." CLUSTER*. Vol. 3. 2003

[2] Abawajy, Jemal H. "*Fault-tolerant scheduling policy for grid computing systems.*" Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International. IEEE, 2004

[3] Alkhanak, Ehab Nabiel, Sai Peck Lee, and Saif Ur Rehman Khan. "*Cost-aware challenges for workflow scheduling approaches in cloud computing environments: Taxonomy and opportunities."* Future Generation Computer Systems 50 (2015): 3-21.

[4] Arsuaga-Ríos, María, and Miguel A. Vega-Rodríguez. "*Energy optimization for task scheduling in distributed systems by an Artificial Bee Colony approach.*" Nature and Biologically Inspired Computing (NaBIC), 2014 Sixth World Congress on. IEEE, 2014.

[5] .Barbosa, Jorge, and António P. Monteiro. "*A list scheduling algorithm for scheduling multi-user jobs on clusters.*" International Conference on High Performance Computing for Computational Science. Springer Berlin Heidelberg, 2008.

[6] Bilgaiyan, Saurabh, Santwana Sagnika, and Madhabananda Das. "*An analysis of task scheduling in cloud computing using evolutionary and swarm-based algorithms.*" International Journal of Computer Applications 89.2 (2014).

[7] Blanco, Héctor, et al. *"Multiple job co-allocation strategy for heterogeneous multi-cluster systems based on linear programming." The Journal of Supercomputing* 58.3 (2011): 394-402.

[8] Bokhari, M. U., Mahfooz Alam, and Faraz Hasan. "*Performance analysis of dynamic load balancing algorithm for multiprocessor interconnection network.*" Perspectives in Science 8 (2016): 564-566.

[9] Civicioglu, Pinar, and Erkan Besdok. "*A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms.*" Artificial intelligence review 39.4 (2013): 315-346.

[10] Dorronsoro, B., Nesmachnow, S., Taheri, J., Zomaya, A. Y., Talbi, E. G., & Bouvry, P. (2014). *A hierarchical approach for energy-efficient scheduling of large workloads in multicore distributed systems.* Sustainable Computing: Informatics and Systems, 4(4), 252-261.

[11] Farahnakian, F., Ashraf, A., Pahikkala, T., Liljeberg, P., Plosila, J., Porres, I., & Tenhunen, H. (2015). *Using ant colony system to consolidate vms for green cloud computing*. IEEE Transactions on Services Computing, 8(2), 187-198.

[12] Fang, Yiqiu, Fei Wang, and Junwei Ge. "*A task scheduling algorithm based on load balancing in cloud computing.*" International Conference on Web Information Systems and Mining. Springer Berlin Heidelberg, 2010.

[13] Feng, Yu Jianjun Sun Shudong Zheng. "*Job-shop scheduling study by dynamic evaluation based immune algorithm* [J]." Chinese Journal of Mechanical Engineering 3 (2005): 004.

[14] Ferrandi, F., Lanzi, P. L., Pilato, C., Sciuto, D., & Tumeo, A. (2010). *Ant colony heuristic for mapping and scheduling tasks and communications on heterogeneous embedded systems*. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 29(6), 911-924.

[15] Gabaldon, Eloi, et al. "*Blacklist muti-objective genetic algorithm for energy saving in heterogeneous environments*." *The Journal of Supercomputing* 73.1 (2017): 354-369.

[16] Guzek, M., Pecero, J. E., Dorronsoro, B., Bouvry, P., & Khan, S. U. (2010, June). *A cellular genetic algorithm for scheduling applications and energy-aware communication optimization.* In High Performance Computing and Simulation (HPCS), 2010 International Conference on (pp. 241-248). IEEE.

[17] HOU, You-hua, et al. "*Analysis on active power fluctuation characteristics of large-scale grid-connected wind farm and generation scheduling simulation under different capacity power injected from wind farms into power grid [J]."* Power System Technology 5 (2010): 013.

[18] .Javadi, Bahman, Mohammad K. Akbari, and Jemal H. Abawajy. "Performance analysis of heterogeneous multi-cluster systems." *Parallel Processing, 2005. ICPP 2005 Workshops. International Conference Workshops on*. IEEE, 2005.

[19] Jia, H. Z., Nee, A. Y., Fuh, J. Y., & Zhang, Y. F. (2003). *A modified genetic algorithm for distributed scheduling problems*. Journal of Intelligent Manufacturing, 14(3-4), 351-362.

[20] Javanmardi, Saeed, et al. "*Hybrid job scheduling algorithm for cloud computing environment."* Proceedings of the Fifth International Conference on Innovations in Bio-Inspired Computing and Applications IBICA 2014. Springer International Publishing, 2014.

[21] Javanmardi, S., Shojafar, M., Amendola, D., Cordeschi, N., Liu, H., & Abraham, A. (2014). *Hybrid job scheduling algorithm for cloud computing environment.* In Proceedings of the Fifth International Conference on Innovations in Bio-Inspired Computing and Applications IBICA 2014 (pp. 43-52). Springer International Publishing.

[22] Liu, J., Luo, X. G., Zhang, X. M., Zhang, F., & Li, B. N. (2013). *Job scheduling model for cloud computing based on multi-objective genetic algorithm.* IJCSI International Journal of Computer Science Issues, 10(1), 134-139.

[23] Liu, S. L., Liu, Y. X., Zhang, F., Tang, G. F., & Jing, N. (2007). *Dynamic web services selection algorithm with QoS global optimal in web services composition. Ruan Jian Xue Bao*(Journal of Software), 18(3), 646-656.

[24] Li, Jun-Qing, Quan-Ke Pan, and Kai-Zhou Gao. *"Pareto-based discrete artificial bee colony algorithm for multi-objective flexible job shop scheduling problems."* The International Journal of Advanced Manufacturing Technology 55.9-12 (2011): 1159-1169.

[25] Li, S., Li, G., Wang, X., & Liu, Q. (2005). *Minimizing makespan on a single batching machine with release times and non-identical job sizes.* Operations Research Letters, 33(2), 157-164.

[26] Lorpunmanee, S., Sap, M. N., Abdullah, A. H., & Chompoo-inwai, C. (2007). *An ant colony optimization for dynamic job scheduling in grid environment*. International Journal of Computer and Information Science and Engineering, 1(4), 207-214.

[27] Masdari, M., ValiKardan, S., Shahi, Z., & Azar, S. I. (2016). Towards workflow scheduling in cloud computing: *A comprehensive analysis. Journal of Network and Computer Applications*, 66, 64-82.

[28] Makhlouf, Sid Ahmed, and Belabbas Yagoubi. "*Resources Co-allocation Strategies in Grid Computing*." *CIIA*. 2011.

[29] Mirjalili, Seyedali, Seyed Mohammad Mirjalili, and Andrew Lewis. "Grey wolf optimizer." *Advances in Engineering Software* 69 (2014): 46-61.

[30] Moganarangan, N., Babukarthik, R. G., Bhuvaneswari, S., Basha, M. S., & Dhavachelvan, P. (2016). *A novel algorithm for reducing energy-consumption in cloud computing environment*: Web service computing approach. Journal of King Saud University-Computer and Information Sciences, 28(1), 55-67.

[31] Pinel, F., Dorronsoro, B., Pecero, J. E., Bouvry, P., & Khan, S. U. (2013). *A two-phase heuristic for the energy-efficient scheduling of independent tasks on computational grids.* Cluster Computing, 16(3), 421-433.

[32] Priya, S. Baghavathi, M. Prakash, and K. K. Dhawan. "*Fault tolerance-genetic algorithm for grid task scheduling using check point."* Grid and Cooperative Computing, 2007. GCC 2007. Sixth International Conference on. IEEE, 2007.

[33] Qu, H., Mashayekhi, O., Terei, D., & Levis, P. (2016). *Canary: A Scheduling Architecture for High Performance Cloud Computing*. arXiv preprint arXiv:1602.01412.

[34] Sajedi, Hedieh, and Maryam Rabiee. *"A metaheuristic algorithm for job scheduling in grid computing*." International Journal of Modern Education and Computer Science 6.5 (2014): 52.

[35] Sheikhalishahi, M., Ebrahimipour, V., Shiri, H., Zaman, H., & Jeihoonian, M. (2013). *A hybrid GA–PSO approach for reliability optimization in redundancy allocation problem.* The International Journal of Advanced Manufacturing Technology, 68(1-4), 317-338.

[36] Singh, S., & Chana, I. (2015). QRSF: *QoS-aware resource scheduling framework in cloud computing*. The Journal of Supercomputing, 71(1), 241-292.

[37] Singh, Sukhpal, and Inderveer Chana. "*Energy based efficient resource scheduling: a step towards green computing*." Int J Energy Inf Commun 5.2 (2014): 35-52.

[38] Tao, F., Zhang, L., Venkatesh, V.C., Luo, Y. and Cheng, Y., 2011. *Cloud manufacturing: a computing and service-oriented manufacturing model.* Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 225(10), pp.1969-1976.

[39] Vigneswari, T., and Ma Maluk Mohamed. "*Optimal GridScheduling Using Improved Artificial Bee Colony Algorithm*." *International Journal of Computer, Electrical, Automation, Control and Information Engineering* 8 (2014).

[40] Wang, Lizhe, et al. "*Energy-aware parallel task scheduling in a cluster."* Future Generation Computer Systems 29.7 (2013): 1661-1670.

[41] Xu, Yuming, Kenli Li, Jingtong Hu, and Keqin Li. "*A genetic algorithm for task scheduling on heterogeneous computing systems using multiple priority queues*." Information Sciences 270 (2014): 255-287

[42] Yang, Chao-Tung, et al. "*Well-balanced allocation strategy for multi-cluster computing environments." Future Trends of Distributed Computing Systems, 2008. FTDCS'08. 12th IEEE International Workshop on*. IEEE, 2008.

[43] YING, Chang-tian, Jiong YU, and Xing-yao YANG. "*Energy-aware task scheduling algorithms in cloud computing [J]*." Microelectronics & Computer 5 (2012): 044.

[44] Zhao, Jianfeng, and Hongze Qiu. "*Genetic algorithm and ant colony algorithm based Energy-Efficient Task Scheduling*." Information Science and Technology (ICIST), 2013 International Conference on. IEEE, 2013.