© Universiti Tun Hussein Onn Malaysia Publisher's Office

IJIE

Journal homepage: http://penerbit.uthm.edu.my/ojs/index.php/ijie
ISSN : 2229-838X e-ISSN : 2600-7916

The International Journal of Integrated Engineering

Enhanced Job Ranking Backfilling Based on Linear and Logarithmic Ranking Equations

Hazem Al-Najjar*, S S N Alhady, Junita Mohammad Saleh

School of Electrical & Electronic Engineering, Engineering Campus, Nibong Tebal, Penang
Universiti Sains Malaysia (USM), Penang, MALAYSIA

*Corresponding Author

DOI: https://doi.org/10.30880/ijie.2019.11.08.001
Received 6 March 2018; Accepted 29 August 2019; Available online 30 December 2019

Abstract: Grid system is used by many researchers' and scholars all over the world to solve the complicated and complex problems in different sciences. Job ranking backfilling is the most used model by many researchers in grid system to improve the performance of job scheduling algorithm. The model aims on serving the smallest job in the queue. As a second improvement of job backfilling, researchers proposed job ranking backfilling that serve job based on ranking equation. This paper proposes an enhance job ranking algorithm based on using linear and logarithmic ranking equations. Both proposed ranking equations used curve estimation model to predict on the variables' coefficients. By simulation and after different tests, the average results of job ranking backfilling with linear ranking equation outperform conventional job ranking backfilling with improvement equal 3.2% and 56.53% in total execution time and average waiting time, respectively. In addition, job ranking backfilling with logarithmic ranking equation shows average improvement equal 1.78% and 46.62% in total execution time and average waiting time, respectively. The results indicate that the proposed ranking equations would improve conventional job ranking backfilling in high and low demand grid system under different conditions.

Keywords: job scheduling, job ranking Backfilling, grid computing, logarithmic and linear Ranking equation

1. Introduction

Grid Job scheduling is defined as changing the flow of jobs to improve the performance metrics such as total execution time, average waiting time, bandwidth and so forth. Many job scheduling algorithms have been suggested such as First Come First Serve (FCFS), longest job first and shortest job first, Min-Min, Min-Max. Besides that, different researchers have suggested different algorithms using different techniques as follows. Rodero et. al. [1] proposed a Job Ranking Backfilling (JRB) to improve the performance of a grid system, by advancing jobs using a rank value of each job. Ahmed et al. [2] proposed dynamic priority job scheduling algorithm based on multiple criteria such as CPU resources requirement, IO resource requirement and job criticality. The new proposed model aims to optimize the job starvation problem. The results indicate that the starvation problem is solved and the performance is enhanced. Moreover, to create a fairness between jobs, da Silva [3] proposed a new backfilling model to reduce fragmentat ion and to obtain better response times. At the same time, fairness between jobs is kept low and predictability remains high. The results show that the performance of the system is improved compared with previous model. Fanfakhri [4] proposed new multi-objective optimization algorithm to optimize the execution time, the energy consumption and the cost of booked resources in the grid system. The proposed algorithm used new optimization function to select the best frequencies that can improve the performance of the optimized metrics. Moreover, the proposed algorithm gives the minimum energy consumption and minimum cost with maximum performance.

Corresponding author: Hazem_najjar@yahoo.com

Moreover to find optimal sequences of jobs in grid system many solutions and techniques have been suggested. Zhang [5] proposed heuristic approach based on particle swarm optimization to solve job scheduling in grid system. The proposed model aims to generate an optimal schedule to get the minimum completion time. By simulation, the results show that the particle swarm optimization algorithm is able to get the better schedule results than previous models. Furthermore, to improve serving the performance in case of resources failure, Idris [6] proposed Job Scheduling with Fault Tolerance using Ant Colony Optimization. The proposed model is based on resource failure rate and checkpoint roll back recovery strategy which aims of reducing the amount of work that is lost upon failure of the system by immediately saving the state of the system. The experimental results show that there is an improvement in the user's QoS requirement. In [7] researchers proposed dynamic scheduler that can find an optimal schedule to complete jobs within minimum makespan using an intelligent metaheuristics method based on firefly algorithm. The simulation results illustrated that the proposed model performed better than previous scheduling models.

On the other hand, in [8] researchers proposed new job scheduling model based on improving Cuckoo Search (ICS) optimization method to serve jobs so that makespan and completion time of job scheduling are optimized. The ICS results revealed that the proposed algorithm provides better results in comparison with Cuckoo Search (CS) and could be used to improve the performance of a grid system. In addition, different hybrid techniques were used by many researchers to enhance the performance of job scheduling models. Improving the backfilling concept is used by many researchers to improve and enhance grid performance as proposed in [9-13]. Other researchers tried to target the run time prediction model as proposed in [14]. Moreover, backfilling techniques have been used by many scholars to find a new way to solve the problem of job scheduling as shown in [15-19].

This paper is focused on improving job ranking backfilling algorithm by changing the ranking equation of job serving part. The new enhancement will use a new experimental ranking equations based on optimal curve estimation model. Finally, this paper is divided into the following sections. In Section 2, job ranking backfilling model is explained, where enhanced job ranking backfilling model is discussed in Section 3. In Section 4, experiments, results and discussion are explained. The conclusions and future trends are drawn in Section 6.

2. Job Ranking Backfilling Model

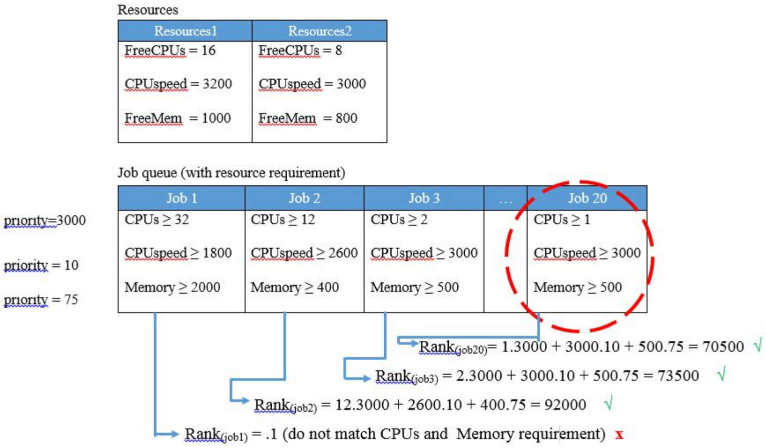

Fig. 1- Job ranking backfilling model [1]

The Job Ranking Backfilling (JRB) was proposed by Rodero et. al. [1] to overcome job backfilling scheduling policy problem. In JRB, to advance a job from scheduler's queue two models were proposed, first model is depending on the manual method, which implies defining the load of the jobs manually. The second model is depending on automatically guessing the jobs load by creating a prediction model. In the JRB model, the following definition of the job load is used to advance jobs depending on resources requirements as follows:

- Check the job resources requirement along with the available resources.
- Compute a rank value for each job to find the smallest jobs using equation 1. If job's requirements are not matched, then it gets a rank value of -1.

$$Rank\ value\ = 3000 * No\_CPU + 75 * Memory\_size + \qquad 10 * CPU\_Speed \qquad (1).$$

- Select the job with the smallest positive rank value and assign it to start execution.

To understand the concept of the JRB algorithm the following example is given as explained in Fig. 1. In which, two resources contain 16 and 8 CPUs, receptively and 20 jobs in the scheduler's queue. Firstly, Equation 1 will be used to calculate the rank for each job and the jobs that have highest requirements compared with the resources in the environment will get -1, where the smallest rank job will be advanced to be executed.

3. Enhanced Job Ranking Backfilling Model

To improve the performance of job ranking backfilling, new ranking equations are developed. The developed equations are based on linear and logarithmic functions. The linear equation uses number of CPU and memory size with constant as shown in Equation 2, where the logarithmic equation uses the logarithmic value of number of CPU and memory size with constant as shown in Equation 3. To calculate linear and logarithmic functions, an optimal curve estimation technique is used. The output of the curve estimation is a run time of a job. The run time is used to find the rank equation that can define the smallest and largest jobs. In which, the smallest rank job value is considered as a job that needs less resources and short time to execute, where the large rank job value is considered as a job that needs large resources and long time to execute a job. The proposed ranking equations are as follows:

$$\text{Linear } Rank\ value\ = F1 * No\_CPU + F2 * Memory\_size\ + constant \qquad (2)$$

$$\text{logarithmic } Rank\ value\ = F3 * Log_{10}(No\_CPU) + F4 * Log_{10}(Memory\_size\ ) + constant \qquad (3)$$

where F1 and F2 are the factors multiply with number of CPU and memory size using the linear function. F2 and F3 are the factors multiply with number of CPU and memory size using the logarithmic function. The proposed equations omitted the behavior of CPU speed to minimize the number of variables that needed in the ranking equations. To run the curve estimation model, the predefined parameters should be tuned to find the best factors that can describe the linear and logarithmic functions. The parameters are chosen based on the previous works that considered those values as optimal values to be used as shown in Table 1.

Table 1 - Curve estimation parameters for linear and logarithmic functions

Parameter	Value
Estimation Method	Levenberg-Marquardt
Maximum iterations	100
Sum of squares convergence	$1x10^{-8}$
Parameter convergence	$1x10^{-8}$
Dependent Variable	Job run time
Linear model expression	Equation 2
logarithmic model express	Equation 3

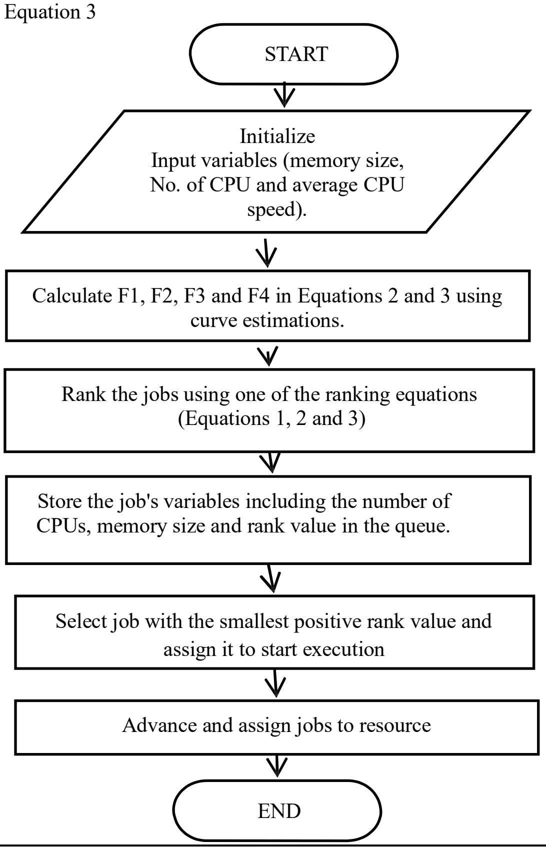
START
Initialize
Input variables (memory size, No. of CPU and average CPU speed).
Calculate F1, F2, F3 and F4 in Equations 2 and 3 using curve estimations.
Rank the jobs using one of the ranking equations (Equations 1, 2 and 3)
Store the job's variables including the number of CPUs, memory size and rank value in the queue.
Select job with the smallest positive rank value and assign it to start execution
Advance and assign jobs to resource
END

Fig. 2 - Enhanced job ranking backfilling model.

Moreover, to understand the overall proposed model, the flowchart in Fig. 2 is used. Each job submitted to the scheduler contains three variables including number of CPUs, memory size and average CPU speed. The first step of building the enhanced job ranking backfilling model is to propose two ranking equations to rank the jobs based on linear function and logarithmic function. The ranking equations, after that, is used to rank the incoming jobs and store the rank value with the job's variables in the queue. The scheduler chooses the smallest ranked job in the queue to be served by resource.

4. Experiments, Results and Discussion

To verify the effectiveness of the proposed model, this section presents two main stages including finding ranking equation test and evaluating the performance of the proposed job scheduling model. The first stage uses curve estimation model to find the factors including F1, F2, F3 and F4. After that the enhanced job ranking backfilling will use the proposed equation to test the performance of the job scheduling system. To evaluate the enhance job ranking backfilling model three different tests are adopted including creation time test, number of CPUs test and number of jobs test. Moreover, the test adopts total execution time and average waiting time as performance metrics to test the scheduling system. Finally, the default experimental setup for the tests are explained as follows:

- Jobs use a poisson distribution as a job creation time.
- 200 CPUs in the default number of processors in experimental environment.
- The number of jobs is equal 10000.

Job ranking backfilling scheduling model, enhanced ranking backfilling scheduling model based on linear equation and enhanced ranking backfilling scheduling model based on logarithmic are denoted as JRB, JRB_Linear and JRB_Log10 respectively.

4.1 The Proposed Ranking Equations

To improve the job ranking backfilling two ranking equations are proposed using curve estimation model. The curve estimation uses Equations 1 and 2 with the run time as output. The results of the linear and logarithmic ranking equations are shown in Table 2 to 5 and from Table 6 to 9, respectively. For Linear rank equation, Table 2 shows that optimal values for F1, F2 and constant achieved after five iterations and the optimal values are 6.565,-0.011 and 257.099, respectively. To check the significant of the factors, Table 3 shows the significant of each factor and the standard error. The results show that all the factors are significant and the standard error for F1, F2 and constant is 2.474, 0.456, and 48.633, respectively. The correlation test shows that there are a weak correlations between factors, which indicate that no relationship between factors can be found in a linear function as shown in Table 4. Besides that, Table 5 shows an analysis of variance for the regression and residual models. The results show that R-square is equal .001 which calculated using Equation 4. The overall curve estimation for linear equation shows that the chosen factors can predict on 0.01% of the job's run time. This indicate that the optimal factors for linear equation cannot predict on the run time efficiently.

$$R\ squared = 1 - \frac{Residual\ Sum\ of\ Squares}{Corrected\ Sum\ of\ Squares} \quad (4)$$

Table 2 - iteration history for linear rank equation.

Iteration	Residual Sum	F	F	consta
0	$1.98x10^{11}$	1	1	1
1	$1.98x10^{11}$	1.589	0.1518	1.0420
2	$1.98x10^{11}$	7.384	0.2939	1.4804
3	$1.98x10^{11}$	5.400	0.1308	103.90
4	$1.97x10^{11}$	6.564	-	257.09
5	$1.97x10^{11}$	6.56481	-0.01142	257.098

Table 3 - Parameter estimates for linear rank equation.

Facto	Estima	Standard.	Lower Bound	Upper Bound
F1	6.5648	2.4740	1.715	11.414
F2	-	0.4565	-	0.8834
constant	257.0989	48.63309	161.7682	352.429

Table 4 - Factors correlation for linear rank equation.

	F	F	consta
F1	1		-
F2	0.053458	1	-0.16964
constant	-0.37746	-0.16964	1

Table 5 - Analysis of variance for linear rank equation.

Source	Sum of Square	Degree of	Mean Squares
Regression	1.07x1	3	3.56x1
Residual	1.97x10	99	197520
Uncorrected	1.99x10	100	
Corrected Total	$1.98\text{x}10^{11}$	9999	

Moreover, for logarithmic ranking equation, Table 6 shows that optimal values for F3, F4 and constant achieved after six iterations and the optimal values are 568.757, 1.513 and 10.130, respectively. To check the significant of the factors, Table 7 shows the significant of each factor and the standard error. The results show that only F3 is significant and the standard error for F3, F4 and constant is 160.308, 99..117, and 152.679, respectively. The correlation test shows that there are a strong correlations (the absolute value is more than 0.7) between factors, which indicate that a relationship between factors can be found in a logarithmic function as shown in Table 8. Besides that, Table 9 shows an analysis of variance for the regression and residual models. The results show that R-square is equal .003 which calculated using Equation 4. The overall curve estimation for logarithmic equation shows that the chosen factors can predict on 0.03% of the job's run time. This indicate that the optimal factors for logarithmic equation cannot predict on the run time efficiently. In contract, the results show that logarithmic have more ability to predict on job's run time compared with linear model.

Table 6 - Iteration history for logarithmic rank equation.

Iteration	Residual Sum	F	F	consta
0	1.99	1	1	1
1	1.98	21.153	8.3666	25.787
2	1.98	72.876	-	31.118
3	1.98	185.36	46.603	-
4	1.97	577.97	89.688	17.700
5	1.97	568.75	91.514	10.12
6	$1.97\text{ x}10^{11}$	568.756	91.51291	10.1301

Table 7 - Parameter estimates for logarithmic rank equation.

Facto	Estima	Standard.	Lower Bound	Upper Bound
F3	568.7	160.30	254.51	882.99
F4	91.512	123.99	-	334.57
constant	10.13012	152.6792	-289.152	309.41

Table 8 - Factors correlation for logarithmic rank equation.

	F3	F4	constant
F3	1	0.841905	-0.90576
F4	0.841905	1	-0.92894
constant	-0.90576	-0.92894	1

Table 9 - Analysis of variance for logarithmic rank equation.

Source	Sum of Square	Degree of	Mean Square
Regression	1.52x1	3	5.06
Residual	1.97	99	197069
Uncorrected	1.99	100	
Corrected Total	$1.98\text{ x}10^{11}$	9999	

Finally, Equations 2 and 3 can be rewritten as follow:

$$\text{Linear } Rank\ value = 6.565 * No_{CPU} - 0.011 * Memory_{size} + 57.099 \quad (5)$$

$$\text{logarithmic } Rank\ value = 568.756 * Log_{10}(No\_CPU) + 91.513 * Log_{10}(Memory\_size) + 10.130 \quad (6)$$

Equations 5 and 6 are used as a ranking equation instead of experimental ranking equation, to check the performance of the enhanced job ranking backfilling model in grid system. The behavior of the proposed ranking equations are proposed in the following sections.

4.2 Testing the Performance of Job Rank backfilling based on Ranking Equation

The experimental tests are divided into three main matrices including job creation time test, resources experimental test, Number of jobs test. The default parameters will be used in all tests and any changes will be mentioned in the test. Moreover, the Average Waiting Time (AWT) and Total Execution Time (TET) are used to evaluate the proposed job scheduling model. The unit for all performance metrics are in seconds.

(a) Jobs Creation Time Test

Two job distributions are used to test the performance of the proposed ranking equation, which are Poisson distribution and discrete uniform distribution as shown in Equations 7 to 8.

$$\text{Poisson} = f(k, \lambda) = \frac{\lambda^k e^{-\lambda}}{k!} \qquad (7)$$

$$\text{Discrete Uniform} = f(k; a, b) = \frac{\lfloor k \rfloor + a - 1}{b - a + 1}, where\ k \in [a, b] \qquad (8)$$

The JRB, JRB based on linear ranking equation and JRB based on logarithmic ranking equation are denoted in the Figure as JRB, JRB_Linear and JRB_Loh10, respectively, where the number of the distribution is denoted as PO and Uni for poisson distribution and discrete uniform distribution, respectively. The results of the job creation time test is shown in Fig. 3 and Table 10. The results show that the JRB algorithms show higher improvement using poisson distribution for the both proposed ranking equations. For poisson distribution, JRB_Log10 shows improvement in multiplies equal 1.062 and 2.575 in total execution time and average waiting time respectively, where JRB_Linear shows improvement in multiplies equal 1.054 and 3.665in total execution time and average waiting time respectively. Moreover, For discrete uniform distribution, JRB_Log10 shows improvement in multiplies equal 1.0003and 1.1308 in total execution time and average waiting time respectively, where JRB_Linear shows improvement in multiplies equal 1.0001 and 1.2469 in total execution time and average waiting time respectively. Finally, the results indicate that the proposed ranking equations improve the performance of job ranking backfilling model in total execution time and average waiting time using different distribution. Besides that, logarithmic ranking equation show higher improvement in total execution time where linear ranking equation shows higher improvement in average waiting time. The results show an evidence that the proposed ranking equations can serve the jobs efficiently in high demand grid system.

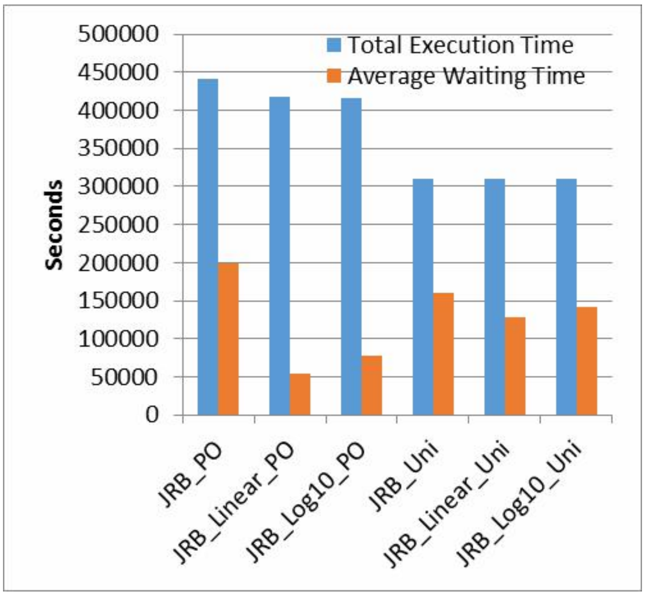

Fig. 3 - Total execution time and average waiting time using Poisson and discrete uniform distributions

Table 10 - Improvement between Proposed models and JRB model.

Distribution	Proposed Models	JRB			
		Improvement in Multiplies (x)		Improvement in Percentages (%)	
		TET	AWT	TET	AWT
Poisson	**JRB_Linear**	1.0543	**3.6649**	5.1489	**72.7143**
	JRB_Log10	**1.0617**	2.5752	**5.8149**	61.1685
Uniform	**JRB_Linear**	1.0001	1.2469	0.0102	19.8018
	JRB_Log10	1.0003	1.1308	0.0268	11.5653

(b) Resources Experimental Test

To understand the effect of the number of resources on the job scheduling models and the ability of the proposed ranking equations in enhancing the performance of a grid system under different conditions, an experimental test is used. This test uses different number of CPUs from 200 to 1000 CPUs with step of 100, where the rest of parameters are set to the default values. The results of the total execution time and average waiting time are shown in Figures 4 to 5, respectively. In all tests, the results show that JRB_Linear outperforms JRB model in total execution time and average waiting time. The range of improvement for total execution time and average waiting time are 1.004 to 1.110 and 3.387 to 3.691, respectively as shown in Table 11. In addition, JRB_Log10 outperforms JRB model in four cases (number of CPUs 200, 300, 400 and 1000) in total execution time and in all cases in average waiting time. The improvement in multiplies for total execution time and average waiting time are 1.003 to 1.014 and 3.119 to 2.568, respectively. The average results show that JRB_linear improves the total execution time and average waiting time compared with JRB_Log10, with improvement of multiples equal 1.026 and 3.551, respectively. Finally, the results indicate that the proposed ranking equation can improve job ranking backfilling using small and large demand grid system. Besides that, a linear ranking equation can efficiently improve job ranking backfilling model compared with logarithmic ranking equation in small and large demand grid system.

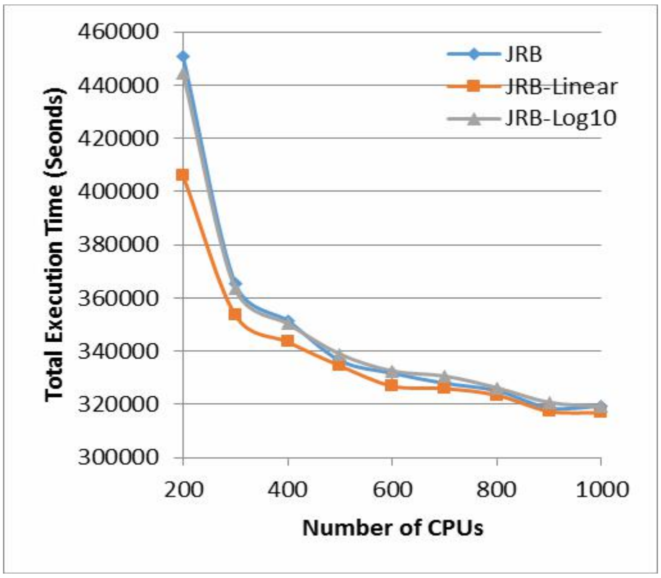

Fig. 4 - Total execution time, number of CPUs test

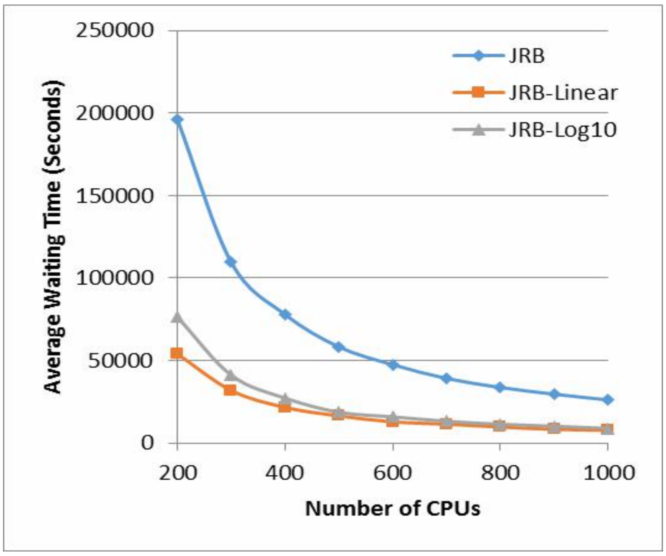

Fig. 5 - Average waiting time, number of CPUs test.

Table 11
Improvement between Proposed models and JRB model using different number of CPUs.

	CP U	Improvement in Multiplies (x)		Improvement in Percentages (%)	
		TET	AWT	TET	AWT
JRB-Linear	200	1.11014	3.63028	9.92108	72.45388
	300	1.03319	3.43858	3.21196	70.91823
	400	1.02348	3.60004	2.29390	72.22250
	500	1.00592	3.54110	0.58829	71.76020
	600	1.01488	3.69121	1.46573	72.90858
	700	1.00599	3.42530	0.59573	70.80551
	800	1.00527	3.46295	0.52386	71.12286
	900	1.00403	3.56525	0.40157	71.95150
	100	1.00789	3.38665	0.78311	70.47226
	0	**1.02585**	**3.55077**	**2.51968**	**71.83710**
JRB-Log10	200	1.01405	2.56839	1.38590	61.06511
	300	1.00529	2.66008	0.52609	62.40719
	400	1.00325	2.85879	0.32364	65.02019
	500	0.99281	3.11895	-0.72469	67.93790
	600	0.99745	2.98543	-0.25574	66.50397
	700	0.99176	2.96240	-0.83042	66.24357
	800	0.99664	2.95424	-0.33738	66.15032
	900	0.99353	2.91555	-0.65141	65.70113
	100	1.00039	2.95018	0.03883	66.10375
	0	**1.00008**	**2.77068**	**0.00787**	**63.90774**

(c) Number of Jobs Test

This experiment aims to test the efficiency of the proposed ranking equations compared with different number of jobs (different datasets). The test uses 10 different datasets from 1000 to 10000 with step of 1000, where the rest of parameters are set to the default values. The results is shown in Figures 6 to 7 for total exertion time and average waiting time, respectively, where the improvement results are calculated in Table12. For total execution time, the JRB_Linear and JRB_Log10 results show that the range of improvement in multiplies is between 1.017 to 1.148 and 1.008 to 1.062, respectively. Moreover, on average waiting time the JRB_Linear and JRB_Log10 results show that the range of improvement in multiplies is between 3.368 to 4.239 and 2.324 to 2.870, respectively. On average, the JRB_Linear shows improvement in total execution time and average waiting time with improvement in multiplies equal to 1.047 and 3.614, respectively, where JRB_Log10 shows improvement in multiplies in average waiting time only with value equal 2.609. Finally, overall results indicate that the proposed ranking equations can improve job ranking backfilling model under high and low demand environment by using different number of jobs.

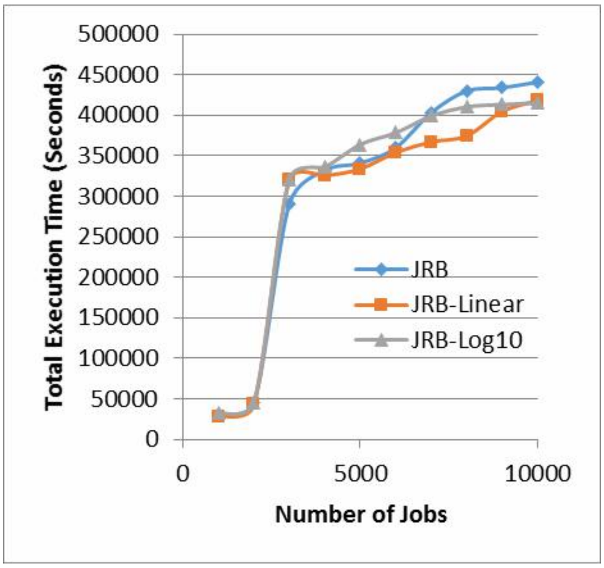

Fig. 6 - Total execution time, number of jobs test.

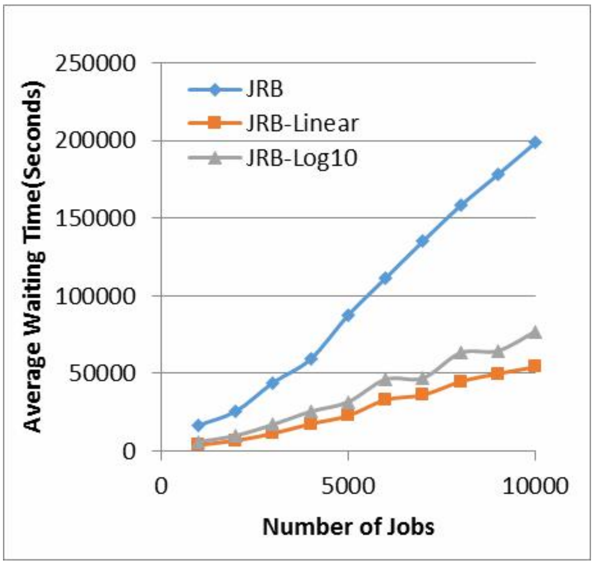

Fig. 7 - Average waiting time, number of jobs test.

Table 12 Improvement between Proposed models and JRB model using different number of jobs.

		J	
	Jobs	Improvement in Multiplies (x) TET	Improvement in Percentages (%) TET
JRB-Linear	10	1.093504	8.5509
	00	4.23884	76.4086
	20	2	5
	00	1.063074	5.933185
	30	3.75015	73.3344
	00	3	2
	40	0.908937	-10.0187
	00	3.85976	74.0917
	50	9	1
	00	1.021096	2.065988
	60	3.37526	70.3726
JRB-Log10	10	0.962747	-3.86945
	00	2.77604	63.9775
	20	6	4
	00	1.007962	0.789909
	30	2.56743	61.0506
	00	6	3
	40	0.905412	-10.4469
	00	2.56632	61.0337
	50	6	9
	00	0.987711	-1.24417
	60	2.3241	56.9742

5. Conclusion and Future Work

This paper focuses on how to improve the performance of job ranking backfilling (JRB) algorithm by changing the ranking equation of the smallest job. The new ranking equations are designed using optimal curve estimation by considering two functions including linear and logarithmic functions. Besides that, the proposed ranking equations are used with conventional JRB after changing the ranking equation to study the performance of the new JRB. The results indicate that changing the ranking equation would have significant improvement in total execution time and average waiting time of scheduling system with high performance when linear ranking equation. Besides that, the results of linear ranking equation outperform logarithmic ranking equation, which indicates that it would be a linear relationship between number of CPU and memory size with run time of job. For future work, more analysis and discussion are needed to explore the relationship between job's variables and run time of job, to build a precise model that can improve the performance of grid system. In addition, more metrics should be added to cover more performance metrics in the grid environment such as bandwidth and resource utilization, energy consumption and so on.

References

[1] Rodero, I., Guim, F., & Corbalan, J. (2009, June). Evaluation of coordinated Grid scheduling strategies. In 11th IEEE International Conference on High Performance Computing and Communications, 2009. HPCC'09. pp. 1-10.

[2] Ahmad, E. S., Ahmad, E. I., & Mirdha, E. S. (2017). A Novel Dynamic Priority Based Job Scheduling Approach for Cloud Environment. In International Research Journal of Engineering and Technology (IRJET). Vol 4, issue 6. pp.518-522.

[3] Da Silva, R. F., Glatard, T., & Desprez, F. (2013, August). Workflow fairness control on online and non-clairvoyant distributed computing platforms. In European Conference on Parallel Processing, Springer, Berlin, Heidelberg. pp. 102-113.

[4] Fanfakhri, A. B. M., Yousif, A. Y., & Alwan, E. (2017). Multi-objective Optimization of Grid Computing for -Performance, Energy and Cost. Kurdistan Journal of Applied Research, 2(3), pp74-79.

[5] Zhang, L., Chen, Y., Sun, R., Jing, S., & Yang, B. (2008). A task scheduling algorithm based on PSO for grid computing. *International Journal of Computational Intelligence Research*, *4*(1), pp. 37-43.

[6] Idris, H., Ezugwu, A. E., Junaidu, S. B., & Adewumi, A. O. (2017). An improved ant colony optimization algorithm with fault tolerance for job scheduling in grid computing systems. PloS one, 12(5), e0177567.

[7] Ghosh, T. K., Das, S., Barman, S., & Goswami, R. (2017). Job scheduling in computational grid based on an improved cuckoo search method. International Journal of Computer Applications in Technology, 55(2), pp. 138-146.

[8] Singh, P., Dutta, M., & Aggarwal, N. (2017). A review of task scheduling based on meta-heuristics approach in cloud computing. Knowledge and Information Systems, pp. 1-51.

[9] Gomathi, S., & Manimegalai, D. (2011, July). An adaptive grouping based job scheduling in grid computing. In International Conference on Signal Processing, Communication, Computing and Networking Technologies (ICSCCN), pp. 192-196.

[10] Sharma, R., Soni, V. K., Mishra, M. K., & Das, S. (2010, July). A Secure Resource and Job scheduling model with Job Grouping strategy in Grid computing. In 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), 2010, Vol. 4, pp. 329-333.

[11] Liu, Q., & Liao, Y. (2009, March). Grouping-based fine-grained job scheduling in grid computing. In. First International Workshop on Education Technology and Computer Science, 2009. ETCS'09, vol. 1, pp. 556-559.

[12] Kuzmanovska, A., Mak, R. H., & Epema, D. (2016, May). Koala-f: A resource manager for scheduling frameworks in clusters. In 2016 16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), pp. 80-89.

[13] Cunha, B., Madureira, A., Pereira, J. P., & Pereira, I. Evaluating the Effectiveness of Bayesian and Neural Networks for Adaptive Scheduling Systems. In 2016 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1-6.

[14] Mehrsai, A., Karimi, H. R., Rügge, I., & Scholz-Reiter, B. (2011, September). Application of learning pallets for real-time scheduling by use of artificial neural network. In 5th International Conference on Software, Knowledge Information, Industrial Management and Applications (SKIMA), pp. 1-7.

[15] Klusácek, D., & Rudová, H. (2012, May). Performance and fairness for users in parallel job scheduling. In *Workshop on Job Scheduling Strategies for Parallel Processing*. pp. 235-252. Springer, Berlin, Heidelberg.

[16] Toporkov, V. V., Toporkova, A., Tselishchev, A., Yemelyanov, D., & Potekhin, P. (2015). Multi-Level Job Flow Cyclic Scheduling in Grid Virtual Organizations. *Procedia Computer Science*, *51*, pp. 845-854.

[17] Lokhande, S. F., Chavhan, S. D., & Jadhao, S. R. (2015). Grid computing scheduling jobs based on priority using backfilling. IJEECSE, 2(2), pp. 68-72.

[18] Kaur, A., & Chhabra, A. (2017). A Review on Various Job Scheduling Algorithms. *International Journal of Computational Intelligence Research*, *13*(3), pp. 359-367.

[19] Cao, T., Huang, W., He, Y., & Kondo, M. (2017, May). Cooling-Aware Job Scheduling and Node Allocation for Overprovisioned HPC Systems. In 2017 IEEE International Parallel and Distributed Processing Symposium (IPDPS), pp. 728-737.