A New Fuzzy based Evolutionary Optimization for Job Scheduling with TLBO

Ch. Srinivasa Rao
RVR & JC College of Engineering
Guntur, India

B. RaveendraBabu, Ph.D.
VNR VignanaJyothi Institute of Engineering & Technology,
Hyderabad, India

ABSTRACT
Grid computing is a frame work that shares data, storage, computing across heterogeneous and distributed locations to meet the current and growing computational demands. Thispaper proposes a novel evolutionary optimization approachusing fuzzy Teaching Learning Based Optimization (TLBO) for resource scheduling in computational grids. The fuzzy TLBOgeneratesan efficient schedule to complete the jobs within a minimum period of time. The performance of the proposed fuzzy based TLBOalgorithm evaluate with various other nature heuristic algorithms, GeneticAlgorithm (GA), Simulated Annealing (SA), Differential Evolution, and fuzzy PSO. Experimental results have shown the efficiency and prominence of new proposed algorithm in producing optimal solutions for the selected benchmark job scheduling problems compared to other algorithms.

Keywords
Grid Computing, Job Scheduling, TLBO

1. INTRODUCTION
The increasing scientific demand in the new millennium can be satisfied by the Grid computing and Grid technologies. Grid computing facilitates a large computing capacity to solve complex problems by sharing data, storage, computing across geographically dispersed area[1]. The geographically distributed computing resources are linked through the internet in a Grid-like manner to complete computing in less time than before. Though, Grid Computing has wide variety of application areas including medicine, science, and research areas, it has many challenges. One of the most important challenges is job scheduling in computational grid. Krauter et al. provided a useful survey on grid resource management systems, in which most of the grid schedulers such as AppLes, Condor, Globus, Legion, Netsolve, Ninf and Nimrod use simple batch scheduling heuristics [2]. According to FatosXhafa and Ajith Abraham, heuristic and meta-heuristic methods are efficient in solving many complex problems, also useful in the Grid computing domain, especially for scheduling and resource allocation [3]. They have analyzed why the meta-heuristic methods are appropriate for job scheduling in computational grids several heuristic algorithms are available in the literature. Among these one important group is evolutionary algorithms (EA). Some of the Evolutionary Algorithms are Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Differential Evolution (DE).Jarvis et al. ProposedGA for scheduling jobs and compared with FCFS by the minimization of makespan. It has proved that GA can generate a good quality solution than batch scheduling heuristics [4]. Braun et al. evaluated the efficacy of batch queuing heuristics, tabu search, genetic algorithm, and simulated annealing in minimizing makespan [5].

Fuzzy particle swarm optimization (PSO) algorithm is proposed by Hongbo Liu et al. for scheduling jobs [6]. The PSO outperforms the genetic algorithm and simulated annealing approaches. Recently, Srinivasarao and RaveendraBabu developed a DE based solution for job scheduling algorithms [7]. The same authors developed a fuzzy based DE for scheduling jobs [8]. Though, these algorithms provide a near optimal solution for such complex problems, most of these requires number of control parameters in advance such as crossover rate and mutation rate, which influences the effectiveness of the solution. Determining the optimum values for these controlling parameters is very difficult in practice.Rao and Kalyankar,Rao and Patel have introducedthe Teaching-Learning-Based Optimization (TLBO) algorithm which does not require any algorithm specific parameters [9] [10] [11] [12] [13] [14] [15]. TLBO is developed based on the natural phenomena of teaching and learning process of a class room. TLBO contains two phases - teacher phase and learning phase [16].Like many population based algorithms, the TLBO also contains population. Solution vectors are the learners and dimensions of each vector is termed as subjects. Best learner in the population is a teacher.As Fuzzy provides more prominent solutions compared to complex approaches, it is developed a new fuzzy based TLBO for the job scheduling problems. This paper proposes the fuzzy based TLBO and evaluates the performance of the proposed with four different data sets varying in size and capacity. The experimental results showed the improved performance of the proposed algorithm.

2. SCHEDULING PROBLEM
A Schedule is a specification of mapping of the jobs to specific time intervals of the grid resources. The grid job scheduling problem consists of scheduling *m* jobs with given processing time on *n* resources. Let $J_j(j = \{1, 2, 3,\ldots,m\})$ are the independent user jobs on $R_i(i = \{1, 2, 3,\ldots,n\})$ are the heterogeneous resources aimed to minimize the time of completion by utilizing the nodes efficiently. The speed of each resource is expressed in number of cycles per unit time (CPUT). The length of each job is expressed in number of cycles. The information related to job length and speed of the resource is assumed to be known, based on user supplied information, experimental data and application profiling or other techniques [17].

The objective of the proposed job scheduling algorithm is to minimize the makespan. Makespan is a measure of the throughput of the heterogeneous computing system. Let $C_{i,j}(i\in\{1,2,...,n\}, j\in\{1,2,...,m\})$ be the completion time that the resource R_i finishes the job J_j , $\sum C_i$ represents the time that the resource R_i finishes all the jobs scheduled for itself. Makespan is defined as $C_{max} = \max \{\sum C_i\}$ [18].

3. GRID JOB SCHEDULING ALGORITHM USING FUZZY TLBO

TLBO is a nature inspired heuristic algorithm which does not require any program specific parameters compared to other existing evolutionary algorithms [19] [20]. The process of proposed fuzzyTLBOfor job scheduling is as follows:

1. Create an initial Class randomly, X of size NP in which each learner, X_i represent a membership matrix as follows, where X_{ijk} represents the degree of membership of the j^{th}resource to the k^{th}job.

$$\text{Membership matrix } (X_i)=\begin{bmatrix} X_{i,1,1} & X_{i,1,2} \cdots & X_{i,1,n} \\ X_{i,2,1} & X_{i,2,2} \cdots & X_{i,2,n} \\ \vdots & \vdots & \vdots \\ X_{i,m,1} & X_{i,m,2} \cdots & X_{i,m,n} \end{bmatrix}$$

for each element in the matrix X_i, the element

$X_{ijk} = \mu_R(R_j, J_k),\ j\epsilon\{1,2,\dots,m\},\ k\epsilon\{1,2,\dots,n\}$. μ_R is the membership function, the value of $X_{i,j,k}$ means the degree of membership that the grid node R_j would process the job J_k in the feasible schedule solution. In the grid job scheduling problem, the elements of the solution must satisfy the following conditions:

$X_{ijk} \in [0,1], j \in \{1,2,\dots,m\}, k \in \{1,2,\dots,n\}$.

$$\sum_{j=1}^{m} X_{ijk} = 1, j \in \{1,2,\dots,m\}, k \in \{1,2,\dots,n\}.$$

2. Compute makespan of each individual.
3. Determine Teacher, a best learner (with minimum makespan) from step 2.
4. Calculate mean of the Class.
5. A new set of improved learners can be generated by adding a difference of Teacher and mean of the class to each learner in the current generation, t, as follows

$$X_i(t+1) = X_i(t) + r*(X_{best}(t) - T_F M(t))$$

T_F is the teaching factor with a random value between 1 and 2. The value of T_F can be computed using the following equation

$$T_F = round(1 + rand(1))$$

6. The knowledge of the learners can be increased by the interaction of one another in the class. For a learner, i, another learner is selected, j, randomly from the class and the performance of learner i can be improved as follows

$$X_i(t+1) = \begin{cases} X_i(t) + r*(X_i(t) - X_j(t)), if & f((X_i(t)) < f(X_j(t)) \\ X_i(t) + r*(X_j(t) - X_i(t)), if & f((X_j(t)) < f(X_i(t)) \end{cases} \quad \text{where}$$

$f(\cdot)$ is the objective function to be minimized.

7. Steps from 2 to 6 are repeated till a difference of fitness value of fittest individuals (best learner) in any two successive generations is less than 0.0001.Best learner is the best solution in the run.

4. EXPERIMENTAL RESULTS AND DISCUSSIONS

Four different datasets with Resource Job Pairs (3,13), (5,100),(8,60), (10,50) are considered to evaluate the performance of proposed algorithm. The data sets details are as follows:

1. R=[4 3 2], J=[6 12 16 20 24 28 30 36 40 42 48 52 60]
2. R =[12.4239 5.7513 8.1589 15.0673 11.5667], J =[22.8309 84.2453 63.6209 15.1097 22.2990 61.5055 63.7290 38.3067 58.3645 46.2396 6.3017 4.6641 32.6431 3.2605 39.6288 68.9454 11.0986 5.4632 62.0148 61.6370 3.5445 3.6028 20.6273 59.5180 7.6429 38.0217 63.8822 72.3282 69.8816 10.2397 46.5268 45.2992 36.6185 17.0534 68.2132 70.5229 73.2959 48.8817 56.3745 13.8626 46.1739 72.1565 89.4985 28.7640 26.9674 86.8291 24.7703 80.8774 91.0230 24.7256 25.4526 6.8759 9.6816 64.7999 20.7069 84.6992 19.0422 18.7377 99.4410 45.0995 35.3247 32.7933 37.7777 40.5375 59.9695 13.7352 5.7366 46.9426 87.2470 93.5552 27.9160 17.7094 87.5398 25.3123 65.2915 96.7550 67.1633 87.2973 2.9729 15.4270 82.2381 44.1563 89.2515 74.0210 69.3577 35.9190 18.2714 17.2500 20.7294 43.4002 85.8856 50.0445 81.9616 47.1554 46.8207 46.1675 42.3975 90.3578 2.5472 31.1458]
3. R =[15.3018 5.2359 10.4958 8.8038 14.4782 12.6694 8.39052.2591], J =[55.5640 45.5983 70.0676 62.8884 79.8925 95.7707 53.2139 88.2539 18.9497 98.0152 28.6018 26.7283 87.8227 74.2560 15.3788 3.1522 89.6020 21.5155 31.2749 66.8214 29.8720 47.9840 8.3486 98.8568 59.1136 43.5026 52.5202 34.7272 44.4248 24.1431 58.8211 76.5158 53.9227 64.7716 22.4888 39.2222 78.7662 68.7229 47.1873 57.6472 79.8326 7.7999 61.0812 6.9263 42.7067 31.8899 87.6880 3.4709 77.2591 97.1428 99.0281 79.3084 44.9885 50.8345 22.9684 65.0622 33.3635 96.0897 73.2099 42.3714]
4. R=[8.7921 4.2824 9.9593 6.1981 2.1677 5.0973 7.5023 5.8584 8.2262 8.9273], J =[47.5638 82.0766 67.2558 5.8846 42.8020 25.2504 10.3421 44.3888 38.4286 3.5581 12.9746 20.7079 16.7939 66.8965 89.0968 5.4739 67.4668 97.2160 63.5576 77.8423 75.9944 2.7355 68.7442 15.8149 83.6164 67.8997 22.3112 85.4078 10.2197 70.5414 96.8417 42.4647 58.7418 51.6142 26.5727 53.5337 60.8352 72.5179 7.9647 32.7610 37.5439 82.6506 76.7400 88.1984 5.4819 19.1094 59.1463 5.7235 80.4647 6.3488]

Table 2.Performance comparison of the algorithms using the parameter makespan

Algorithm	Resource Job Pair			
	(3,13)	(5,100)	(8,60)	(10,50)
GA	47.1167	85.7431	42.9270	38.0428
SA	46.6000	90.7338	55.4594	41.7889
PSO	46.2667	84.0544	41.9489	37.6668
DE	46.0500	86.0138	43.0413	37.5748
Fuzzy TLBO	46.0000	85.5519	41.7367	35.5262

Table 3.Performance comparison of the four algorithms with the Time of completion in seconds

Algorithm	Resource Job Pair			
	(3,13)	(5,100)	(8,60)	(10,50)
GA	302.9210	2415.9000	2263.0000	2628.1000
SA	332.5000	6567.8000	6094.9000	6926.4000
PSO	106.2030	1485.6000	1521.0000	1585.7000
DE	81.5203	435.8865	337.7940	346.3016
Fuzzy TLBO	102.7210	407.8231	412.9547	342.2213

Table 4: Performance comparison of the four algorithms with the standard deviation in 100 runs

Algorithm	Resource Job Pair			
	(3,13)	(5,100)	(8,60)	(10,50)
GA	0.7700	0.6217	0.4150	0.6613
SA	0.4856	6.3833	2.0605	8.0773
PSO	0.2854	0.5030	0.6944	0.6068
DE	0.2916	0.3146	0.5274	0.6722
Fuzzy TLBO	0	0.0209	0.1389	0.4386

The proposed fuzzy TLBO runs hundred times on each data set. Table-2 reports the average makespan of hundred runs from various algorithms for different resource job pairs.

Similarly, Table-3 and Table-4 demonstrate the time required in seconds to converge the solution in a single run and standard deviation of the makespan in hundred iterations.

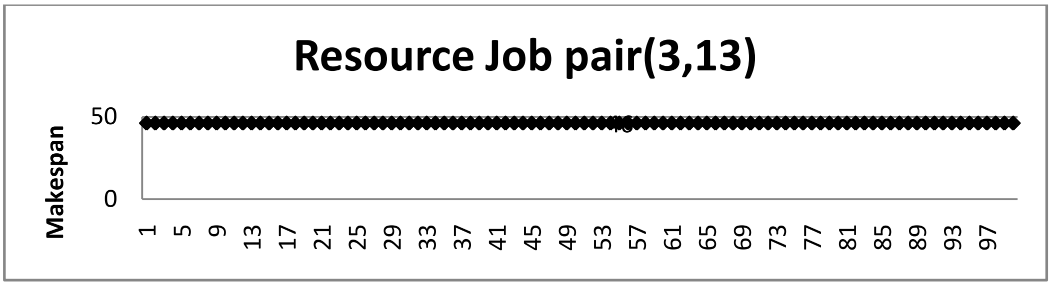

Figure-1 Makespan of (3,13)

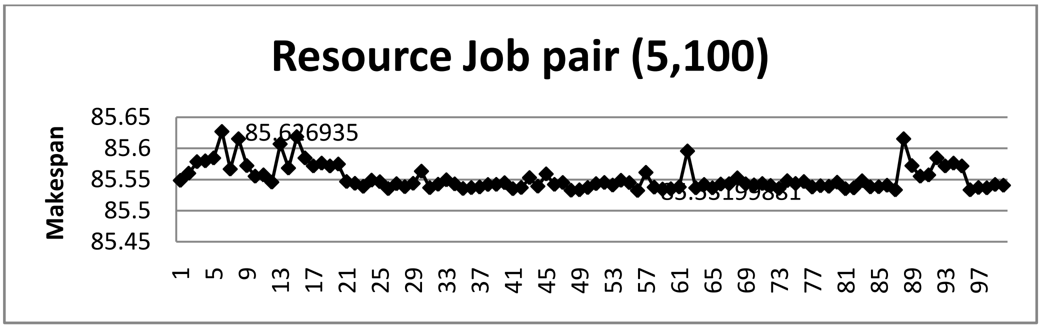

Figure-2 Makespan of (5,100)

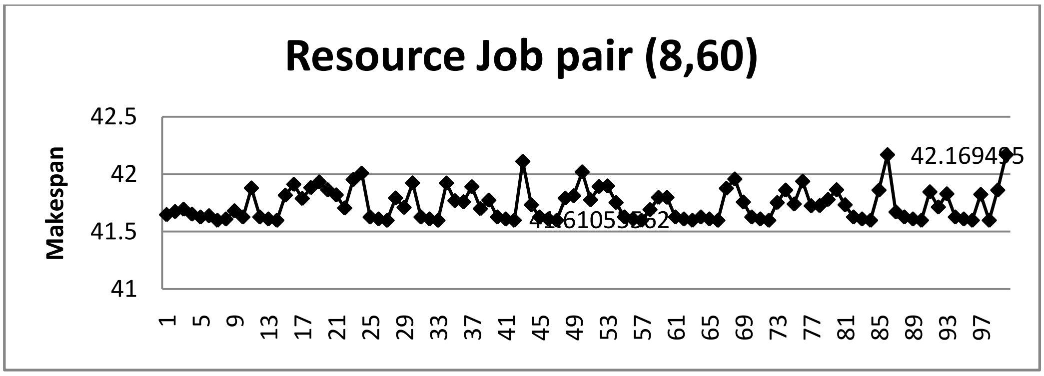

Figure-3 Makespan of (8,60)

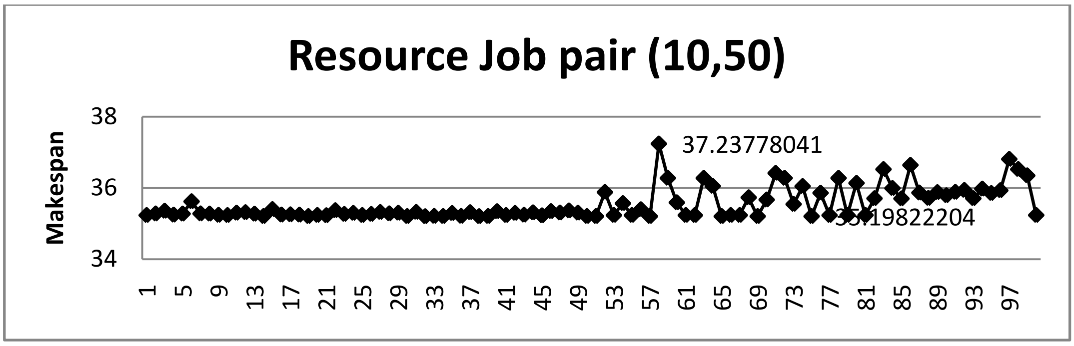

Figure-4 Makespan of (10,50)

The observed makespan of fittest individual in each of hundred runs is plotted in the following figures. Figure-1 contains resource job pair (3,13) makespan, Figure-2 reports (5,100), Figure-3 plots(8,60), and Figure-4 displays(10,50) makespan in various runs.

Table 5.Relative performance observed with various algorithms

Algorithm	**Resource Job Pair**				**Average**
	(3,13)	**(5, 100)**	**(8,60)**	**(10,50)**	
GA	1.1167	0.19119	1.19032	2.51657	1.253695
SA	0.6	5.18189	13.72272	6.26267	6.44182
PSO	0.2667	-1.49751	0.21222	2.14057	0.280495
DE	0.05	0.46189	1.30462	2.04857	0.96627

Table-5 depicted that overall improvement of Fuzzy TLBO over GA in all cases is 1.25, over SA is 6.44, over PSO is 0.28, and over DE observed that 0.97. Fuzzy TLBO is equally performs well with PSO and reported in Figure-9.

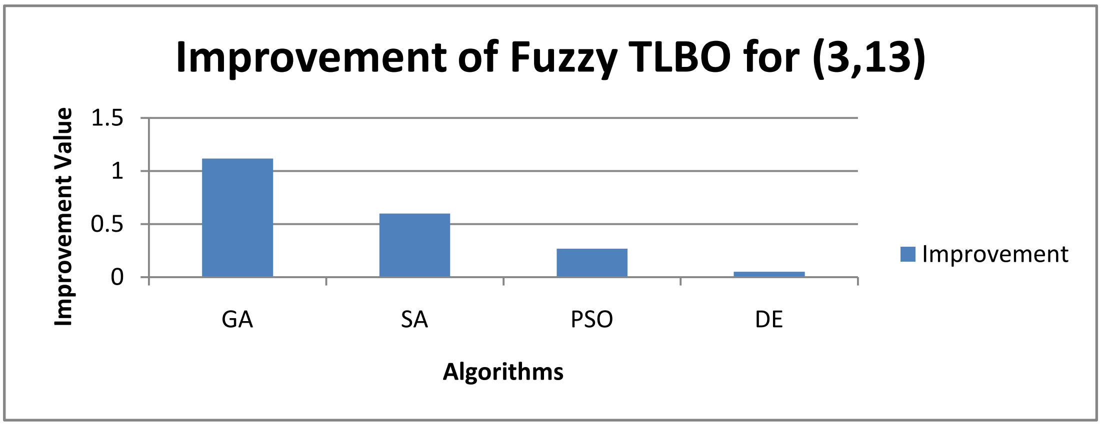

Figure-5 Improvable performance of Fuzzy TLBO for (3,13)

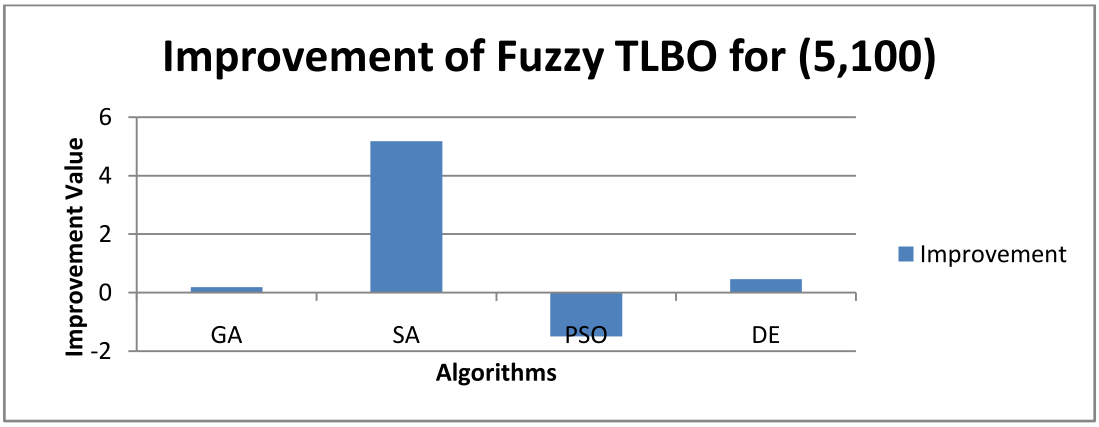

Figure-6 Improvable performance of Fuzzy TLBO for (5,100)

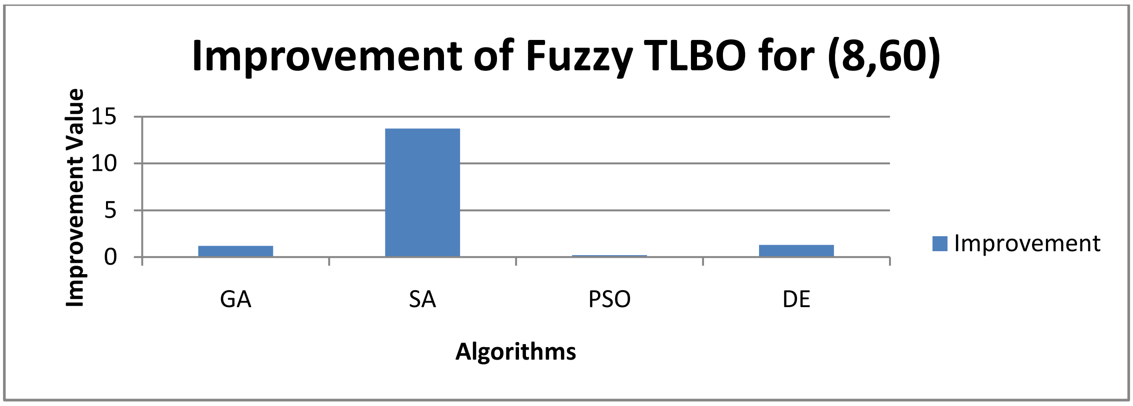

Figure-7 Improvable performance of Fuzzy TLBO for (8,60)

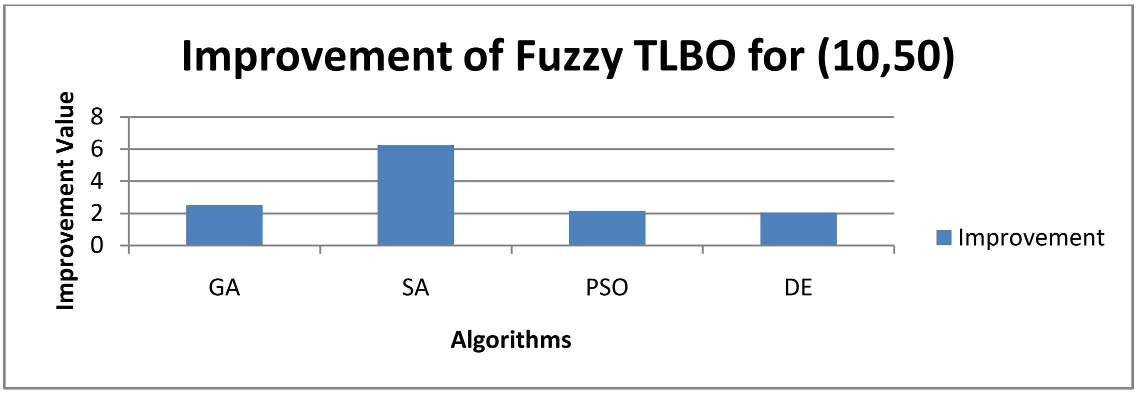

Figure-8 Improvable performance of Fuzzy TLBO for (10,50)

It is extended the study by reporting improved performance of Fuzzy TLBO towards makespan over other algorithms. From Figure-5, GA exhibits least performance for resource job pair (3,13). Figure-6 demonstrates that SA has shown highest makespan, whereas, Fuzzy PSO is better than Fuzzy TLBO by 1.49751. The improvable performance of PSO is more in the case of resource job pair (8,60) and it equally performs well with GA. From Figure-8, it is observed thatPSO and DE are equally performedwell and approximately DE improvement is 2.

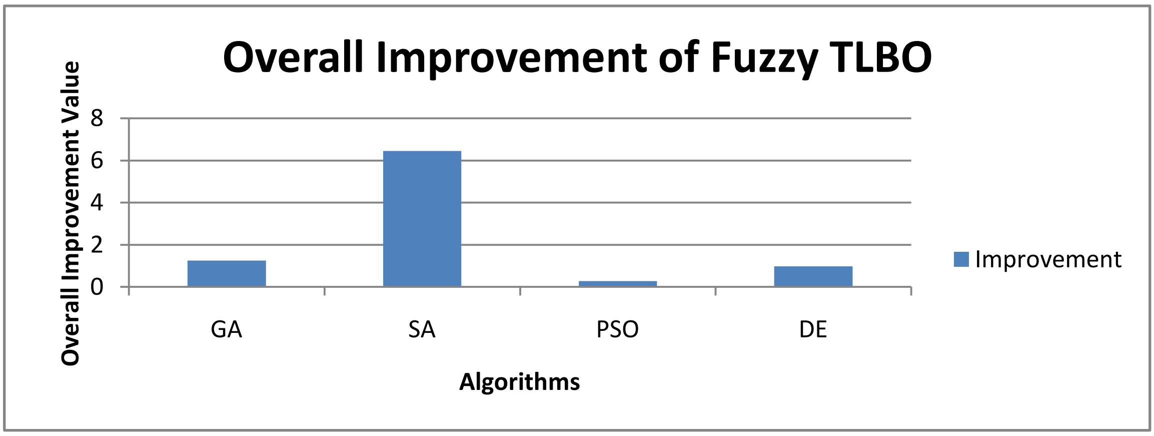

Fig.9 Relative performance of Fuzzy TLBO

5. CONCLUSIONS

The proposed Fuzzy TLBO has been developed incorporating fuzzy logic in Teaching Learning Based Optimization algorithm. The performance of Fuzzy TLBO is studied using various data sets and compared with various other evolutionary algorithms. The experimental results have shown that Fuzzy TLBO reported optimal solution in each case towards makespan. From the observation, Fuzzy TLBO is equally good with PSO. The future endeavour is to develop new algorithms which provide optimal solutions for various scheduling criteria, preemptive scheduling, etc.

6. ACKNOWLEDGMENT

We would like to thank, Professor K.KarteekaPavan for supporting the work proposed in this paper.

7. REFERENCES

[1] Job Scheduling in Grid Computing, KhushbooYadav, Deepika Jindal, Ramandeep Singh, International Journal of Computer Applications (0975 – 8887) Volume 69–No.22, May 2013, pp 13-16.

[2] K.Krauter, R.Buyya, M.Maheswaran,A taxonomy and survey of grid resource management systems for distributed computing, Software-Practice and Experience,32:135-164, 2002.

[3] FatosXhafa, Ajith Abraham, "Computational models and heuristic methods for Grid scheduling problems", Future Generation Computer Systems vol. 26, pp 608 – 621, 2010

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

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

[6] H.Liu, A.Abraham, A.E.Hassanien, Scheduling Jobs on computational grids using a fuzzy particle swarm optimization algorithm, Future Generation Computer Systems (2009).

[7] Ch.SrinivasaRao, B.RaveendraBabu, DE Based Job Scheduling in Grid Environments, Journal of Computer Networks, 2013 Vol. 1, No. 2, 28-31.

[8] Ch.Srinivasa Rao, B.Raveendra Babu, (2014) "A Fuzzy Differential Evolution Algorithm for Job Scheduling on Computational Grids" International Journal of Computer Trends and Technology (IJCTT), July 2014, Vol. 13, No. 2, Pg.72-77 ISSN:2231-2803 DOI:10.14445/22312803/IJCTT-V13P116.

[9] Rao, R.V. &Kalyankar, V.D. (2012a). Parameter optimization of modern machining processes using teaching–learning-based optimization algorithm. Engineering Applications of Artificial Intelligence, http://dx.doi.org/10.1016/j.engappai.2012.06.007.

[10] Rao, R.V. &Kalyankar, V.D. (2012b). Multi-objective multi-parameter optimization of the industrial LBW process using a new optimization algorithm.Journal of Engineering Manufacture, DOI: 10.1177/0954405411435865

[11] Rao, R.V. &Kalyankar, V.D. (2012c). Parameter optimization of machining processes using a new optimization algorithm. Materials and Manufacturing Processes, DOI:10.1080/10426914.2011.602792

[12] Rao, R.V. & Patel, V. (2012a). An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems. International Journal of Industrial Engineering Computations, 3(4), 535-560.

[13] Rao, R.V. & Patel, V. (2012b). Multi-objective optimization of combined Brayton and inverse Brayton cycle using advanced optimization algorithms, Engineering Optimization, doi: 10.1080/0305215X.2011.624183.

[14] Rao, R.V. & Patel, V. (2012c). Multi-objective optimization of heat exchangers using a modified teaching-learning-based-optimization algorithm, Applied Mathematical Modeling, doi:10.1016/j.apm.2012.03.043.

[15] Rao, R.V. & Patel, V. (2012d). Multi-objective optimization of two stage thermoelectric cooler using a modified teaching-learning-based-optimization algorithm. Engineering Applications of Artificial Intelligence, doi:10.1016/j.engappai.2012.02.016.

[16] Rao, R.V. &Savsani, V.J. (2012). Mechanical design optimization using advanced optimization techniques. Springer-Verlag, London.

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

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

[19] Rao, R.V., Savsani, V.J. &Vakharia, D.P. (2011).Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems. Computer-Aided Design, 43 (3), 303-315.

[20] Rao, R.V., Savsani, V.J. &Vakharia, D.P. (2012a). Teaching-learning-based optimization: A novel optimization method for continuous non-linear large scale problems. Information Sciences, 183 (1), 1-15.