Optimization of resource allocation in computational grids

Debashreet Das, Rashmita Pradhan, C.R.Tripathy

Dept. of Computer Science and Engineering,
Veer Surendra Sai University of Technology, Burla, Sambalpur

Abstract

The resource allocation in Grid computing system needs to be scalable, reliable and smart. It should also be adaptable to change its allocation mechanism depending upon the environment and user's requirements. Therefore, a scalable and optimized approach for resource allocation where the system can adapt itself to the changing environment and the fluctuating resources is essentially needed. In this paper, a Teaching Learning based optimization approach for resource allocation in Computational Grids is proposed. The proposed algorithm is found to outperform the existing ones in terms of execution time and cost. The algorithm is simulated using GRIDSIM and the simulation results are presented.

Keywords

Teaching-Learning Based Optimization (TLBO), Resource allocation, PSO, ACO

1. Introduction

Grid Computing is an emerging computing area with a high potential of storage capacity from heterogeneous sources embedded with computational power. The mechanism of Grid computing is a system where the distributed grids are inter-connected through wide-area networks [3]. The imperatives of application of Grid computation is proved from its acceptance and use by various important sectors to-day.

Generally, the Grid computing is classified into various forms like Computational Grids, Meta Grids, Smart Grids, Data Grids and Desktop grids [1,2]. Irrespective of the forms, Grid Computing encounters some challenges and one major challenge is allocation of resources [4]. Resource allocation is understood as the method of assigning (matching) each of the tasks to a machine and (ordering) scheduling the execution of the respective tasks on each machine. The resource comprises CPU cycles, memory, bandwidth, disk applications, data base on remote systems, the scientific data, etc. the principal goal of this activity is the management of the mentioned resources in an efficient manner so as to provide optimal services to the users. Further, adaptability to the ensuring changes with regard to availability of resources is also an important issue. Identifying the impediments to the goal of allocating the resources to the request of the user and selecting the appropriate resource to a particular task remains a challenge in the grid computing system. The reason being that, the resources are owned by various organizations and they

DOI:10.5121/ijgca.2015.6204

have their own resource-usage policy. It, therefore, requires defining the strategy for an efficient allocation of resources simultaneously considering the owner's usage policies. Thus, this paper is an attempt to study the existing resource allocation strategies adopted on computational grids, with an objective of

(i) How to minimize the computation cost and
(ii) Using the makespan, as an appropriate optimization technique.

The literature review disclosed various optimization techniques in grid computation such as Ant Colony Optimization[32], Genetic Algorithm[36,50], Particle Swarm Optimization[34,35] and Simulated Annealing[37]; which are incorporated in this study for accomplishment of the purpose of this study. In addition to this, a Teaching-Learning Based Optimization technique (TLBO) is studied to find out the efficacy of allocating resources in computational grids.

The paper comprises five sections:

Section-2 reviews the relevant research works so far,
Section-3 studies the efficacy of TLBO approach
Section-4 portrays the simulation results and
Section-5 presents the concluding remarks.

2. Related Work

The mechanism of resource allocation is one of the main challenges of Grid Computing. In the past, several research works have been done on resource allocation in grids. However, very limited attempts have been made to study the techniques used for optimization of the resource allocation in grids. In this section, the previous works on resource allocation in Grids using various optimization techniques are reviewed.

In [5,6] Foster et. al. defined the Grid system as a combination of heterogeneous resources which facilitates resource-sharing among a set of participants (some provide resources, others consume them). The Grid in essence, is expected to encompass the following three points. These are: (i) The co-ordinated resources are not subject to centralized control i.e., they run under the domain of virtual organization and (ii)The standards, protocols and interfaces used are standardized, open and for a general purpose i.e., the interoperability of the resources allows seamless integration with anything. The quality of service delivered is not trivial.

In [7, 8] it is reported that the resource allocation in grid computing systems is a NP complete problem. For the various economic approaches such as commodity market model, posted price model, bargaining model, tendering/contract-net model, auction model, bid-based , proportional resource sharing model, and bartering model for grid resource allocation were introduced [8].The different techniques for Grid resource allocation described in the literature can be categorized into two basic types: Static and Dynamic. A static based resource allocation constitutes a fixed data entry or fixed accounting scheme such as a fixed access to a computer node. Based on this approach, Tiboret. al.[9] have proposed a method. Their main objective was to assign an application's processes to the computing server that can present the required Quality of Service as well as execute the processes in a cost effective way. They presented a protocol to identify the computing servers that can execute the application with minimal cost as well as they can provide the required Quality of Service for the application. The resource exploration and the assignment process were modeled as a tree and the execution of a process took place through the search of a solution tree. In [9],the authors also came up with a protocol that allocated processes to the computing server. According to the approach of Somasundaram and Radhakrishnan[10], the incoming jobs from different users are collected and stored in a job list and the available resources are stored in the resource list. In their proposed algorithm, they took care of job's memory and the CPU requirements along with the priority of jobs and resources. Sulistioet. al.[11] proposed the Swift Scheduler(SS) using GridSim which maps jobs from the resource queue and the resources from the job queue with the help of some heuristic functions. According to the method in [11], the job allocations and the resource selection processes are executed using a heuristic searching algorithm. The said algorithm is based on Shortest Job first and it minimizes the average waiting time of jobs. As a result, the turn-around time is minimized and resource utilization is found to be more than the before. They carried out the Swift Scheduler test in GridSim through a number of jobs as well as resources against the total processing time, resource utilization and cost. Then, the average execution times on different resources and the data availability improved substantially because of the simple replication strategy. The work by Moreno[12] addressed the issues that the resource broker has to tackle processes like resource discovery resource selection, job scheduling , job monitoring and migration. In[13,14] a resource management system[RMS] was discussed and the models of grid RMS availability by considering both the failures of Resource Management(RM) Servers and the length limitation of request queues were developed. The resource management system (RMS) can divide service tasks into execution blocks (EB), and send these blocks to different resources. To provide a desired level of service reliability, the RMS assigns the same EB to several independent resources for parallel (redundant) execution.

A dynamic based resource allocation is a process whereby dynamic mechanisms adapt their participation conditions according to the change of available resource quantities. Based on this , Leila et.al.[15] illustrated how this method can be used by combining the best fit algorithm and the process migration. According to their approach, a resource reservation is decided by an administration based on the monitoring outcome specified by the system at a given time and the applications' requirements may dynamically be transformed at run-time. Berman et. al.[16] presumed a global grid network where resources are distributed all over the globe. In their approach, the users put forward applications to their local network scheduler. The scheduler afterwards allocates resources to each application taking into consideration the application's service level agreement without an administrator intrusion. There the scheduler selects resources related to the application requirements and allocates them to the requesting application. The resource manager links a separate thread for each registered grid application, while the resource observer daemon runs on each host to gather information regarding resources and to send them to the recorder database.

In literature, various other allocation approaches have also been proposed. The resource allocation in grids is generally possible through auction and commodity market based model. The work in [17] is described the use of utility functions for resource allocation using various optimization methods . They divide the optimization problem into two levels of sub-problems in order to reduce the computational complexity.

In [18] ,Buyya et. al. came up with a distributed computational economy-based framework, called GRACE ,for resource allocation and for regulation of available resources. Cui et. al.[19] recommended a price-based resource allocation model to maximize the aggregate utility of flows, by maximal clique associated outline prices for wireless channel entrée coordination. In[20] , a system was designed in the region of centralized broker which served as a platform for buyers and sellers to interact with each other. Whenthe nodes with resources intend to sell, initially they come online and registration is done on their own with the broker having unique ID. There, the buyer comes to the broker, looking for a resource with specific capacity and availability using its ID. The broker after that, searches its list of available sellers. The sellers may be the PC users, the dedicated storage providers, the companies or organizations.

Saeed et. al[21] introduced a novel market based algorithm for grid resource allocation where the grid resource allocation could be measured as a double auction in which the resource manager operate as an auctioneer or resource owners and the jobs act as buyers and sellers. Based on the said approach, resource allocation becomes an activity of each participant in the auction.

Wolskiet. al[22] introduced an auction based approach to allocate resources(CPU and disk storage). The basic idea behind the method is that the highest bidder gets the resource and the cost is found out by the bid price. In a double auction model, the consumer and provider submit bids and requests respectively during the time of trading. If at any time the bids and requests match, the trade is executed. A continuous double auction(CDA) based protocol checks a perfect match between the buyer and the seller. The immediate detection of compatible bids was proposed by Izakianet. al[23]. When no match is found, the task query object is stored in a queue till the time to live(TTL) expires or a match is found. A combinatorial auction based resource allocation protocol is a method, in which a user bids a price value for each of the possible combinations of resources required for its task execution [24]. It usesan approximation algorithm for solvingthe combinatorialauction anda grid resource allocation problem. A compensation based grid resource allocation is proposed in [25].

The resource allocation in grid environment is a complex undertaking due to its heterogeneity and dynamic nature aroused by wide area sharing. In the past, different optimization techniques were adopted by researchers. Buyyaet. al.[26] introduced an economic framework for grid. Due to this framework, one needs to pay financial cost for using resources to its owners. It leads to motivate resource owners to share their resources. Since then, a number of resource allocation algorithms have considered cost and economic profit in the objective function [27].

The authors in [28, 29] proposed a game theory and nash equilibrium method to optimize resource allocation while the work in [30] introduced a swift scheduler method. Daweiet. al[31] applied a novel heuristic, min-min algorithm and Ant Colony Optimization(ACO) algorithm which is a probabilistic technique for solving NP- Complete problems. ManpreetSingh [32] in his algorithm used multiple kinds of resources to balance resource utilization by minimizing the total execution time and cost. The said algorithm not only improves the performance of the system but also adapts to the dynamic grid system. Viswanathet. al.[33] used seeded genetic algorithm(SGA) to measure the performance through a local stochastic search procedure and Zhijie Li et.al[34,35] proposed Particle Swarm Optimization(PSO) resource allocation in grids where a grid system consists of a number of user- tasks that are needed to be assigned to different resources for execution, such that different user's objectives are optimized and the constraints with limited resources are satisfied. To solve this intractable problem, they proposed an algorithm with a universal utility function which combines both the time and cost to find out the optimal solution to resource allocation.

FatosXhafhaet. al.[36] proposed an experimental study on resource allocation in grids based on Genetic Algorithm(GA). They used two replacement strategies steady state GA(SSGA) and

Struggle GA(SGA), where SGA outperforms SSGA with its convergence process.The authors in[37] proposed a heuristic simulated annealing algorithm , that can be used to solve high-dimensional non-linear optimization problems for multi-site land use allocation(MLUA) problems. Their optimization model minimizes the development costs and maximizes the spatial compactness. The authors in [38] proposed a bi-objective optimization problem using Tabu Search, consisting of minimization of the makespan and flow time.

Many researchers have emphasized upon the importance of optimization of both the execution time and cost for computation of allocating resources. The selection of appropriate resources for a particular task is one of the major challenging work in the Computational Grids. The modern heuristic algorithms proposed in the literature for resource allocation in Grids include Genetic Algorithm (GA) ,the algorithm of Simulated Annealing(SA), Artificial Bee Colony algorithm(ABC)[43], Differential Evolution(DE)[44],Heuristic Search(HS)[45],Grenade

Explosion Method(GEM)[46],Intelligent Water Drop method(IWD)[47], Monkey Search(MS)[48] and Cuckoo Search(CS)[49]. In [40] it has been shown that the above mentioned search paradigms introduced several problems including performance limitation, problem in coding of network weight and selection of genetic operator. To overcome the limitations of GA and SA , the authors in [34] used PSO for resource allocation problem. In the said work, they have compared the PSO with GA and SA and have proved PSO to perform better. The notable characters of PSO are its fast convergence, less parameters to adjust and coding in real numbers.

The main limitations of the above mentioned heuristic techniques are that different parameters are required for proper working of these algorithms. Proper selection of the parameters is essential for the searching of the optimum solution by these algorithms. A change in the algorithm parameters changes the effectiveness of the algorithm. However, Genetic algorithm (GA) provides a near optimal solution for a complex problem having large number of variables and constraints. This is mainly due to the difficulty in determining the optimum controlling parameters like size of population, cross-over rate and mutation rate. The same is the case also with PSO which uses inertia weight, social and cognitive parameters. Also ABC[43] requires optimum controlling parameters of number of bees, limit, etc. The HS[45] requires harmony memory consideration rate, pitch adjusting rate and the number of improvisations. Hence, efforts need to be made to develop an optimization technique which is free from the above said problems.

In the above context, the Teaching- Learning based Optimization (TLBO)[40] is considered to be an effective soft computing tool due to some of its inherent advantages. The method TLBO in [40] can be applied for large scale non-linear optimization problems for finding the global solution.

Our major objective of resource allocation in grids is the effective allocation of heterogeneous resources to tasks. This in turn can achieve reduction of execution time and computation cost. However, as the number of task increases, the optimization of the objective function becomes more difficult. Under the above circumstances, there arises a need to formulate an objective function by taking both the above said problems into consideration.

In the next section, a TLBO based method is proposed to optimize the resource allocation in Computational Grids.

3. Application and Test:

The resource allocation in Grid computing system needs to be scalable, reliable and smart. It should be adaptable to change with regard to the allocation mechanism, depending upon the environment and its user requirements. Therefore, a scalable and optimized approach is essentially needed. In this section, a Teaching Learning Based Optimization approach for Resource Allocation in Grids (TLBORAG) is tested. The algorithm along with stepwise description is also presented.

3.1. The Teaching-Learning Based optimization in Grids

The Teaching-Learning based algorithm can be defined as a top-level, general search heuristics approach for arriving at an optimal solution [39,40,41]. It is one of the evolutionary algorithms that provide a better optimized solution to the search problem. Based on the number of inhabitants, it gets motivated from the traditional learning process from the school or elementary level. The Teaching-Learning method[42] consists of two phases. In the initial phase, students directly receive the information from the supervisor (teacher) where the students can interact in a face-to-face mode with the instructor. In the next phase, which is called Learner's phase, the students gain knowledge or information by interacting with their friends. This step can also be termed as un-supervised learning process. The best student thus found by the instructor in the initial phase, is assigned the duty of explaining the information or knowledge in a proper way to other students.

Here, the problem formulation is based on TLBO for optimization of resource allocation in Grids. The main aim in this paper was to find out the first objective function i.e. minimizing the cost and the time to allocate resources in a Computational Grid.

For the teaching learning based optimization technique, the programme is divided into two phases: Teacher's phase and Learner's phase. In this approach, the teachers are the tasks and learners are considered as the resources. The objective here is to obtain a global solution by allocating resources with minimum execution time and cost factor. For the TLBO, the population is considered as group (class) of learners[39,40] which is represented as a group of machines (resources) in Grid Computing. The teachers are assumed to impart information to learners. So they are represented as Grid broker or task factory containing number of tasks to distribute to the learners (resources) to be executed. The proposed and executed algorithm is named as TLBORAG algorithm.

Next, the proposed optimization technique was implemented through the following algorithm, then followed by an approach simulation using GridSim.

Teacher's Phase:

```
{
Initialize parameters of TLBO;
      Initialize parameters of grid task and grid resource;
For each learner(resource) in the population do {
   Randomly allocate task;
          Calculate the time and cost individually; }

*/Teacher's Phase */
```

For each learner (resource) in the population do {
Calculate the mean of the time and cost column-wise;
Calculate the fitness function;
Shift mean towards the minimum fitness (assumed as best teacher);
Find the difference mean and update solution;
Save solution if better;
}
/ Learner's Phase */*
While (stopping criteria not met) do
{
For each learner do {

Randomly select two learners (resources)
Calculate fitness value
Check for better fitness value
Update solution according to better fitness value obtained
}
Stop if stopping criteria satisfies;
Output time and cost for optimal allocation;
*/*Stop */*

3.2. Analysis of the Proposed algorithm:

A brief description of the algorithm TLBORAG is presented here.

Teachers Phase: The teacher (Grid broker or task factory) contains a set of tasks to distribute geographically all over and get it executed by the learners(resources or processing elements)according to the learner's level (resource capacity), which is practically not possible. On the other way, the teachers (task factory) can only move depending upon the capacity of the class (i.e. depending on the availability of resources). So, a process was carried out to solve this problem in an effective manner depending on many factors: firstly the mean at each iteration of the execution time and cost of tasks was calculated and then, tried to move the mean towards the required level. Then the solution was updated according to the existing method and a new mean was found. The difference was found between two means and was multiplied with a teaching factor. Accordingly the result was modified with the existing solution.

Learners Phase: Usually a learner (resource/machine) learns something new if the other learner has more knowledge than him (i.e., if a machine is more efficient or more suitable for the task to be carried out then the better resource is considered). So, a modification is needed by checking the fitness of two random learners (resource), to find out whose fitness value is better. From these two the better solution is accepted, or else the solution is modified and the two phases are carried out iteratively until the maximum generation is reached and the best solution is accepted. This procedure was followed to complement the Teachers phase in this study.

3.3. **Definition of the Problem**:

The step wise procedure for the implementation of the proposed TLBORAG algorithm for in Computational Grids is described below:

Step 1: *Defining the Problem*

The optimization problem was defined in the beginning and the optimization parameters were initialized.

The initialized population was the size of the number of learners(resources) and they were fixed in this problem. The other parameters were the number of generation (iterations) and design constraints (D_n) i.e., time and cost.

The optimization problem was defined as: Minimize f(x), where X ε x_i= 1,2,…………,D_n and f(x) was the objective function.

Since, a grid job is divided into "N" no. of tasks[34], the expression for allocation of resources in an optimal manner can be represented by an "N"-dimensional vector X_i =(x_{i1}, $x_{i2,\ldots\ldots\ldots,}$ x_{iN}), where, each element "x_{ij}" represents that the "j^{th}" task is allocated to resource "x_{ij}" for execution. The fitness function used [34] can be defined and represented using the following equation:

$$f(x) = \theta \ln\left(\max_{n=1}^{N}\sum_{r=1}^{R} t_r^n\right) + (1-\theta)\ln\left(\sum_{n=1}^{N}\sum_{r=1}^{R} e_r^n\right) +$$
$$\frac{1}{2c}\sum_{r=1}^{R}\left\{\left[\min\left(0, \mu_r + c\left(M_r - \sum_{n=1}^{N} m^n \left\| sign(x_{in} - r)\right| - 1\right|\right)\right)\right]^2 - \mu_r^2\right\}$$

Step 2: *Initialization of the population (set of the resources denoted as learners)*
A random population was generated according to the number of learners (resources) present and then resources were allocated to the tasks randomly at the initial stage .The execution cost and time was found out at the preliminary stage and stored in the matrix. The fitness value was also calculated for the random allocation and stored in a linear array.

Step 3: *Teacher's Phase:*
The mean of the constraints was found out at the preliminary stage (i.e. at first iteration) column wise, which gave the mean of the individual constraint (time and cost). This mean M_D was stored in the linear array. The task having the best minimum fitness value was considered to be the teacher for that iteration. The teacher tried to shift the mean value from M_D towards the minimum fitness value task so that the allocation was done with minimum execution time and cost value for that iteration. Now, the new mean M_{new} became the corresponding value of the time and cost from the minimum fitness found. The difference between the two means was found out by multiplying the teaching factor with the M_D and then subtracting the M_{new} from it .Then, the result is multiplied with a random number between [0,1].

Now, the result was added to the current solution i.e. the constraint values were updated with the formula

$X_{new} = X_{old}$ + Difference

If the learner's(resources) respective constraint values gave better fitness values than the previous one, then it was accepted, else, the new one was considered and the next phase was started.

Step 4: *Learner's Phase:*

two resources(learner) were selected randomly to carry out the task assigned by the teacher. Let the tasks be X_i and $X_j$where i≠ j. The fitness value of both the resources for execution of the task was calculated as: if $f(X_i) < f(X_j)$ then the solution to$X_{new} = X_{old} + r(X_i- X_j)$ was updated else it was updated to

$X_{new} = X_{old} + r(X_j- X_i)$. Iteration was performed until all the resources are checked and a solution having better function value was found. If the better function value solution was found, then it was accepted as X_{new}.

Step 5: Termination criterion:
if the maximum iteration is reached till 100 and the result is noted, the algorithm was terminated, otherwise from Step 3 onwards, the steps were executed until a better function value for the above problem was obtained.

3.3. Simulation:

The main aim of this experiment was to demonstrate the effectiveness of Teaching Learning based optimization technique. The effectiveness of the technique was required to be evaluated in terms of execution cost at different circumstances i.e. having different number of tasks with different iterations. It is difficult to evaluate the cost and the performance parameter (makespan) for different situations due to the dynamic nature of Grid environment. Therefore, this work simulates a Grid Environment based on java-based discrete-event Grid simulation toolkit called GridSim .The toolkit provides facilities for modeling and simulating Grid resources and Grid users with different capabilities and configuration.

To simulate application and scheduling in GridSim environment requires the modeling and creation of GridSim resources and applications that model the tasks. For the sake of simplicity in analyzing the result, it is assumed that all the resources are stochastically similar within the respective groups with very small variances in their characteristics.

(i) Resource Modeling: Grid resources are modeled and simulated as many as different characteristics speed of processing, time zone, etc. The resource capability is defined as MIPS rating. It is varied with [3 to 5] Million Instruction Per Sec. For every Grid resource, local workload is estimated based on typically observed load conditions. Processing cost for each Grid resource is considered as (0,1,2) G$(Grid Dollar).

(ii) Application Modeling: Grid tasks are modeled as many as from 5 to 20. Each teacher consists of 5, 10 or 20 tasks. Each task is heterogeneous in terms of task length and input file size. Task length is varied randomly from [3 to 10] Million Instructions i.e., 0 to 10% random variation in task length is introduced to model heterogeneity in different tasks.

4. **Results and Discussions:**

The Table 1 compares the execution time and the cost of the proposed algorithm with PSO[34] for (task size, N=5 and for number of iteration=100) along with the fitness value for each allocation. The maximum value for execution time for TLBO is 9.76 seconds and the maximum value for cost of execution in terms of grid dollars is 11.01$ with a maximum fitness value of 20.77 whereas for PSO, the maximum value for execution time is found to be 10.95 seconds and maximum value for cost of execution in terms of grid dollars is 11.46$ with a maximum fitness value of 22.41

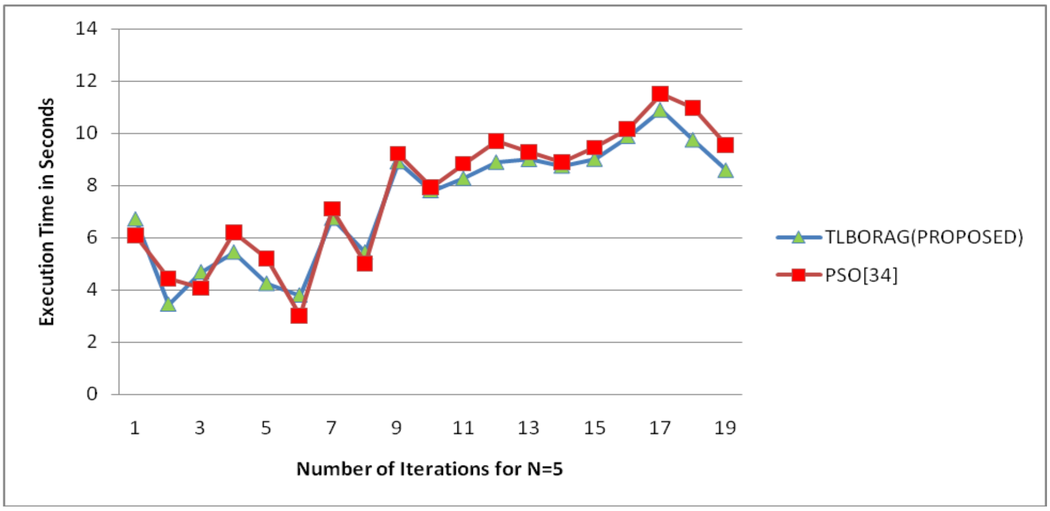

Figure:1 Comparison of Execution Time and cost(for task size N=5)

In Fig 1, we have taken the number of iterations in X-axis and the execution time in seconds in Y-axis. The graph gives a comparative view of execution time of our proposed algorithm with PSO , with the observation that our proposed algorithm gives same performance for small number of iterations with no. of tasks remaining constant at N=5. However, at the 19th iteration, it is observed that our proposed TLBO algorithm takes less time for execution compared to PSO.

Table 1:

Comparison of Execution Time and cost(for task size N=5)

TLBORAG[Proposed]			PSO[34]		
X		f(X)	X		f(X)
TIME(Sec)	COST(G$)		TIME(Sec)	COST(G$)	
6.72	2.436	9.156	6.09	2.9	8.99
3.45	5.214	8.664	4.45	5.83	10.28
4.69	6.92	11.61	4.08	7.32	12.12
5.45	1.452	6.902	6.2	2.54	8.74
4.25	8.324	12.574	5.2	8.82	14.02
3.8	7.41	11.21	3.02	7.91	10.93
6.75	9.325	16.075	7.1	10.20	17.3
5.45	4.698	10.148	5.01	6.32	11.92
8.92	2.587	11.507	9.21	3.12	12.12
7.81	2.147	9.957	7.92	2.95	10.87

8.28	1.09	9.37	8.83	3.99	12.82
8.90	2.7	11.6	9.69	4.68	14.37
9.01	3.6	12.61	9.26	4.78	14.04
8.76	3.94	12.7	8.9	4.85	13.75
9.02	4.2	13.22	9.45	5.45	14.9
9.89	6.4	16.29	10.15	8.45	18.6
10.91	8.02	18.93	11.50	9.38	20.88
9.76	11.01	20.77	10.95	11.46	22.41
8.59	11.56	20.15	9.54	12.49	22.03

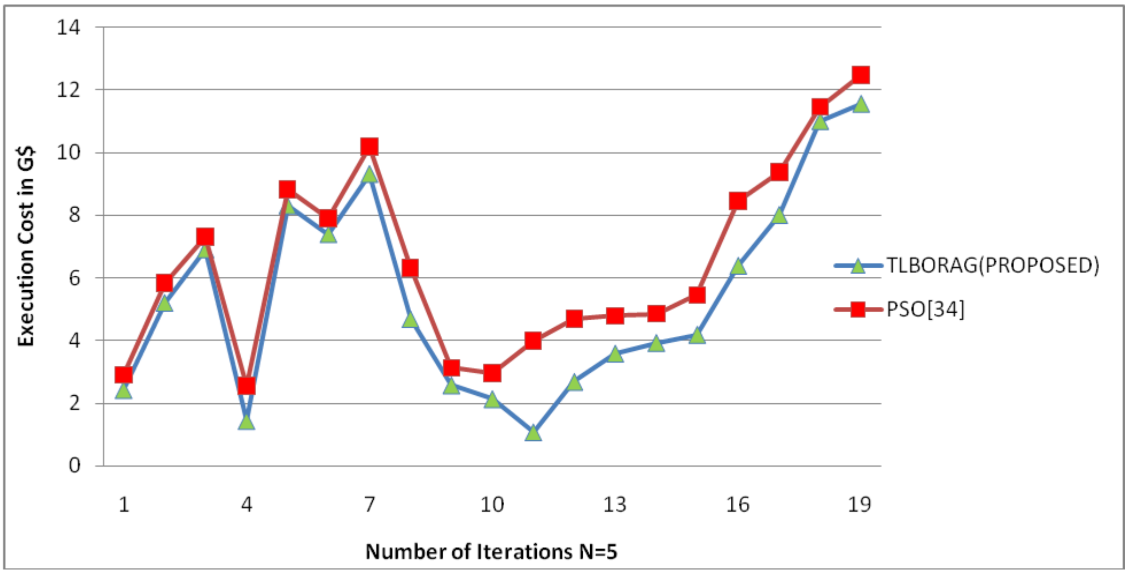

Fig 2 : Comparison of Execution Cost(N=5)

In Fig 2, we have taken the number of iterations in X-axis and the execution cost in Grid Dollars(G$) is taken in Y-axis. The graph gives a comparative view of execution cost of our proposed algorithm with PSO for 5 tasks, and it is observed that the TLBO consumes nearly less execution cost than the PSO[37] for N=5 tasks.

Table 2 :
Comparison of Execution time and cost(for task size=10)

TLBORAG[Proposed]			PSO[34]		
X		F(X)	X		F(X)
TIME	COST		TIME	COST	
1.63	5.91	7.54	2.40	7.57	9.97
2.24	6.61	8.85	3.36	8.62	11.98
3.18	7.14	10.32	4.21	9.69	13.9

3.39	7.28	10.53	5.68	9.99	15.67
3.81	8.69	12.5	4.79	10.63	15.42
5.67	9.49	15.16	6.23	11.42	17.65
6.65	10.67	17.32	7.14	12.12	19.26
6.71	11.78	18.49	8.89	13.79	22.68
6.92	12.36	19.28	8.96	14.82	23.78
7.36	13.82	21.18	9.27	15.67	24.94
8.68	13.91	22.59	10.63	15.92	26.55
9.14	14.80	23.94	11.49	16.01	27.5
10.06	15.28	25.34	12.38	17.83	30.21
10.29	17.38	27.67	12.77	20.42	33.19
11.68	20.43	32.11	13.04	22.67	35.71
11.94	21.67	33.61	13.82	23.14	36.96
12.26	23.86	36.12	14.11	26.28	40.39
12.79	24.49	37.28	14.99	27.76	42.75
13.82	25.66	39.48	15.63	28.16	43.79
13.96	28.39	42.35	16.69	34.34	51.03

The Table 2 shows the execution time and execution cost for higher number of tasks after the task size is increased to 10. Here our proposed algorithm gives better results than PSO[35] by observing the fitness values. The maximum value for execution time for TLBO is 13.96 seconds and cost is 28.39 G$ and fitness value is 42.35; the maximum execution time for PSO is 16.69 seconds and the execution cost is 34.34 G$ with fitness value of 51.03

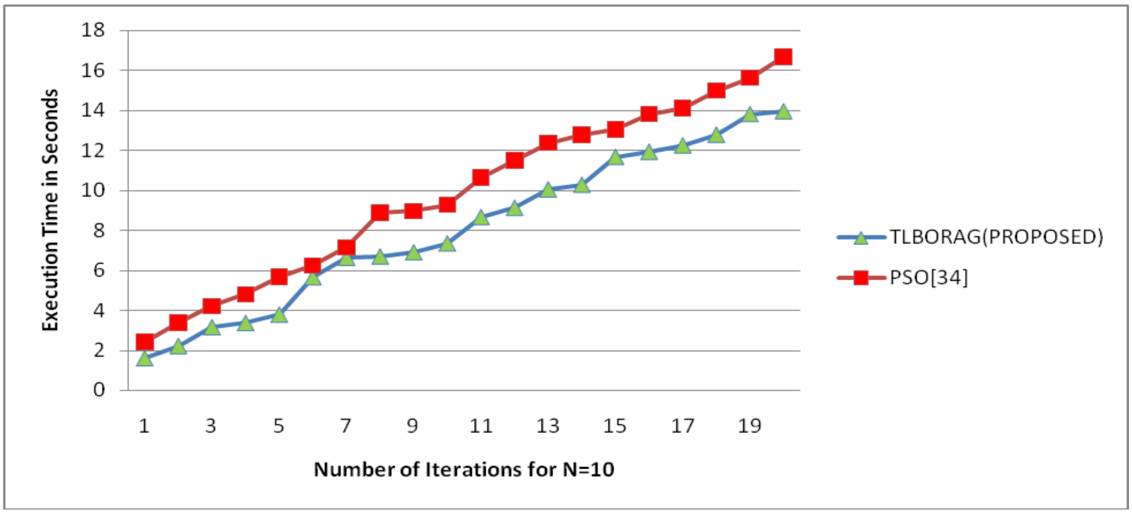

Fig 3: Comparison of Execution time(N=10)

In the Fig 3, we have taken the number of iterations in X-axis and the execution time in seconds in Y-axis. This graph gives a comparative view of the execution time of our proposed algorithm with PSO for 10 number of tasks, and it is observed that TLBO takes nearly less execution time than PSO for N=10 tasks.

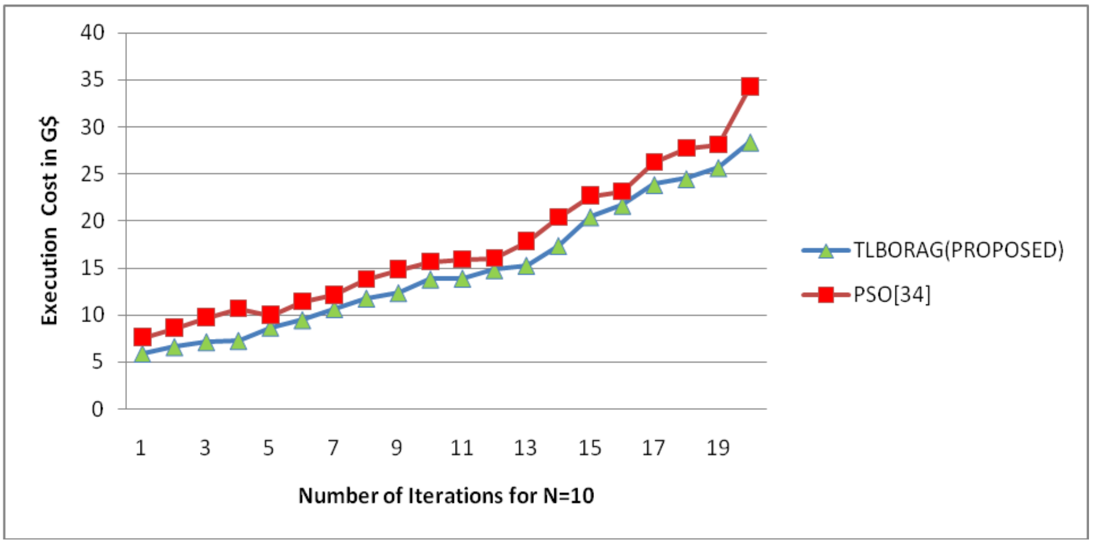

Fig 4: Comparison of Execution Cost(N=10)

In the Fig 4, we have taken the number of iterations in X-axis and the execution cost in Grid $ is taken in Y-axis. This graph gives a comparative view of execution cost of our proposed algorithm with PSO for 10 tasks, and it is observed that the TLBO consumes less execution time than PSO for N=10 tasks.

The table 3 shows the execution time and cost for higher number of tasks after the task size is increased to 20. Here, our proposed algorithm gives better result than PSO[35] by observing the corresponding fitness values. It is found that for TLBO, the maximum value for execution time is 10.16 seconds and maximum value for cost of execution in terms of grid dollars is 68.93$ with a maximum fitness value of 79.09 whereas for PSO, the maximum value for execution time is 14.77 seconds and maximum value for cost of execution in terms of grid dollars is 74.31$ with a maximum fitness value of 89.08

Table 3:
Comparison of Execution time and cost(for task size=20)

TLBORAG[Proposed]			PSO[34]		
X		f(X)	X		f(X)
TIME	COST		TIME	COST	
1.04	12.63	13.67	3.94	17.86	21.8
2.36	13.42	15.78	4.02	18.34	22.36
2.78	14.01	16.79	4.94	18.98	23.92
3.62	15.63	19.25	5.63	19.01	24.64
4.78	15.82	20.6	7.81	19.63	27.44

4.99	16.66	21.65	8.34	20.97	29.31
5.66	16.98	22.64	8.80	22.80	31.6
5.73	17.46	23.19	9.14	24.82	33.96
6.04	29.63	35.67	10.12	28.96	39.08
6.96	24.84	31.8	10.80	31.03	41.83
7.14	29.45	36.59	11.14	34.48	45.62
7.79	31.78	39.57	11.24	37.89	49.13
7.91	36.25	44.16	11.79	42.62	54.41
8.12	42.73	50.85	12.62	49.83	62.45
8.62	48.46	57.08	12.94	52.57	65.51
8.94	52.59	61.53	13.03	57.63	70.66
9.03	57.63	66.66	13.42	62.14	75.56
9.42	62.76	72.18	13.95	69.26	83.21
9.95	66.81	76.76	14.27	71.81	86.08
10.16	68.93	79.09	14.77	74.31	89.08

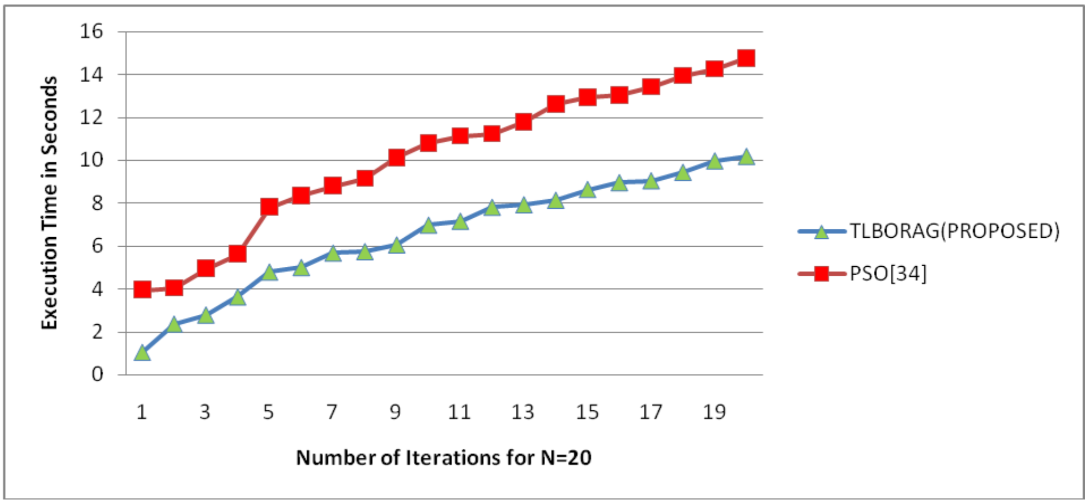

Fig 5: Comparison of Execution Time(for N=20)

In Fig 5, we have taken the number of iterations in X-axis and execution time in seconds is taken in Y-axis. This graph gives a comparative view of execution time of our proposed algorithm with

PSO for twenty tasks, and it is analysed that TLBO takes less execution time than PSO for N=20 tasks

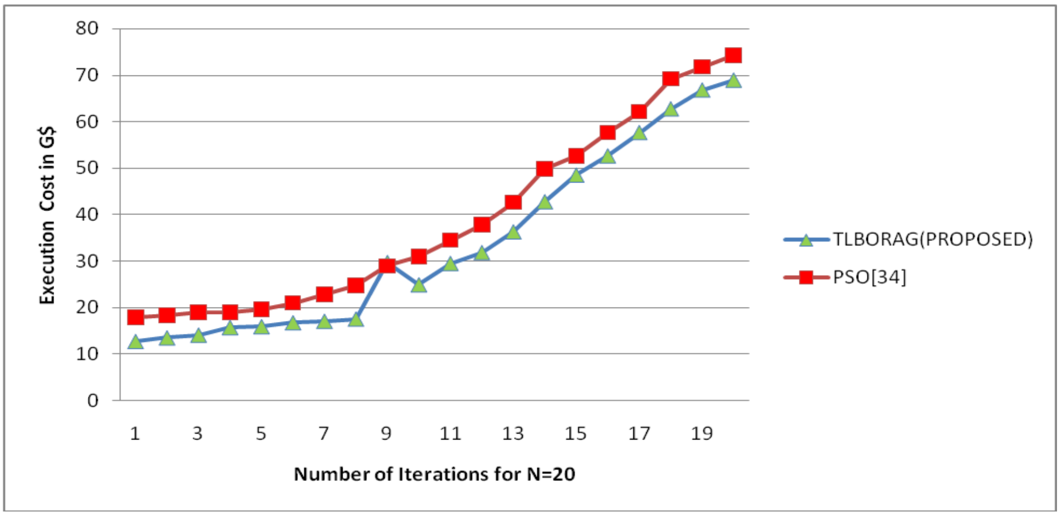

Fig 6 : Comparison of Execution Cost(N=20)

In Fig 6, we have taken the number of iterations in X-axis and execution cost in Grid $ is taken in Y-axis. This graph gives a comparative view of execution cost of our proposed algorithm with PSO for 20 tasks , and it is analyzed that the TLBO requires less execution cost than PSO for N=20.

5.Conclusion:

In this section, we discussed and compared the results with the execution time and execution cost for PSO[34] and TLBO. In PSO the time and cost is found to increase when task size is increased above the value of 12, but, in case of TLBO, we get a better solution with a better fitness value.

In this article, issues and challenges involved in resource allocation in Grid Computing have been addressed. The effectiveness of the proposed solution has been verified using GridSim Toolkit version 5.2 for simulation of heterogeneous resources, controllable and repeatable test environment.

In our approach, we have considered the makespan and cost as performance factors.

6.References:

[1] Chetty M, Buyya R, "Weaving Computational Grids: How Analogous Are They with Electrical Grids?", *IEEE Explore on Computing in Science and Engineering*, pp. 61-71, Vol.4, issue 4, July/August 2002

[2] Dong Fangpeng "Scheduling Algorithms for Grids", *Technical Report*, 2006

[3] I.Foster, C.Kesselman, and S.Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations", *International Journal of High Performance Computing Applications*, 2001

[4] Buyya R, Abramson D, and Giddy J, "An architecture for a Resource Management and Scheduling System in a Global Computational Grid", *Proceedings, the 4th International Conference on High Performance Computing in Asia-Pacific Region(HPC Asia 2000)* pp. 283-289, Vol. 1

[5] Carl Kesselman and Ian Foster , "The Grid 2: Blueprint for a New Computing Infrastructure", *Elsevier* Second edition, Nov. 2003

[6] Foster et. al "What is the Grid? A Three point Checklist", *Argonne National Laboratory and Universityof Chicago*, 20 July 2002

[7] Gunther S., Kurt V., and Jan B., "A Commodity Market Algorithm for pricing Substitutable Grid Resources", *Future Generation Computer Systems*, Vol(5): pp 688-701, 2007

[8] Buyya R., Abramson D., Giddy J. and Stockinger H., " Economic Models for Resource Management and Scheduling in Grid Computing", *The Journal of Concurrency and Computation*, 14 pp. 1507-1542,2002.

[9] TiborG., "A Resource Allocation protocol for providing Quality of Service in Grid Computing using a policy-based approach". *AICTE-ICIW Proceedings of the Advanced Int'l Conference onTelecommunications and Int'l Conference on Internet and Web Applications and Services, IEEE Computer Society, Washington DC, USA 2006*.

[10] Somasundaram K. and Radhakrishnan S., "Task Resource Allocation in Grid using Swift Scheduler", *International Journal of Computers, Communications and Control*,Vol. 4(2): 158-166,2009,

[11] Sulistio, Anthony U.C., Srikumar V., Borut R., and Rajkumar B., "A Toolkit for Modelling and Simulating Data Grids: An Extension to GridSim", *Concurrency and Computation: Practice andExperience(CCPE)*, 20(13): 1591-1609, Online ISSN:1532-0634, Printed ISSN:1532-0626, 2008

[12] Moreno R., Alonso-Conde A.B., "Job Scheduling and Resource Management techniques in Dynamic Grid Environments", *Vol. 2970, Lecture Notes in Computer Science, Springer*, pg. 25-32, 2004

[13] Dai Yaan-Shun ,Xie Min and Poh Kim-Leng, "Availability Modelling and Cost Optimization for the Grid Resource Management System*", IEEE Transactions on Systems, Man and Cybernetics, Part-A*, Vol. 38, Issue 1, Jan 2008, Pg. 170-179

[14] Dai Yuan-Shun and Levitin G., "Optimal Resource Allocation for Maximising Performance and Reliability in Tree-Structured Grid Services", *IEEE Transactions on Reliability*, Vol. 56, Issue 3, Sept. 2007, Pg. 444 – 453

[15] Leila I., "Dynamic Resource Allocation Mechanisms for Grid Computing", *Proceedings of IEEEInternational Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities(Trident – Com2007)*

[16] Berman F., Fox C.G., and Hey J.G.Anthony, "Grid Computing: Making the Global Infrastructure a Reality", *Wiley Series in Communications, Networking and Distributed System, 2003*

[17] Czajkowski K., Foster I., Karonis N., Kesselman C., Martin S., Smith W., and Tuecke S., " A Resource Management Architecture for Metacomputing Systems", *Lecture Notes in Computer Science*, Vol. 1459, pg. 62, Jan 1998

[18] Buyya R., Abramson D., Giddy J., "Nimrod/G: An architecture for a resource management and scheduling system in a global computational Grid", *International Conference on HighPerformanceComputing in Asia-Pacific Region(HPC Asia 2000)*, Beijing, China, IEEE Computer Society Press, USA, Pg. 283-289, Vol. 1

[19] Cui Y., Xue Y., and NahrstedtK.,"Optimal Resource Allocation in Overlay Multicast", *IEEE Transactions on Parallel and Distributed Systems*, Aug. 2006, Vol. 17, Pg. 808-823

[20] Krauter K., Buyya R., and Maheswaran M., " A Taxonomy and Survey of Grid Resource Management System for Distributed Computing", *IEEE Transactions on Software Engineering: Practice and Experience*, Vol. 32, Pg. 135-164, 2002.

[21] Parsa S., Shokri A., and Nourossana S., "A novel market-based grid resource allocation algorithm", *International Conference on Network Digital Technologies*, Tehran, Iran,Pg. 146-152,2009.

[22] R.Wolski, J.Brevik, J.S.Plank and T. Bryan, "Grid resource allocation and control using computational economics", *Berman F., Fox G., and Hey A., editors, Grid Computing: Making The Global Infrastructure a Reality, John Wiley & Sons*, 2003.

[23] Izakian H., Abraham Ajith and Ladani T.B., " An Auction Method for Resource Allocation in Computational Grids", *Future Generation Computer Systems*, 26(2): 228-235, 2010

[24] Das Anubhav and Grosu Daniel, "Combinatorial Auction-based Protocols for Resource Allocation in Grids", *19th IEEE International Symposium on Parallel and Distributed processing(IPDPS'05) – Workshop* 13, Vol. 14, 2005

[25] Wang X., TeoY.M.,andGozali J.P., "A Compensation-based Scheduling Scheme for Grid Computing", *In Proceedings of the Seventh International Conference on High Performance Computing and Grids in Asia pacific Region*, 2004.

[26] Buyya R., Giddy J., and Abramson D., "A case for economy grid architecture for service-oriented grid computing", *10th IEEE International Heterogeneous Computing Workshop(HCW 2001*), San Francisco, CA

[27] Young L., McGoughS.,Newhouse S., and Darlington J. , "Scheduling Architecture and Algorithms within the ICENI Grid Middleware", *UK e-Science All Hands Meeting, 2003, Pg. 5-12, Notitingham, UK*

[28] Osborne M.J. and A.Rubinstein, "A Course in Game Theory", *MIT Press, Cambridge, MA, London* 1994

[29] Chun-Tian C. and L.Zhi-Jie Li "Parallel algorithm for grid resource allocation based on nash equilibrium ", *IEEE International Conference on Machine Learning and Cybernetics*, pp. 4383-4388,2006

[30] Somasundaram K. and Radhakrishnan S. "Task Resource Allocation in Grid using Swift Scheduler", *International Journal of Computers, Communications and Control*, Vol. IV, 158-166, 2009.

[31] Dawei S., Guiran C., Lizhong J. and Xingwei W. "Optimizing Grid Resource Allocation by Combining Fuzzy Clustering with Application Preference", *International Conference on Advanced ComputerControl(ICACC)* 2nd pp:22-27,2010

[32] Singh Manpreet "Grid resource allocation based on ant colony optimization", *Journal of Advances inInformation Technology* ,Vol 1, pp:133-135 2010.

[33] Vishwanath A., Himanshu P. and Tracy M. "An Approximate algorithm for resource allocation using combinatorial auctions", *Intelligent Agent technology, IAT'06, IEEE/WIC/ACM International Conference,* pp:571-578,2006

[34] Zhijie Li, Xiangdong Liu, DuanXiaodong and Y. Jian " Grid resource Allocation Based on Particle Swarm Optimization",*IEEE 2nd International Conference on Power Electronics and Intelligent Transportation System*, pp. 93-96, 2009.

[35] Liu Hongbo, Abraham Ajith and Hassanien E.A. "Scheduling jobs on computational grids using a fuzzy particle swarm optimization algorithm",*Future Generation Computer Systems*, pp. 1336-1343, 2010

[36] XhafaFatos, Barolli L. and DurresiArjan "An Experimental Study on Genetic Algorithm for Resource Allocation in Grids",*Journal of Interconnection Networks*, Vol. 8, No. 4, pp. 427-443,2007

[37] Jeroen C., Aerts J.H., Heuvelink B.M. "Using Simulated Annealing for resource allocation" *International Journal of Geographical Information Science*, 2002

[38] XhafaFatos,Carretero "A Tabu Search Algorithm for Scheduling Independent Jobs in Computational Grids" ,*Computing and Informatics*, Vol. 28,pp. 1001-1014, 2009.

[39] Crepinsek M., Liu Shih-His and Mernik Luka, "A note on teaching-learning based optimization algorithm", *Journal of Information Sciences, Elsevier* , 2012 pp. 79-93

[40] Rao R.V., Savsani, Vakharia D.P., "Teaching-learning-based optimization: an optimization method for constrained mechanical design optimization problems", *Computer Aided Design*, 43(3)(2011) pp. 303-315

[41] Rao R.V., Savsani, Vakharia D.P., "Teaching-learning-based optimization: an optimization method for continuous non-linear large-scale problems", *Information Sciences*, 183(1)(2012) pp. 1-15

[42] Qu B.Y., Liang J.J., Suganthan P.N., "Niching particle swarm optimization with local search for multi-modal optimization", *Information Sciences*, 197(2012) pp.131-143

[43] Karaboga D., Basturk B., "On the performance of artificial bee colony algorithm(ABC)", Applied Soft Computing, 8(1) (2008)

[44] Storn R., Price K., "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces", Journal of Global Optimization 11(4)(1997) pp.341-359

[45] Lee K.S., Geem .W.Z., " A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice", Computer Methods in Applied Mechanics and Engineering, 194(36-38)(2005) 3902-3933.

[46] Ahrari A., AtaiA.A.,"Grenade explosion method – a novel tool for optimization of multimodal functions", Applied Soft Computing 10(4)(2010) 1132-1140.

[47] Hosseini-Shah.H,"The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm", International Journal of Bio-Inspired Computation 1(1-2)(2009) 71-79.

[48] Mucherino A., Liberti L.,Lavor C., MaculanN.,"Comparisons between an exact and a metaheuristic algorithm for the molecular distance geometry problem", In proceedings of the 11th Conference on Genetic and Evolutionary Computation,2009,pp.333-340.

[49] Yang X.,Deb.S.,"Cuckoo search via Levy flights", World Congress on Nature and Biologically Inspired Computing(2009) 210-214.

[50] Goldberg D.,"Genetic Algorithms in Search Optimization and Machine Learning", Addison-Wesley, 1989.