International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 21 (2017) pp. 10809-10815

Multi-Objective Virtual Machine Placement using Improved Teaching Learning Based Optimization in Cloud Data Centers

Dilip Kumar
Assistant Professor, Department of Computer Science and Engineering,
National Institute of Technology, Jamshedpur, Jharkhand, India.
Orcid Id: 0000-0003-2834-6479

Dr. Tarni Mandal
Associate Professor, Department of Mathematics,
National Institute of Technology, Jamshedpur, Jharkhand, India.
Orcid Id: 0000-0002-2563-4494

Abstract

The energy consumption of a data center is the critical research issue, i.e. Virtual Machine (VM) placements to satisfy the resource requirements with minimum energy consumptions and active servers. The Multi-Objective Virtual Machine Placement (MOVMP) is a representation of a kind of combinatorial optimization problem. In this paper, Teaching Learning Based Optimization (TLBO) is used to solve the MOVMP problem. Our approach accounts for the multi-objective resource management and the simulation based result validate the effectiveness of TLBO compared to First Fit (FF), Best Fit (BF) and Genetic Algorithm (GA).

Keywords: Multi-Objective, Virtual Machine, Teaching Learning Based Optimization, Metaheuristics, Genetic Algorithm, Energy Aware Computing.

INTRODUCTION

Cloud computing is based on the concept of dynamic provisioning, which is applied to services, computing capability, storage, networking, and infrastructure to meet user requirements. The resources are made available for the users through the Internet and offered on a pay-as-use basis from different Cloud computing vendors.

Technology, that makes cloud computing feasible are virtualization, cyber-infrastructure, and service orient infrastructure [1]. It offers scalable and elastic computing and storage services. The resources used for these services can be metered and the users can be charged only for the resources they used. From a provider's perspective, maximize profits by minimizing the operational costs. The adoption and deployment of cloud computing platforms have many attractive benefits, such as reliability, quality of service and robustness [2]. Cloud users can access computing resources without having to own, manage, and maintain them. There are three common cloud computing models known as Infrastructure as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS).

Virtualization

In the core technology, cloud computing uses the virtualization [3-6], which separates resources and services from the underlying physical delivery environment. The resources of a single physical machine (PM) are sliced into multiple isolated execution environments for multiple virtual machines (VMs). Virtual Machine Placement (VMP) is an important topic in cloud environment virtualization, in particular in IaaS model. VMP maps a set of virtual machines to a set of physical machines. For the cloud providers, a good VMP solution should maximize resource utilization and minimize power consumption [7].

The virtual technology enables on-demand or utility computing a just-in-time resource provisioning model in which computing resources, such as CPU, memory, and disk space are made available to applications only as needed [8]. Through virtualization, a cloud provider can ensure the quality of service (QoS) delivered to the users, while achieving a high server utilization and energy efficiency. VM placement is an important approach for improving power efficiency and resource utilization in cloud infrastructures. Several research works [9, 10] addressed the importance of placing VMs appropriately. Vogels [11] quoted the benefit of packing VMs efficiently in server consolidation. The proxy placement [12, 13] and object placement/replacement [14, 15] for transparent data replication bear some resemblance to the issues we face. Since, they all attempt to exploit the flexibility available in determining proper placement. In this paper, physical machines (PMs) or hosts or servers have the same meaning.

A Process for VM Placement

- For each server, compute application resource requirement using server's resource usage statistics over a period of time (e.g., several days/weeks/months).
- Choose a target server with compatible virtualization software, comparable CPU types, similar network connectivity, and usage of shared storage.
- Place the first VM on the first PM.

- Place the second VM to be place on the same PM if it can satisfy the resource requirements. If not, add a new PM and place the VM on this new machine.
- Continue until each of the VMs has been placed on a PM, adding a new PM when required.

CPU Power Consumption

The measurement results shown in Fig. 1 by specpower.com, even with nearly zero percentage of CPU utilization, a server can cost up to 50%-60% of the maximum power consumption [16-19]. Thus, it is better to push up the CPU utilization rate to achieve better energy efficiency.

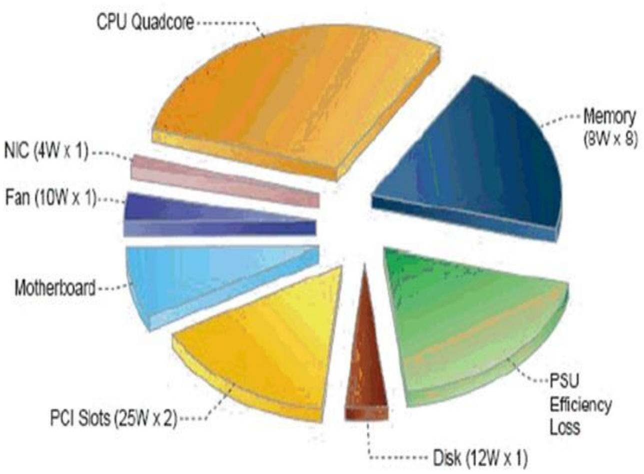

Figure 1: Benchmark of power consumption at various CPU utilization [18]

Evolutionary Algorithms

Currently, variety of traditional mathematical programming methods [20-22] are available to solve MOPs. However, some researchers [23-26], have identified several limitations of traditional mathematical programming approaches to solve MOPs. Some of them are the following:

- We need to run many times those algorithms to find several elements of the Pareto optimal set.
- Many of them require domain knowledge about the problem to be solved.
- Some of those algorithms are sensitive to the shape or continuity of the Pareto front.

These complexities call for alternative approaches to deal with certain types of MOPs. Among these alternative approaches, we can find Evolutionary Algorithms (EAs), which are stochastic search and optimization methods that simulate the natural evolution process. At the end of 1960s, Rosenberg [27] proposed the use of genetic algorithms to solve MOPs. However, it was until 1984, when David Schaffer [28] introduced the first actual implementation of what it is now called a Multi-Objective Evolutionary Algorithm (MOEA). From that moment on, many researchers [4], [29-33] have proposed a wide variety of MOEAs. As other stochastic search strategies (e.g., simulated annealing, ant colony optimization, or particle swarm optimization).

Genetic Algorithm (GA): is a probabilistic optimization algorithm that imitates the progression of natural evolution. The biological evolution process in chromosomes became the idea of GA. It is based on the idea of the fittest survival where new better solutions are obtained by recombination of with each other. Algorithm 1 shows the pseudo-code for GA algorithm for the optimal solution in cloud computing.

Algorithm 1: *Pseudocode of a Genetic Algorithm*	
1	$t \leftarrow 0$
2	Generate an initial population *P(t)*
3	**while** *the stoping criterion is not fulfilled* **do**
4	Evaluate the objective vector ***f*** for each individual in *P(t)*.
5	Assign a fitness for each individual in *P(t).*
6	Select from *P(t)* a group of parents *P'(t)* preferring the fitter ones.
7	Recombine individuals in *P'(t)* to create a child population *P"(t)*
8	Mutate individuals in *P"(t)*
9	Combine *P(t)* and *P"(t)* and select the best individuals to get *P(t+1).*
10	$t \leftarrow t + 1$.
11	**end.**

The fitness assignment scheme requires a ranking of the individuals according to a preference relation and then, assigning a scalar fitness value to each individual using such rank. The selection for reproduction (line 6) is carried out as in the single objective case, for instance, using tournament selection. In contrast, the selection for survival (line 9), intended to maintain the best solutions so far (i.e., elitism). It helps the VM allocator to choose the right machine for the allocation of resources.

SYSTEM MODEL

In a cloud environment, we have a pool of server nodes with applications running on them. The Data Center is fully virtualized and all the applications are running on VMs. The problem of VM placement across a pool of server nodes is related to the multidimensional vector packing problems. Dimensions in the packing problem are resource utilizations.

International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 21 (2017) pp. 10809-10815

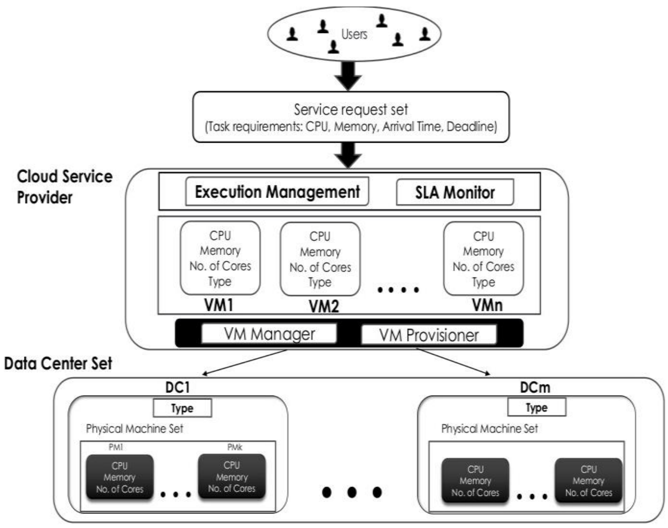

Figure 2: Cloud Computing Architecture

In our work, we used two dimensions to characterize a VM and a server node CPU and memory. We do not consider the disk size dimension because we assume that network-attached storage (NAS) is used as main storage across the cluster. If two VMs are running on the same server, the CPU utilization of the server is estimated as the sum of the CPU utilizations of the two VMs. To prevent CPU and memory usage of a server from reaching 100%, we have to impose an upper bound on resource utilization of a single server with some threshold value. The main idea behind this is that 100% utilization can cause severe performance degradation and VM live migration technology consumes some amount of CPU processing capability on the migrating node. Fig. 2 shows the Cloud Infrastructure of the system. These are some assumptions of virtualization system:

- Virtualization technologies allow the creation of multiple virtual machines on any of the available host.
- Hosts consumes energy in an idle state to perform maintenance functions and denoted as $P_{min.}$
- Hosts consumes more energy as per utilization of the CPU by the VMs.
- Hosts consumes maximum energy at the pick level and denoted as P_{max}.

Energy Consumption Modeling

The power consumption of servers can be accurately described by a linear relationship between the power consumption and CPU utilization [34, 35]. In order to save energy, servers are turned off when they are idle. Hence, their idle power is not part of the total energy consumption. Finally, we defined the power consumption of the ith server as a function of the CPU utilization as shown in Eq. (1).

$$E_i(\tau) = \begin{cases} (P_{max} - P_{min}) * \dfrac{U_i(\tau)}{100} + P_{min}, & U_i(\tau) > 0 \\ 0 & otherwise \end{cases} \quad (1)$$

Where the $U_i(\tau)$ represent the CPU utilization of i^{th} host server R_i at time τ. P_{max} and P_{min} are the power consumption at highest level and at idle respectively.

Resource Wastage Modeling

The remaining resources available on each server may vary greatly with different VM placement solutions. To fully utilize multidimensional resources, the following equation is used to calculate the potential cost of wasted resources:

$$W_i = \frac{\left|L_i(p) - L_i(m)\right| + \varepsilon}{U_i(p) + U_i(m) + \varepsilon} \quad (2)$$

Where W_i denotes the resource wastage of the i^{th} server, $U_i(p)$ and $U_i(m)$ represent the normalized CPU and memory resource usage (i.e., the ratio of used resource to total resource). $L_i(p)$ and $L_i(m)$ represent the normalized remaining CPU and memory resource. ε is a very small positive real number (= 0.0001 to avoid the divide by zero).

Optimization formulation

Suppose ***m*** number of VMs that are to be placed on ***n*** servers. Let R_{pj} be CPU demand of each VM, T_{pi} be the threshold of CPU utilization associated with each server, R_{mj} be the memory demand of each VM, and T_{mi} be the threshold of memory utilization associated with each server. We use two binary variables x_{ij} and y_i . The binary variable x_{ij} indicates if VM_j is assigned to server i and the binary variable y_i indicates whether server i is in use or not. The placement problem can be formulated as [36] as eq. (3)-(8) :

$$Min \sum_{i=0}^{n} E_i = \sum_{n=1}^{\infty}\left[y_i * \left(\left(p_i^{max} - p_i^{min} \right) * \sum_{i=1}^{n}\left(x_{ij} * R_{p_j} \right) + p_i^{min} \right) \right] \quad (3)$$

$$Min \sum_{i=1}^{n} W_i = \sum_{i=0}^{n}\left[y_i * \frac{\left| \left(T_{p_i} - \sum_{j=1}^{m}\left(x_{ij} * R_{p_j} \right) \right) - \left(T_{ni} - \sum_{j=1}^{m}\left(x_{ij} * R_{ni} \right) \right) \right| + \varepsilon}{\sum_{j=1}^{m}\left(x_{ij} * R_{p_j} \right) + \sum_{j=1}^{m}\left(x_{ij} * R_{ni} \right)} \right] \quad (4)$$

Subjected to:

$$\sum_{i=1}^{n} x_{ij} = 1 \quad \forall\ j \in J \quad (5)$$

$$\sum_{j=1}^{m} R_{p_j} * x_{ij} \le T_{p_i} * y_i \quad \forall\ i \in I \quad (6)$$

$$\sum_{j=1}^{m} R_{m_j} * x_{ij} \le T_{m_i} * y_i \quad \forall\ i \in I \quad (7)$$

$$y_i,\ x_{ij} \in \{0, 1\} \quad (8)$$

Constraint Eq. (5) assigns a VM_j to only one of the servers. Constraints Eq. (6) and Eq. (7) model the capacity constraint of the server. Constraint Eq. (8) defines the domain of the variables of the problem. Given a set of ***m*** VMs and a set of ***n*** PMs, there are a total of ***m*n*** possible VMP solutions.

TEACHING LEARNING BASED OPTIMIZATION (TLBO)

TLBO is a one of the best among meta-heuristic techniques with highly competitive performances [37]. The prominent feature of this computational technique is that it is free from algorithm specific parameters. it uses the mean value of the number of students of the classroom to update the solution and have greediness to accept a good solution [38]. TLBO initiates with a group of tentative solutions, called learners, dispersed randomly in the problem search space. The learners update their knowledge due to course of time, i.e., fitness, through two stage learning process: “Teacher phase” and the “Learner phase”. The ever best learner obtained through the iterative process is considered as the final solution. The learning phases of TLBO can be summarized as below [39,40].

Teacher phase

In this phase, learners learn through the teacher who tries to improve the existing mean result $Mean_{d,t}$ of the class at iteration t for each dimension d towards him or her so let the new mean is $Teacher_{d,t}$. The difference of the mean results is then evaluated as below:

$$\nabla Mean_{d,t} = random(Teacher_{d,t} - TF_{d,t} * Mean_{d,t}) \tag{9}$$

where $TF_{d,t}$ is the teaching factor given by eq. 10 and each dimension is updated using eq. 11.

$$TF_{d,t} = \frac{Mean_{d,t}}{Teacher_{d,t}} \tag{10}$$

$$Xnew_{d,t} = Xold_{d,t} + \nabla Mean_{d,t} \tag{11}$$

Accept $Xnew_{d,t}$ if it gives better fitness value.

Learner phase

In this phase each learner improves his or her knowledge by interacting randomly with other learners. A learner interacts randomly with other learners for enhancing his or her knowledge. A learner learns new things if the other learner has more knowledge than him or her. For the iteration t, two learners are randomly chosen. If they have better knowledge than the learner, the difference between them is added in the learner knowledge according to Eq. (12), and Eq. (13).

$$Xnew_{d,t} = Xold_{d,t} + rand(X_y - X_z);\ f(X_z) < f(X_y) \tag{12}$$

$$Xnew_{d,t} = Xold_{d,t} + rand(X_z - X_y);\ f(X_y) < f(X_z) \tag{13}$$

where $y \neq z$. Accept $Xnew_{d,t}$ if it gives better function value.

Algorithm 2 decribes the proposed TLBO for VM Placement algorithm.

Algorithm 2: *Pseudocode of **MOVMP** using **improved TLBO***

```
t ← 0
Objective functions f1 and f2 calculated as Eq (3) and (4).
Generate an initial students of the classroom using greedy search algorithms, some random search and constraints handling algorithms. (Place VM as per Constraints Eq. (5)-(9) )
Calculate the Objective vector functions f1 and f2 for whole students of the classroom as Eq (3) and Eq (4).
while the stoping criterion is not fulfilled do
  // Teacher phase //
  Calculate the mean of each design variable x_mean
  Identify the best solution(teacher)
  for i = 1 to N do
    Calculate the teaching factor
    T_(i, F) = round[1 + rand(0, 1) * (2-1) ]
    Modify solution based on best solution(teacher)
    xnew^i = x^i + ∇Mean_d,i
    Calculate objective functions for new mapped students.
    if xnew^i is better than x^i then
      x^i = xnew^i
    End
    // End of Teacher phase //
    // Student phase //
    Randomly select another learner (x^j), such that j ≠ i
    if x^i is better solution than x^j than
      xnew^i = x^i + rand(0,1)*(x^i - x^j)
    else
      xnew^i = x^i + rand(0,1)*( x^j - x^i)
    End
    if xnew^i is better than x^i then
      x^i = xnew^i
    End
    // End of Student phase //
  End
  t ← t + 1.
End
```

SIMULATION RESULTS

In this paper, The performance of proposed TLBO is evaluated in the term of the number of active servers, the energy consumption , the servers utilization. The methods are investigated on the CloudSim simulator, number of VMP requests are 100 to 1000.

Like GA, TLBO is also a population-based algorithm, which implements a group of solutions to proceed to the optimum solution. GA requires the crossover probability, mutation rate, and selection method to improvisations. Unlike other optimization techniques TLBO does not require any algorithm parameters to be tuned, thus making the implementation of TLBO simpler. TLBO uses the best solution of the iteration to change the existing solution in the population, thereby increasing the convergence rate. As in GA, which uses selection, crossover and mutation. TLBO uses two different phases, the 'teacher phase' and the 'learner phase'. TLBO uses the mean value of the population to update the solution.

In Active servers experiment, the number of VMP requests are 100, 200, 300, ..., 1000. The result shown in Fig. 3 gives the comparison results of active servers with First Fit (FF), Best Fit (BF) and Genetic Algorithm(GA). As shown in Fig. 3 with the different size of VMP requests, the TLBO approach always gives the minimum number of active servers.

Figure 3: Comparison on Number of Active Servers

In Fig. 4 shows the comparison results of servers/PMs utilization of the all active servers/PMs. As shown in Fig. 4 with the different size of VMP requests, the TLBO approach always gives the better utilization compare to the other approaches.

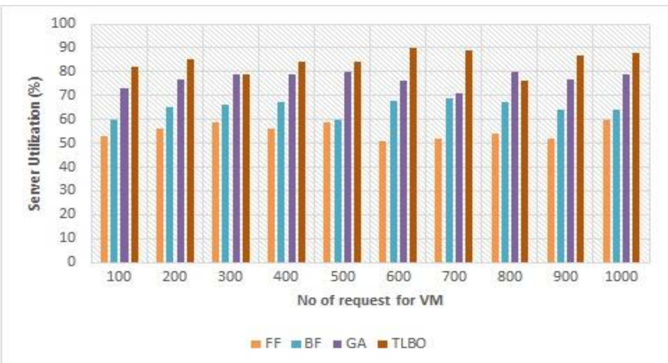

Figure 4: Server Utilization

In Fig. 5 shows the total energy consumptions of all active servers. As shown in Fig. 5 with the different size of VM requests, the TLBO approach always gives the lowest energy consumption with compare to the other approaches.

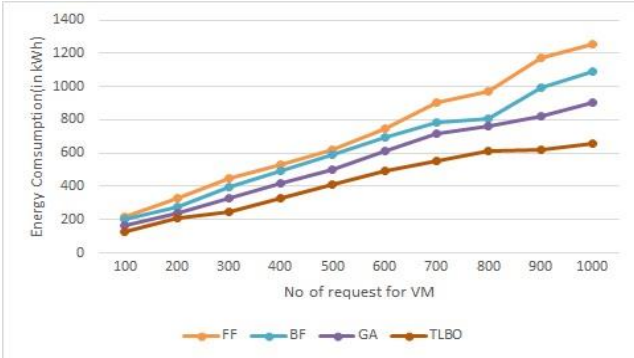

Figure 5: Total Energy Consumption in cloud infrastructure

CONCLUSION

In this paper, we present the optimized virtual machine placement optimization approach for a cloud data center to reduce the total energy consumptions with better VMP algorithm using TLBO. Our approach accounts for the multi-objective resource management. The simulation based result validate the effectiveness of our approach compared to the other approaches. Future work focused on providing the VMP with the migrations.

REFERENCES

[1] M. A. Vouk, Virtualization of Information Technology Resources. In Electronic Commerce: A Managerial Perspective 2008. 5th Edition, Prentice-Hall Business Publishing, 2008.

[2] M. Randles, D. Lamb, E. Odat, and A. Taleb-Bendiab, "Distributed redundancy and robustness in complex systems," Journal of Computer and System Sciences, vol. 77, no. 2, pp. 293–304, 2011.

[3] K. Li, H. Zheng, J. Wu, and X. Du, "Virtual machine placement in cloud systems through migration process," International Journal of Parallel, Emergent and Distributed Systems, vol. 30, no. 5, pp. 393–410, 2015.

[4] C. A. C. Coello and G. T. Pulido, "Multiobjective optimization using a micro-genetic algorithm," in Proceedings of the genetic and evolutionary computation conference gecco, 2001, pp. 274–282.

[5] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," Future Generation Computer Systems, vol. 28, no. 5, pp. 755–768, 2012.

[6] T. D. Braun, H. J. Siegel, N. Beck, L. L. B¨ol¨oni, M. Maheswaran, A. I. Reuther, J. P. Robertson, M. D. Theys, B. Yao, D. Hensgen et al., "A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems," Journal of Parallel and Distributed computing, vol. 61, no. 6, pp. 810–837, 2001.

[7] H. Zhao, Q. Zheng, W. Zhang, Y. Chen, and Y. Huang, "Virtual machine placement based on the vm performance models in cloud," in Computing and Communications Conference (IPCCC), 2015 IEEE 34th International Performance. IEEE, 2015, pp. 1–8.

[8] D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy, and G. Jiang, "Power and performance management of virtualized computing environments via lookahead control," Cluster computing, vol. 12, no. 1, pp. 1–15, 2009.

[9] M. Cardosa, M. R. Korupolu, and A. Singh, "Shares and utilities based power consolidation in virtualized server environments," in Integrated Network Management, 2009. IM'09. IFIP/IEEE International Symposium on. IEEE, 2009, pp. 327–334.

[10] L. Grit, D. Irwin, A. Yumerefendi, and J. Chase, "Virtual machine hosting for networked clusters: Building the foundations for autonomic orchestration," in Proceedings of the 2nd International Workshop on Virtualization Technology in Distributed Computing. IEEE Computer Society, 2006, p. 7.

[11] W. Vogels, "Beyond server consolidation," Queue, vol. 6, no. 1, pp. 20–26, 2008.

[12] K. Li and H. Shen, "Proxy placement problem for coordinated enroute transcoding proxy caching," Computer Systems Science and Engineering, vol. 19, no. 6, pp. 327–335, 2004.

[13] L. Keqiu and S. Hong, "Optimal proxy placement for coordinated enroute transcoding proxy caching," IEICE transactions on information and systems, vol. 87, no. 12, pp. 2689–2696, 2004.

[14] K. Li, H. Shen, F. Y. Chin, and S. Q. Zheng, "Optimal methods for coordinated enroute web caching for tree networks," ACM Transactions on Internet Technology (TOIT), vol. 5, no. 3, pp. 480–507, 2005.

[15] K. Li, H. Shen, F. Y. Chin, and W. Zhang, "Multimedia object placement for transparent data replication," Parallel and Distributed Systems, IEEE Transactions on, vol. 18, no. 2, pp. 212–224, 2007.

[16] VmWare. (2012) Vmware. Online: www.vmware.com

[17] SPECpower. (2013) Benchmark results summary of hitachi. Online: www.spec.org/power_ssj2008/ results/ res2013q1/powerssj2008-20130305-00600.html

[18] H. Packard. (2011) Power regulator for proliant servers. Online: www.h10032.www1.hp.com/ctg/Manual/ c00300430.pdf

[19] L. Gray, A. Kumar, and H. Li, "Characterization of specpower_ssj2008 benchmark," in SPEC Benchmark Workshop, 2008.

[20] K. Miettinen, Nonlinear multiobjective optimization. Springer Science & Business Media, 2012, vol. 12.

[21] M. Kaisa, "Nonlinear multiobjective optimization," Boston, Massachusetts: Kluwer Academic Publishers, 1999.

[22] S. Greco, J. Figueira, and M. Ehrgott, "Multiple criteria decision analysis," Springer's International series, 2005.

[23] C. A. C. Coello, "A comprehensive survey of evolutionary-based multiobjective optimization techniques," Knowledge and Information systems, vol. 1, no. 3, pp. 269–308, 1999.

[24] K. Deb, Multi-objective optimization using evolutionary algorithms. John Wiley & Sons, 2001, vol. 16.

[25] D. B. Fogel, Artificial intelligence through simulated evolution. Wiley-IEEE Press, 2009.

[26] D. Fogel and Z. Michalewicz, "How to solve it: Modern heuristics," Ed Springer, 2000.

[27] R. S. Rosenberg, "Stimulation of genetic populations with biochemical properties: I. the model," Mathematical Biosciences, vol. 7, no. 3, pp. 223–257, 1970.

[28] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in Proceedings of the 1st international Conference on Genetic Algorithms. L. Erlbaum Associates Inc., 1985, pp. 93–100.

[29] N. Srinivas and K. Deb, "Muiltiobjective optimization using nondominated sorting in genetic algorithms," Evolutionary computation, vol. 2, no. 3, pp. 221–248, 1994.

[30] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," IEEE transactions on Evolutionary Computation, vol. 3, no. 4, pp. 257–271, 1999.

[31] J. rey Horn, N. Nafpliotis, and D. E. Goldberg, "Multiobjective optimization using the niched pareto genetic algorithm", in IEEEIC pp. 61, 801–2296, 1993.

[32] C. M. Fonseca, P. J. Fleming et al., "Genetic algorithms for multiobjective optimization: Formulationdiscussion and generalization." in ICGA, vol. 93. Citeseer, 1993, pp. 416–423.

[33] J. Knowles and D. Corne, "The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation," in Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on, vol. 1. IEEE, 1999.

[34] L. Minas and B. Ellison, "The problem of power consumption in servers," Intel Corporation. Dr. Dobb's, 2009.

[35] Y. C. Lee and A. Y. Zomaya, "Energy efficient utilization of resources in cloud computing systems," The Journal of Supercomputing, vol. 60, no. 2, pp. 268–280, 2012.

[36] A. J. Younge, G. Von Laszewski, L. Wang, S. Lopez-Alarcon, and W. Carithers, "Efficient resource management for cloud computing environments," in Green Computing Conference, 2010 International. IEEE, 2010, pp. 357–364.

[37] H. Bouchekara, M. Abido, and M. Boucherma, "Optimal power flow using teaching-learning-based optimization technique," Electric Power Systems Research, vol. 114, pp. 49–59, 2014.

[38] R. V. Rao, V. J. Savsani, and D. Vakharia, "Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems," Computer-Aided Design, vol. 43, no. 3, pp. 303– 315, 2011.

[39] R. V. Rao and V. Patel, "Multi-objective optimization of two stage thermoelectric cooler using a modified teaching–learning-based optimization algorithm," Engineering Applications of Artificial Intelligence, vol. 26, no. 1, pp. 430–445, 2013.

[40] N. Kanwar, N. Gupta, K. Niazi, and A. Swarnkar, "Simultaneous allocation of distributed resources using improved teaching learning based optimization," Energy Conversion and Management, vol. 103, pp. 387–400, 2015.