Multi-Objective Biogeography-Based Method to Optimize Virtual Machine Consolidation

Kai Shi[1,2], HuiQun Yu(Corresponding Author)[1], Fei Luo[1], GuiSheng Fan[1]

[1] Department of Computer Science and Engineering
East China University of Science and Technology, Shanghai 200237, China
[2]Shanghai Key Laboratory of Computer Software Evaluating and Testing, Shanghai 201112, China
BH4AWS@163.com, yhq@ecust.edu.cn, luof@ecust.edu.cn, gsfan@ecust.edu.cn

***Abstract*—Virtual machine consolidation (VMC) is an important issue in cloud computing, which can be used to reduce power consumption and achieve reasonable resource allocation. In this paper, an IMBBO algorithm is proposed to solve the multi-objective optimization problem of VMC through improving the classical Biogeography-Based Optimization (BBO). An improved Cosine migration model and an improved mutation model are presented to increase the efficiency of achieving the optimal solution. Meanwhile, three optimization objectives for server power consumption, load balancing and migration resource overhead are mainly addressed. Finally, several experiments are done to evaluate the performance of IMBBO by comparing with Gravitational Search Algorithm (GSA) based on the synthetic and real VM running data. The results show that the IMBBO optimizes VM consolidation with higher efficiency.**

***Keywords*—multi-objective optimization; biogeography-based optimization; virtual machine consolidation; cloud computing**

I. Introduction

CLOUD computing has been popular in the IT industry since 2008. The virtualization is the one of core technology in cloud computing. Generally, data centers use the deployed servers to provide different services and it will underutilize the server and increase the energy costs. A great number of virtual machines (VMs) which are randomly deployed in data centers in order to provide the service for end-users according to the required resources. However, this random deployment may bring about serious power consumption and resource contention. Meanwhile, the load unbalancing is also urgent to be solved. For example, data centers consumed about 1.5% of the total generated electricity in the U.S. in 2006 according to a report published in 2008. In fact, a great number of physical machines (PMs) are in idle state.

Recently, many researchers and institutes have focused on virtual machine consolidation problem, which can effectively reduce the energy consumption in one data center by consolidating VMs to PMs and shut down the idle ones. However, most methods just consider how to consolidate some VMs to some PMs, it will result in the unreasonable resource contention because VMC is a dynamic process. For the cloud providers, a good VMC scheme should maximize resource utilization and minimize power consumption and others.

Since the traditional VMC scheme just focus on how to minimize the consolidation, the similarity among VMs resource utilization isn’t considered. More VMs may be migrated to the same PM that will extremely result in the unstable state of PMs and other VMs. In additional, VMs migration not only occupies the network bandwidth, but also the CPU utilization in source and destination. PMs may cause extra overhead such as more CPU cycle, the high memory operation and network transfer. Therefore, these factors should be considered in VMC in order to save power consumption.

In order to solve the above objective optimization problem, a novel multi-objective VMC algorithm named IMBBO (Improved BBO) is proposed based on improving the classical BBO algorithm. **The major contributions of this paper are as follows.**

1. The VMC problem is formulated as a multi-objective optimization problem, which includes the novel minimizing server power consumption, achieving better loading balance based on VMs’ resource correlation, and reducing migration resource overhead considering in source and destination PMs.

2. A novel multi-objective optimization algorithm, IMBBO, is presented. The key factors of migration and mutation model in IMBBO are modified in order to preferably adapt to the VMC problem. Meanwhile, based on synthetic and real VMs running data, we compare the proposed IMBBO with Gravitational Search Algorithm (GSA), thus explaining the advantage of our method.

The rest of this paper is organized as follows. Section Ⅱ discusses related works. Section Ⅲ formulates the VMC as a multi-objective optimization problem. Section Ⅳ introduces the proposed algorithm, including its main process and key strategies. The simulation results are presented in SectionⅤ. And SectionⅥ is the conclusion.

II. Related Work

Recently, VMCP is one the well research area in cloud computing. Since, this problem is the NP-hard problem, many researchers from all world make their vastly effort to explore the approximate optimal solution. Generally, the VMCP can be

DOI reference number: 10.18293/SEKE2016-151

classified into two categories in terms of consolidation type, including static and dynamic [1].

The first type of research focuses on static single-objective virtual machine consolidation in tradition. Some research was based on off-line historical data and analyses the running feature in order to make a correct decision. E.g. the famous simulation software cloudSim[2], improving energy efficiency [3], etc. These algorithms usually use threshold values to judge the running status of VMs or PMs. Furthermore, these researches just focus on single-objective consolidation optimization. However, in really, there is the correlation among all kinds of resources. Many extra conditions also should be considered in process. E.g. Fei Xu et al. and Raja Wasim Ahmad et al. wrote a survey about managing performance overhead of VMs and VMs migration in cloud [1][4], which definitely summaries these features and correlation.

However, these static methods can't perfectly satisfy the rapid growth in the current data center environment. So the dynamic multi-objective VMC becomes a great attention [1][4]. Since many VMCP just solve a single objective, such as resource utilization, power consumption, etc. But, real VMC solutions often need to consider multi objectives and make a real time decision. E.g. Nguyen Trung Hieu et al. presented a VMCUP algorithm for improving the energy efficiency which dynamically predicts CPU utilization [6]. Chaima Ghribi et al. used B-matching algorithms to minimize the numbers of servers [5]. Chaima Ghribi et al. proposed the allocation and migration algorithms to minimize overall energy consumption [7]. Changming Zhao et al proposed a novel algorithm named Segmentation Iteration Correlation Combination (SICC), which integrates the methods of statistic regression modeling in order to reduce the difference of peak-mean value of VM resource utilization [8].

In recent years, other population research usually adopts heuristic algorithms to solve VMCP, such as heuristic bin packing, Biology-based optimization [9][10], simulated annealing optimization, etc. In bin packing, the problem was formulated as a variant of vector bin packing problem, E.g. First Fit, Best Fit, Next Fit and Best Fit decreasing, etc. Furthermore, The GABA [11], the ant colony methods [12], SAPSO [13] and Simulated Annealing algorithm [14] are also represented. The ant methods used to pack VMs to the least number of PM necessary for the current workload. The SAPSO[13] focuses on self-adaptive particle swarm. The GABA[11] maps the VMs according to estimated future workload. Antonio Marotta et al. focus on a simulated annealing base algorithm which solves VMC by evaluating the attractiveness of the possible VM migrations, etc.

Since this paper focuses on the classical BBO algorithm, Biogeography-based optimization (BBO) is a new evolutionary algorithm firstly proposed in 2008 [15] and is an extension of biogeography theory to evolutionary algorithm [16], which is based on the mathematical model of biological species distribution and migration[17]. The BBO has demonstrated good performance on various unconstrained and constrained benchmark functions [18][19]. Further, it has been applied to real world optimization problems, including sensor selection [15], power system optimization [20] , etc.

Recently, some extension researches also have been presented. Typically, Haiping Ma [21] improved the classical BBO algorithm and analyzes the equilibrium of migration models. Haiping Ma and Dan Simon [22] discussed migration models using markov theory and blended BBO for constrained optimization. In real, BBO maps these factors as suitability index variable (SIV) and habitat suitability index (HSI) to mathematical solution space in order to find the optimal solution of a certain problem.

III. Problem Formulation

The aim goal of this paper is to study the VMCP through multi-objective optimization which is based on IMBBO. To formulate this problem we will deal with it in the mathematical forms:

N : The number of virtual machines in data centers (VMs)

M : The number of physical machines in data centers (PMs)

v_i : $i \in \{1,2,3,\cdots,N\}$, The virtual machine of label i

p_j : $j \in \{1,2,3,\cdots,M\}$, The physical machine of label j

ass_{ij} : The binary value represents whether virtual machine v_i is assigned to physical machine p_j

V : The set of virtual machines, namely $v_1, v_2, v_3, \cdots, v_N$

P : The set of PMs, namely $\mathrm{p}_1, p_2, p_3, \cdots, p_M$

e_{j}^{busy} : The energy consumption of p_j , when u_j =100%

e_j^{idle} : The energy consumption of p_j , when u_j =0% (just running the OS system)

v_{cpu}^i : The CPU demand of v_i

v_{mem}^i : The memory demand of v_i

v_{net}^i : The network demand of v_i (bandwidth)

p_{cpu}^j : The CPU capacity of p_j

p_{mem}^j : The memory capacity of p_j

p_{net}^j : The network capacity of p_j (bandwidth)

S_{kl}^{cpu} : The coefficient of similarity by CPU utilization between v_k and v_l

S_{kl}^{mem} : The coefficient of similarity by memory utilization between v_k and v_l

S_{kl}^{net} : The coefficient of similarity by network utilization between v_k and v_l

$\mathrm{cpu}(pm_j)$: The current CPU utilization of pm_j

$\mathrm{mem}(\mathrm{v}m_i)$: The memory size of vm_i

$\mathrm{net}(pm_j)$: The current NET bandwidth of pm_j

∂_{ex} : The extra CPU utilization coefficient.

T_{stop} : The stop time of virtual machine migration.

VP is a matrix which presents the allocation of VMs to PMs. Each element contains two types of value. If ass_{ij} =1, v_i is assigned to p_j . Otherwise, v_i isn't assigned to p_j , where $1 \le i \le N, 1 \le j \le M$.

$$Min\sum_{j=1}^{M} PowerComsumption_j = \sum_{j=1}^{M}\left[y_j \times \left(e_j^{idle} + \left(e_j^{busy} - e_j^{idle} \right) \times \frac{\sum_{i=1}^{N} ass_{ij} \bullet \left(v_{cpu}^i + \delta_1 \bullet v_{mem}^i + \delta_2 \bullet v_{net}^i \right)}{p_{cpu}^j} \right) \right] \tag{1}$$

$$MinLoadingBalance = \sum_{j=1}^{M}\left[y_j \times \left(\sigma_{cpu} \bullet \frac{\sum_k \sum_{l \neq k} S_{kl}^{cpu}}{C_{vmcount \in P_j}^2} + \sigma_{\mathrm{mem}} \bullet \frac{\sum_k \sum_{l \neq k} S_{kl}^{mem}}{C_{vmcount \in P_j}^2} + \sigma_{net} \bullet \frac{\sum_k \sum_{l \neq k} S_{kl}^{net}}{C_{vmcount \in P_j}^2} \right) \right] \tag{2}$$

$$\min MigrationOverhead = \sum_{j=1}^{M}\left\{ \mathrm{y}_j \times \left(\sum_{i=1}^{N} \left(\mathrm{cpu}(pm_j) \bullet \partial_{ex} + T_{stop} + \frac{mem(vm_i)}{net(pm_j)} + \mathrm{cpu}(pm_l) \bullet \partial_{ex} + \frac{mem(vm_i)}{net(pm_l)} \right) \right) \right\} \tag{3}$$

$$\mathrm{s.}t.\begin{cases} \forall i, \sum_{i=1}^{N} v_{cpu}^i \bullet ass_{ij} \le \sum_{i=1}^{N} p_{cpu}^j \bullet y_j ; \sum_{i=1}^{N} v_{mem}^i \bullet ass_{ij} \le \sum_{i=1}^{N} p_{mem}^j \bullet y_j ; \sum_{i=1}^{N} v_{net}^i \bullet ass_{ij} \le \sum_{i=1}^{N} p_{net}^j \bullet y_j & (4) \\ \sigma_{cpu} + \sigma_{\mathrm{mem}} + \sigma_{net} = 1; ass_{ij} \in \{1,0\}; i \in \{1,2,3,\cdots,N\}, j \in \{1,2,3,\cdots,M\} & (5) \end{cases}$$

As above, **Eq.(1)-Eq.(3)** illustrate the objectives of VMCP and **Eq.(4)-Eq.(5)** are the total constraint condition for these objectives. These demonstrate that all required resources aren't greater than the capacity of one PM.

The first objective **Eq.(1)** is the minimization of server power consumption. The popular power consumption model has been introduced in [9][10] which shows the power consumption is linearly proportional on CPU utilization. However, the classic equation just pays attention to the CPU utilization. The operation of memory and network relating with the CPU cycle doesn't be considered. So more detail information of the new power consumption function is defined which is divided into static and dynamic parts in this paper. When the physical machine doesn't run any tasks (just OS system), the parameter e_{idle}^j is defined in idle status. In addition, the configuration parameters δ_1 and δ_2 present the weight coefficient that the memory and network operation take the proportion of CPU cycle. Finally, this scenario illustrates that PMs can be turn off, when these consume no extra energy.

The second objective **Eq. (2)** is the load balancing based on minimization correlation of resource utilization among PMs in the data center. The phenomenon shows that the load balancing is the similarity average resource utilization among PMs. Since the resources competition will result unstable resource condition among PMs and VMs, the reasonable solution is that these VMs with minimum similarity of resource utilization should be consolidated into same PM. In addition, the configuration parameters, σ_{cpu} , σ_{mem} and σ_{net} , are the weight coefficient, which is satisfied with constraint condition $\sigma_{cpu} + \sigma_{mem} + \sigma_{net} = 1$.Next, the $C_{vmcount \in P_j}^2$ is the permutation and combination in order to get average values of similarity.

The third objective **Eq. (3)** is the minimization of migration resource overhead in data center. When one VM migrates from pm_j to pm_k , the resource overhead may contain the CPU consumption and network bandwidth consumption which is both in source and destination PMs. Besides, the stop time of VMs and migration times, for example, $mem(vm_i) / net(pm_j)$ also should be considered in process. The parameter ∂_{ex} is the proportionality coefficient that the VM migration process may extra occupy the CPU cycle.

IV. Improved Biogeography-Based Optimization

In this section, the brief classical BBO algorithm will be introduced which includes the theory and the important features. Then, the improvement Cosine migration rate model and mutation rate model will be discussed in detail. Finally, the process of IMBBO will be described by the pseudo-codes.

A. Biogeography-Based Optimization

The classical Biogeography-based optimization algorithm is introduced by Simon in [15], which is based on the mathematics of biogeography theory and is a population global optimization approach. This algorithm can guarantee convergence to the optimal solution, if it is given enough generations (iterations). Biogeography studies the geographical distribution of species. Among them, the most important feature is migration and extinction (mutation).

B. The main improvement in IMBBO

1) The Improved Cosine Migration Rate Model

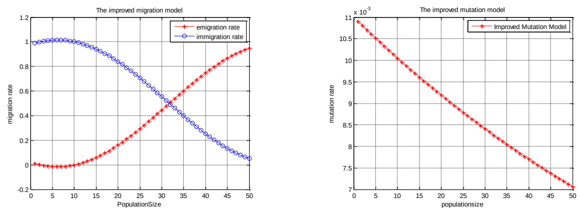
In this paper, the improved migration model is used to represent the migration feature. The linear function in BBO is basic migration model in BBO, which can be used to share SIVs between habitats. However, the linear model is the theoretical model with ideal condition. When there are more or less species in one habitat, the change rate of immigration and emigration will trend to the steady state. Otherwise, the change rate is fierce in the real virtualization environment. So this situation can be analogized in VMCP.

Meanwhile, many VMs are placed in the same PM may greatly increase the probability of resource contention. Finally, some VMs should be migrated to other PMs in order to maintain the running performance. From this, the improved Cosine migration rate model can be illustrated in **Eq. (6)**

$$\begin{aligned} \lambda_{\mathrm{k}} &= \frac{I}{2}\left(\mathrm{Cos}(\frac{k}{n} \bullet \pi + \beta) + 1 \right) + \varepsilon \\ \mu_{\mathrm{k}} &= \frac{E}{2}\left(-\mathrm{Cos}(\frac{k}{n} \bullet \pi + \beta) + 1 \right) + \omega \end{aligned} \tag{6}$$

Where λ_{k} and μ_{k} are presented by the immigration and emigration rates of the number of species k. The I is the maximal immigration rate when the number of species is zero. The E is the maximal emigration rate, when the number of species is to the maximum. So we set I = E in order to decrease experiment complexity in this paper. In addition, the parameter

β is the negative trigonometric offset angle (typically between -$\pi/2$ and 0). It denotes the degree of temporary positive immigration rate feedback in classical BBO. Next, the parameters ε and ω respectively denote the balance values. The migration rate cure for a single habitat is illustrated in **Fig.1(a)**. Further, the character of curve is more similar to the VM consideration features.



(a) **(b)**

Fig.1 Improved Migration and mutation Rate Curve Chart

2) The Improved Mutation Model

The mutation model is another important feature in the classical BBO, which realizes mutation through a uniformly random function to probabilistically replace SIVs by randomly generating new SIVs in a solution. This situation can much better avoid search prematurity. However, the random function doesn't adapt to the VM running environment in real.

The improved descending function in this section, which is compared to the random function in BBO, is a more excellent strategy. With the increasing numbers of iteration, more excellent elitisms will be contained in next generation. The system will more tend to be a stable state. That is, the probability of mutation should be descended in iterations. The **Eq. (7)** represents the improved mutation model.

$$\mathrm{mu}_i = \tau \cdot \exp(-\sigma \cdot i) + \eta \qquad \textbf{(7)}$$

Where the parameter τ is the slope value, it is defined by 0.01 in experiment. In addition, the parameter σ will be used to decide the degree of mutation rate in a time period. Next, the parameter η is the rectify coefficient, it is defined by 0.001 in experiment. The **Fig.1(b)** illustrates the improved mutation rate cure in a single habitat.

```
Pseudo-codes of IMBBO Algorithm
Input: the habitat matrix , migration, mutation model, objectives
Output: the optimization solution and fitness
1: Initialize the parameter and habitats(candidate solutions)
2: While the stop condition is not satisfied
3:       Calculate HIS (fitness) for each habitat
4:       Update migration model and Calculate the rate value
5:       for each habitat(solutions) do
6:       for i=1 to maximum number of SIVs do                Migration
7:           if rand < λ_i (improved) then                   Selection
8:               for j=1 to maximum number of SIVs do
9:                   if rand < μ_j (improved) then           Emigration
10:                      Sharing the SIV information from x_i to x_j
11:                  end if
12:              end for
13:          end if
14:      end for
15:      end for
16:      for i=1 to all habitats do                          Mutation
17:          for j =1 to maximum numbers of SIVs do
18:              if rand < mutation probability(improved) then
19:                  performing the mutation process
20:              end if
21:          end for
22:      end for
23:      process elitism and check the data legalization
24: end while
```

Fig.2 the pseudo-codes of IMBBO

C. The analysis of IMBBO

The main process of IMBBO is similar as BBO. The pseudo-codes are as **in Fig.2**. Further, in order to preferably satisfy the VMCP and performance, the improved migration model, including immigration and emigration rate calculation, is modified in line 7 and 9. In addition, the improved mutation model is also modified in line 18.

The time complexity of IMBBO is as:

$$iteration * [O(1) + 2 * O(m) + 2 * O(mn)] \approx K * O(mn)$$

The *m* presents the habitat size and the *n* presets a number of independent variables called Suitability Index Variables, which simulate the habitat number and features in habitat. So it is proportional to *m*n*. In addition, the space complexity is *m*n*, which presents the matrix size in IMBBO.

V. Evaluation

In this section, experiment environment will be described and the results will be evaluated by using two different instance types, including synthetic instance and real VM running instance. Then, many experiments comparing with GSA algorithm are carried out to verify the performance of our algorithm from different aspects. Finally, we analyze and discuss the experimental results in detail.

A. Experimental and Method

1) Experimental Environment

In the experiment, two types of experiment data are used to verify the performance of IMBBO algorithm which is compared with GSA. Our experiment environment is based on the real VM running environment for ECUST Virtual Cloud Laboratory System. Meanwhile, IMBBO algorithm will be deployed on Monitor Server. The Virtual Cloud Laboratory System was developed on OpenStack. This system will provide basic experiment courses for students. The cluster is composed of a numbers of Dell PowerEdge R730 which is viewed as the computing node. Each server consists 2 physical CPUs (IntelE5-2650 v3 2.3GHz, 25M) and 256GB main memory.

TABLE I

Parameter Values for IMBBO and GSA Algorithm

Parameter	Value	Parameter	Value
Habitats Size:	50	Improved mutate model τ :	0.01
SIV numbers:	200	Improved mutate model σ :	0.01
Elite habitats:	2	Improved mutate model η :	0.001
Maximum immigration rate:	1	Iterations numbers:	1000
Maximum emigration rate:	1	Distance bound norm:	2
The configuration coefficient β	+/-pi/8	G0(GSA):	100
The configuration coefficient ε ω	-0.02	Alfa(GSA):	20
Maximum constant mutate rate:	0.04		

2) Experimental Method

The different configuration coefficient will be used to verify this IMBBO algorithm. Some parameter values of IMBBO algorithm and GSA are defined in **Table I**. In order to getting average and exact results, each set is optimized by running independently 20 times and the average value is reported for each result. The Gravitational Search Algorithm (GSA) will view as the comparing algorithm for verifying the performance of IMBBO algorithm. Meanwhile, in order to keep the performance, two algorithms use the same set of initial data.

In the synthetic data: Servers are based on homogeneous framework. e_j^{idle} and e_j^{busy} are set to 162W and 215W. The synthetic data is configured according to the reference [9]. The items in dataset independently follow the normal distribution, which has been adopted in previous researches, including CPU demands generated with $N(0.15,0.05)$, Memory demands generated with $N(0.10,0.08)$ and Network demands generated with $N(0.03,0.01)$.

TABLE Ⅱ

AMAZON EC2 INSTANCE

Pattern	Instance type	vCPU	Memory	Pattern	Instance type	vCPU	Memory
General	T2.micro	1	1	General	M4.xlarge	4	16
	T2.small	1	2		M4.2xlarge	8	32
	T2.medium	2	4	Compute Optimized	C3.large	2	3.75
	M3.medium	1	3.75		C3.xlarge	4	7.5
	M3.large	2	7.5		C3.2xlarge	8	15
	M3.xlarge	4	15		C3.4xlarge	16	30
	M3.2xlarge	8	30	Memory optimized	R3.large	2	12.25
	M4.lagrge	2	8		R3.xlarge	4	30.5

(http://aws.amazon.com/cn/ec2/instance-types/)

In the real-world data: Sixteen types of instance (**Table Ⅱ**.) referring to Amazon EC2 are also used to simulate in Virtual Cloud Laboratory System, which are divided into three categories including general, computing optimized and memory optimized. Since the reference just contains CPU and memory except network, the network parameter will be set through simulation or get by using the data collection tools which are developed by shell (Linux) or C# (Windows).

B. Experimental Results

1) Experimental Results of Synthetic Instances

In the synthetic data, there are 500 physical machines in one data center. The initial habitat is set with 500. The number of virtual machines is 200. Next, the initial power consumption is 500*162W and the detail result information is presented in **Table Ⅲ** of synthetic data.

TABLE Ⅲ

EXPERIMENTAL RESULT OF SYNTHETIC AND REAL VM RUNNING DATA

Al	Synthetic data AS	CC	SE	MRO	Real VM running data AS	CC	SE	MRO
BBO	99	18037.83	-0.3208	123511.72	30	5827.59	-0.0718	82015.62
IMBBO(1)	**49**	**9937.83**	**-0.4351**	**14004.37**	**19**	**4045.58**	**-0.0891**	**3559.00**
IMBBO(2)	92	16903.83	-0.3090	65954.81	31	5989.58	-0.0740	52782.19
GSA	96	10909.83	-0.2883	42013.09	33	4531.59	-0.0711	7118.01

(**AL:** algorithms, **AS:** active servers, **CC:** cost consumption, **SE:** similarity evaluation, **MRO:** migration resource overhead)

The improved migration rate model and mutation rate model are adopted in IMBBO(1). Meanwhile, the improved migration rate model and constant probability value of mutation model are adopted in IMBBO(2). Finally, the IMBBO(1) algorithm only needs 49 active servers to support these VMs running.

The IMBBO(1) can reduce the power consumption from 500*162 to 9937.83. That is, it saves 87.71% of power.

The similarity evaluation value of IMBBO(1) is -0.4351, which is greatly smaller than others. This situation demonstrates that the algorithm can gain better performance and achieve the smallest similarity among virtual machines in order to reduce resource competition.

The migration resource overhead value of IMBBO(1) reach 14004.37 through 1000 iterations. The CPU extra utilization and transfer times in source and destination PMs are considered in this process. Finally, the IMBBO(1) algorithm can reduce the resource overhead in migration process compared others.

Fig.3(a)-(c) show the comparisons of the three algorithms on synthetic data. The IMBBO algorithm rapidly finds the best solutions. The blue cure presents the BBO. The green cure is the IMBBO(2). The sky blue cure is the GSA. However, the red cure is the IMBBO(1).

In conclusion, the IMBBO algorithm shows the better performance both in reducing power consumption, achieving good load balancing (minimizing similarity values among VMs) and decreasing the migration resource overhead. The reason is that IMBBO algorithm considers the special migration model and dynamic consolidation judgment strategy. At the same time, the mutation rate trends to much more stabilization with the iteration. Besides, the different configuration parameters may display the different performance in the experiment. These parameters need to be adjusted according to the real VM running situation. Finally, the GSA algorithm just focuses on finding the massive particles. The correlation of iteration doesn't been contained through the iteration.

2) Experimental Results of Real VM running Instances

In the real VM running environment, we set 80 VMs which simulate 80 students to attend class in Virtual Cloud Laboratory System. The initial configuration needs 150 servers. The experiment results are described in **Table Ⅲ** of real VM running data.

For experiment, The IMBBO(1) algorithm just needs 19 active servers to support these VMs running. The IMBBO(1) can reduce the power consumption from 150*162W to 4045.58. That is, it saves 83.36% of power. In addition, the similarity evaluation value of IMBBO(1) is -0.0891, which is greatly smaller than others in reducing resource utilization competition. Meanwhile, the migration resource overhead value of IMBBO(1) reach 3559.00 through 1000 iterations.

Fig.3(d)-(f) show the comparisons of the three algorithms on real VM running environment. These curve styles are the same as the synthetic instance. After the 1000 iterations, all of these curves approximately trend to line. Since a great number of VMs run in real data center environment, the appropriate running time and convergence time are the much more important parameter index.

Since these experiments are arranged based on homogeneous server architecture, some extra situations and different configuration parameters haven't been detailedly considered in process. It may exist others factors to influence the process of optimization. Besides, the emergency situation in some VMs also is the reason to affect optimization process.

Since the EA algorithms may lead to premature convergence (local optimization), including the BBO, the different strategies also should be adapted to avoid it. The appropriate parameters adjustment and the different situation analysis should be pre-done for the different objective functions.

VI. CONCLUSION

In this paper, the VMCP is formulated as a multi-objective optimization problem, which contains three configurable objectives, that is, server power consumption minimization, load balancing minimization based on resource utilization similarity and reducing migration resource overhead. A novel multi-objective optimization algorithm named IMBBO is proposed to solve VMCP based on the classical BBO. The migration and mutation model was improved, which can better meet the actual situation of VMCP. The experiment results show that the IMBBO can improve the performance of optimal solutions and convergence characteristic by comparing with GSA by using the synthetic and real VMs running data.

In future, we will focus on parallelization of IMBBO and research re-configuration migration and mutation model in real problem. The appropriate adjustment also should be used to solve the different objective function in order to avoid shortcoming of premature convergence.

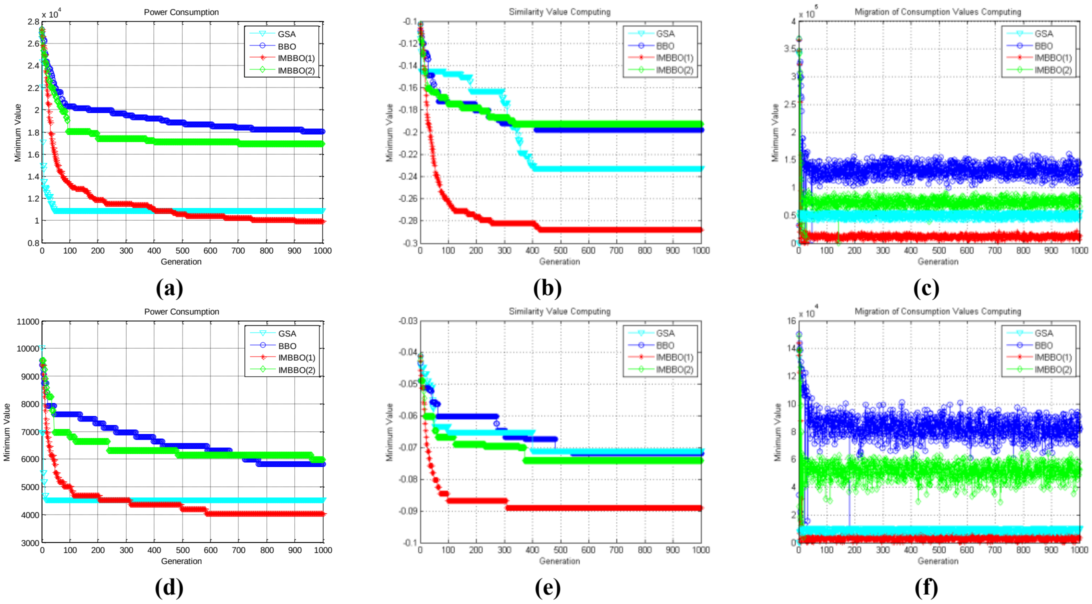


Fig.3 The experiment results of different comparing algorithms in synthetic and real VM running data

ACKNOWLEDGMENT

This research was partially supported by the Nature Science Fund of China under grants No. 61173048, 61300041, 61472139, Specialized Research Fund for Doctoral Program of Higher Education under grant No. 20130074110015.

REFERENCES

[1] Xu, F., et al. "Managing Performance Overhead of Virtual Machines in Cloud Computing: A Survey, State of the Art, and Future Directions."Proceedings of the IEEE 102.1(2014):11-31.

[2] Anton B. "Energy-Efficient Management of Virtual Machines in Data Centers for Cloud Computing" Ph.D. dissertation, Dept. Computing and Information Systems. Philosophy., Melbourne Univ., Melbourne, Australia, 2013.

[3] Dabbagh, M., et al. "Energy-efficient cloud resource management," Proc. IEEE INFOCOM Workshop on Mobile Cloud Computing (INFOCOM 2014), April 2014, pp. 386-391.

[4] Ahmad, R. W., et al. "Virtual machine migration in cloud data centers: a review, taxonomy, and open research issues." Journal of Supercomputing 71(2015):1-43.

[5] Hadji, M., et al. "A virtual machine repacking in clouds: faster live migration algorithms," Proc. ACM SIGCOMM workshop on Distributed cloud computing (SIGCOMM 2014), Aug. 2014, pp. 37-38.

[6] Hieu, N. T., et al. "Virtual Machine Consolidation with Usage Prediction for Energy-Efficient Cloud Data Centers." Cloud Computing (CLOUD), 2015 IEEE 8th International Conference on. IEEE, 2015.

[7] Ghribi, C., et al. "Energy efficient vm scheduling for cloud data centers: Exact allocation and migration algorithms." Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on. IEEE, 2013.

[8] Zhao, C., et al. "A Virtual Machine Dynamic Consolidation Algorithm Based Dynamic Complementation and FFD Algorithm." Communication Systems and Network Technologies (CSNT), 2015 Fifth International Conference on. IEEE, 2015.

[9] Zheng, Q., et al. "Virtual machine consolidated placement based on multi-objective biogeography-based optimization." Future Generation Computer Systems 54.C(2015):95-122.

[10] Zheng, Q., et al. "Multi-objective Optimization Algorithm Based on BBO for Virtual Machine Consolidation Problem." Parallel and Distributed Systems (ICPADS), 2015 IEEE 21st International Conference on IEEE, 2015.

[11] Mi, H., et al. "Online selfreconfiguration with performance guarantee for energy-efficient largescale cloud computing data centers," in Services Computing (SCC), 2010 IEEE International Conference on, July 2010, pp. 514–521.

[12] Gao, Y., et al. "A multi-objective ant colony system algorithm for virtual machine placement in cloud computing." Journal of Computer & System Sciences 79.8(2013):1230-1242.

[13] Jeyarani, R., et al. "Self adaptive particle swarm optimization for efficient virtual machine provisioning in cloud," Int. J. Intell. Inf. Technol., vol. 7, no. 2, pp. 25–44, Apr. 2011.

[14] Marotta, A., et al. "A Simulated Annealing Based Approach for Power Efficient Virtual Machines Consolidation." Cloud Computing (CLOUD), 2015 IEEE 8th International Conference on IEEE, 2015.

[15] Simon, D. "Biogeography-Based Optimization." Evolutionary Computation IEEE Transactions on 12.6(2009):702-713.

[16] Simon, D, et al. "Markov Models for Biogeography-Based Optimization."IEEE Transactions on Systems Man & Cybernetics Part B Cybernetics A Publication of the IEEE Systems Man & Cybernetics Society41.1(2011):299-306.

[17] Du, D., et al. "Biogeography-based optimization combined with evolutionary strategy and immigration refusal."Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on. IEEE, 2009.

[18] Ma, H., et al. "Biogeography-based optimization with blended migration for constrained optimization problems." Proceedings of the 12th annual conference on Genetic and evolutionary computation. ACM, 2010.

[19] Simon, D. "A dynamic system model of biogeography-based optimization." Applied Soft Computing 11.8(2011):5652-5661.

[20] Rarick, R., et al. "Biogeography-based optimization and the solution of the power flow problem." Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on. IEEE, 2009.

[21] Ma, H. "An analysis of the equilibrium of migration models for biogeography-based optimization." Information Sciences180.18(2010):3444–3464.

[22] Ma, H., et al. "Blended biogeography-based optimization for constrained optimization." Engineering Applications of Artificial Intelligence 24.3(2011):517-525.