The Journal of Supercomputing
https://doi.org/10.1007/s11227-023-05136-z

Thermal-aware virtual machine placement based on multi-objective optimization

Bo Liu[1] · Rui Chen[1] · Weiwei Lin[2,3] · Wentai Wu[3] · Jianpeng Lin[2] · Keqin Li[4]

Accepted: 22 February 2023

Abstract

VMP (Virtual Machine Placement) is a crucial technology for energy consumption optimization of the cloud data center. Existing works mainly focus on virtual machine consolidation to increase resource utilization and reduce computing energy consumption. However, existing studies usually ignore the thermal effect that an intensive workload on IT (Information Technology) equipment can raise energy consumption by cooling systems and generate hotspots. In addition, an excessive number of virtual machine migrations increases migration costs and risks violating the SLA (Service Level Agreement) signed with users. In this paper, we present a comprehensive system model and formulate the problem as a constrained multi-objective optimization. We propose a novel thermal-aware VMP strategy to solve the problem by jointly considering virtual machines' migration cost, energy consumption, and heat recirculation around server racks. Our strategy makes placement decisions using MOPFGA (Multi-objective algorithm based on Pathfinder Algorithm and Genetic Algorithm) that combines classic MOPFA and GA enhanced by OBL (Opposition Based Learning) for fast convergence and avoidance of local optimum. Extensive experiments based on CloudSim using real data center workload data from PlanetLab show that our algorithm overcomes the defects of the MOPFA (multi-objective pathfinder algorithm) and GA (genetic algorithm) and significantly improves the overall efficiency of a data center. Compared with several state-of-the-art algorithms, MOPFGA on average reduces virtual machine migrations by 77.52%, increases CRAC (Computer Room Air Conditioner) supply temperature by 1.24%, and reduces cooling energy consumption by 24.78% and computational energy consumption by 23.62%.

✉ Weiwei Lin
linww@scut.edu.cn

✉ Wentai Wu
wentai_wu@outlook.com

Extended author information available on the last page of the article

Published online: 15 March 2023

Keywords Cloud computing · Temperature · Virtual machine placement · Pathfinder algorithm · Energy efficiency · Data centers

1 Introduction

With the rapid growth of cloud computing's popularity, the demand for computation and storage resources is ever-increasing. Meanwhile, expanding cloud data centers has led to massive energy consumption and carbon emissions, raising global concerns. It was predicted that the energy consumption of global data centers will increase by 66% from 2011 to 2035, and the power consumption of data centers will account for about 2% of the total in the United States [1, 2]. Excessive energy usage inevitably leads to high operating costs—Amazon estimates that energy costs accounts for over 42% of the total operation of their data centers [3].

It is imperative for Cloud providers to schedule workload to minimize total system energy of both computing and cooling to reduce operational costs. Generally, the energy consumption of a data center is attributed to the energy consumption of computing equipment (e.g., servers, network) and that of non-computing equipment (e.g., cooling system). On the one hand, physical servers consume the majority of the power supply to the server room. Data center takes advantage of the virtualization technology to provide multiple VM (virtual machine) resources for the cloud service, which is helpful for flexible resource management over the entire infrastructure. Through VM consolidation, the low-workload hosts are shut down or shifted to a low energy consumption mode, with the hosted VMs migrated to other hosts. VM consolidation optimizes load distribution and improves computing resources' power efficiency. However, frequent VM migrations have a lot of negative impacts on the data center [4, 5]. For example, VM migration may undermine the performance of VMs and even temporarily interrupt their service. The migration process incurs a considerable cost of data transmission, which indirectly leads to additional energy consumption and increases the probability of violating SLA. These negative factors due to MC (Virtual machine migration cost) complicate the decision of VM placement [6].On the other hand, the cooling system in the server room can, in some cases, consume more energy than the computing infrastructure due to excessive VM consolidation that leads to the high operating temperature of local nodes [7, 8]. Moreover, the heat-recirculation effect of the server room intensifies the generation of hotspots [8–10]. In order to keep the overall temperature of the data center below the red line temperature (i.e., the maximum allowable operating temperature of the equipment), the CRAC (Computer Room Air Conditioner) must continue to operate at a high load level, which also leads to a sharp increase in overall energy consumption.

To deal with the above problems, optimizing VMP is an effective method. VMP is the process of placing a large number of VMs on appropriate physical hosts to improve operating efficiency and reduce management costs. Traditional VMP strategies typically focus on tackling the workload scheduling problem with a single objective of saving computational energy consumption and hardly consider two crucial aspects, the migration cost of the VM and the cooling cost. Jointly reducing the energy consumption of computing and non-computing devices has become a significant challenge in cloud data center management.

Because of the problems above, this paper proposes a thermal-aware VMP strategy, in which a novel algorithm called MOPFGA for optimized VM placement with the aim of reducing the VM migration cost and the total energy consumption of the data center under the premise of ensuring the temperature constraint. The proposed strategy is distinct from existing solutions in three aspects. First, the strategy is designed to solve a multi-objective optimization problem of energy consumption reduction and VM migration cost reduction while ensuring a low SLA violation rate. Second, the cooling system's energy consumption the energy consumption is modeled in consideration of the heat-recirculation effect that impacts the ambient temperature and in turn the cooling efficiency. Third, by integrating MOPFA [11] and GA [12] with the OBL (Opposition-based Learning) technique [13], the proposed strategy effectively improves the decisions of VMP by optimizing global search efficiency and avoiding early convergence to local optimum in the process of decision search.

The main contributions of this paper are summarized as follows:

1. We propose a new thermal-aware VMP strategy that jointly considers the migration cost of VMs, the power models of servers and the cooling system as well as the heat-recirculation effect.
2. We propose a multi-objective VM placement algorithm MOPFGA as a hybrid solution based on the MOPFA and GA. Compared with the traditional MOPFA, we introduce the crossover and mutation operations of reverse learning and genetic algorithm in MOPFGA, which improves over MOPFA in avoiding local optimum and GA in convergence speed.
3. We conduct a series of simulation-based experiments to verify the effectiveness of MOPFGA. The results show that compared with the existing algorithms, including MITEC-GA, FC-BGSA, PPABFD, ETAS, RACC, and SABA, MOPFGA significantly reduces the overall energy consumption and the cost of VM migrations at a low SLA violation rate while keeping a controllable probability of hotspots.

The rest of the paper is organized as follows: Sect. 2 discusses the related work on VM consolidation. Section 3 describes the system model and problem formulation. Section 4 details the proposed thermal-aware VMP strategy. Section 5 experimentally validates the effectiveness of our MOPFGA. Finally, we conclude the paper in Sect. 6.

2 Related work

In order to reduce the total energy consumption of data centers, many factors need to be considered in scheduling VM migration, such as cooling system efficiency, heat recirculation, and SLA violation rate. However, the high migration cost of VMs also leads to a large amount of waste of resources, so the negative impact of migrations on the server room environment and the cooling system should also be considered. Although much existing related work has studied migration cost control technology and thermal management technology, they have yet to explore how to reduce energy consumption to an ideal level while considering the migration cost and thermal recycling effect. This section summarizes the existing VM placement strategies considering migration costs and temperature factors.

2.1 VM placement based on migration cost in data centers

It is shown in the literature [5] that VM migration can cause additional energy consumption and performance degradation. Many efforts are devoted to reducing the operation cost of the data center by controlling the number of virtual machine migrations. Tao et al. [14] regarded the dynamic VM migration problem as a three-objective optimization model and designed a bucket code learning algorithm based on binary graph matching. Mann et al. [15] proposed to use constraint programming techniques for VMP based on the number of migrations and overloaded CPUs when integrating virtual machines. Rym et al. [16] proposed a solution based on a multi-objective mixed integer linear programming model (MOMILP) for virtual machine placement, aiming at simultaneously minimizing the VM rejection ratio, the amount of wasted resources and the number of used PMs. However, this work lacks a discussion on bandwidth, which will lead to an increase in the SLA violation rate. Xu et al. [17] proposed the migration cost-aware virtual machine integration algorithm. However, they only focused on the number of physical hosts running rather than designing the algorithm from the perspective of reducing energy consumption. Mapetu et al. [6] adopted HIB (Host Imbalance Degree), considering the bandwidth of VM to reduce the number of virtual machine migrations. Hariharan et al. [18] proposed a multi-objective equation based on VM migration cost and proposed an ABSO VM placement algorithm by improving Beetle Swarm Optimization. However, this work does not discuss the metric of the SLA violation rate. Ding et al. [19] combined PPR (Performance-to-power-ratio) proposed host overload, low load detection algorithm, and VM placement algorithm, committed to reducing energy consumption and migration costs. Monireh et al. [20] introduced the DTMC (Discrete-time Markov Chain) model to predict the server's resource utilization efficiency to prevent excessive VM consolidation and proposed an ϵ-MOABC (ϵ-dominance-based multi-objective artificial bee colony) for VM placement. Wang et al. [21] proposed a VM selection policy called AUMT that selects VM with minimum cost in the combination of both average CPU utilization and migration time

2.2 Thermal-aware VM placement in data centers

Due to the increasing energy cost of the data center cooling system, thermal-aware resource management technology has become the research focus. The reference [22] points out that the unreasonable design of the cooling system in the data center will lead to heat-recirculation due to the lousy phenomenon of cooling efficiency reduction caused by the mixing of cold and hot air. The objective of the thermal-aware distribution scheme is to minimize the cooling system cost of the data center while maximizing the supply temperature of CRAC cold air. Sun et al. [23] introduced the concept of thermal-aware load and proposed an online scheduling heuristic algorithm for task scheduling and thermal management. Li Xiang et al. [24] designed a cooling model considering CRAC, air, and server thermal characteristics in the data center using CFD. They also designed a VM placement and migration algorithm called GRANITE to minimize computing and cooling resources. Ilager et al. [25] proposed energy consumption and heat-aware scheduling algorithm ETAS, a meta-heuristic algorithm based on GRASP. ETAS can integrate virtual machines greedily in the case of active prevention of hotspots. Feng et al. [10] proposed a multi-objective VM placement algorithm considering computation, cooling, and network energy consumption. However, they only consider the energy consumption of computation and cooling in the initial placement stage of the VM. In the VM placement process, they only consider the energy consumption of network equipment, and the computation and cooling equipment is lacking. In addition, the work ignores the SLA violation rate metric. In another work [26], they proposed a VM scheduling algorithm based on the heat-recirculation effect. Experiments show that the proposed algorithm can save cooling costs but does not consider the cost of computing equipment. In addition, the work [27] proposed a new failure model that leverages data center heat recirculation. This model considers the spatial distribution, time distribution, and the relevant change of the cross-interference matrix with failures. However, this work also does not consider the impact of migration cost and SLA. Xiao et al. [28] proposed a machine learning approach for VM placement, but they only counted the energy cost of the cooling system in experiments; without considering the temperature constraints and SLA violation, they could not avoid server losses due to high temperatures and the lack of stable service. Aghasi et al. [29] proposed a VM placement algorithm based on BGSA to minimize computational and cooling costs and designed an adaptive fuzzy mechanism to enhance the algorithm. However, there was no apparent advantage in reducing the SLA violation rate. Li et al. [30] innovatively added a fault model to the thermal model but did not consider an increase in SLA violation due to VM migration.

The comparison of the related work can be found in Table 1. Unlike the VMP strategy based on the above two resource management schemes, the proposed thermal-aware VMP strategy considers both thermal-aware and VM migration costs to optimize VM consolidation. The advantages of the VM placement algorithm MOPFGA are mainly reflected in three aspects. Firstly, MOPFGA depends on the air temperature distribution in the computer room, not only considering the computational cost but also the cooling energy consumption. Secondly, MOPFGA also integrates the VM migration cost model while managing resources. Finally, MOPFGA complies with resource constraints in virtual machine consolidation to avoid increasing the SLA violation rate.

Table 1 Related work

	Migration cost based	Thermal-aware	Considering SLAV
Tao et al. [14]	✓	✖	✖
Mann et al. [15]	✓	✖	✓
Rym et al. [16]	✓	✖	✖
Xu et al. [17]	✓	✖	✓
Mapetu et al. [6]	✓	✖	✓
Hariharan et al. [18]	✓	✖	✖
Ding et al. [19]	✓	✖	✓
Monireh et al. [20]	✓	✖	✓
Wang et al. [21]	✓	✖	✓
Sun et al. [23]	✖	✓	✓
Li Xiang et al. [24]	✖	✓	✖
Ilager et al. [25]	✖	✓	✓
Feng et al. [10]	✖	✓	✖
Feng et al. [26]	✖	✓	✓
Feng et al. [27]	✖	✓	✖
Xiao et al. [28]	✖	✓	✖
Aghasi et al. [29]	✖	✓	✓
Li et al. [30]	✖	✓	✖
Our work	✓	✓	✓

3 System model and problem description

Data centers are usually composed of computing devices (e.g., servers, networks) and non-computing devices (e.g., cooling systems). The energy consumption for computation is not only attributed to the capacity of physical hosts but is also affected by VM resources. In addition, the heat-recirculation effect in the server room leads to the generation of hotspots, and the cooling system needs to be dynamically adjusted to ensure that the temperature of the host is within a safe range. Therefore, to propose the thermal-aware VMP strategy, we first propose a workload model to represent the computational resource constraints in the system and minimize the migration cost target. Then, to reflect the effect of computer room heat-recirculation on VM placement, we propose a heat-recirculation model to ensure that the computer room can run within thermal constraints. Finally, a power consumption model considering computational and the cooling energy consumption is proposed according to the above model.

3.1 Workload model

3.1.1 Physical machine resource model

Each heterogeneous host in the data center corresponds to a specific computing resource capacity. For the host h_i at time t, quadruple $\text{Capacity}_t^{h_i} = (mips, storage, ram, bw)$ represents the remaining computing resources, where $\text{Capacity}_t^{h_i}$ represents the host h_i at time t. *mips* represents the available performance capacity in Mips (Million Instructions Per Second), *storage* represents the remaining storage capacity, *ram* represents the remaining internal storage capacity, and *bw* represents the remaining valid bandwidth. Vector represents the remaining resources of all hosts at time t. When the resource utilization rate of the host approaches the limit, the VM performance level on the host will be limited by the host capacity, increasing SLA violation time.

All hosts in the computer room are expressed as:

$$M = \{h_1, h_2, ...h_m\} \tag{1}$$

Equation (1) denotes the data center host set M; the total number of computer room hosts is m.

3.1.2 VM resource model

The computing resources of vm_k request at time t are expressed as quadruple:

$$\text{Request}^{vm_k}(t) = (mips, storage, ram, bw) \tag{2}$$

In Eq. (2), $\text{Request}_k^{\text{vm}}(t)$ denotes the VM vm_k at t time.

The increase in the number of VM migrations leads to increased migration costs [18]. VM migration cost is expressed as:

$$MC = \frac{1}{m} \sum_{i=1}^{m} \left(\frac{\text{Total number of VM}_i}{\text{Number of migration in VM}_i} \right) \tag{3}$$

In Eq. (3), MC represents the migration cost and VM_i represents a VM instance on host h_i. Consolidating VMs will lead to performance degradation, leading to an increase in the SLA violation rate.

3.2 Heat-recirculation model

As shown in Fig. 1, the cold air from CRAC is mainly sent to the main engine's entrance through the room's floor tile. The cold air flows through the host inlet and the hot air from the host outlet circulates into CRAC. However, the hot air from the host outlet will lead to heat-recirculation, increasing the inlet temperature of the host and even generating hotspots. Therefore, CRAC is required to guarantee a lower

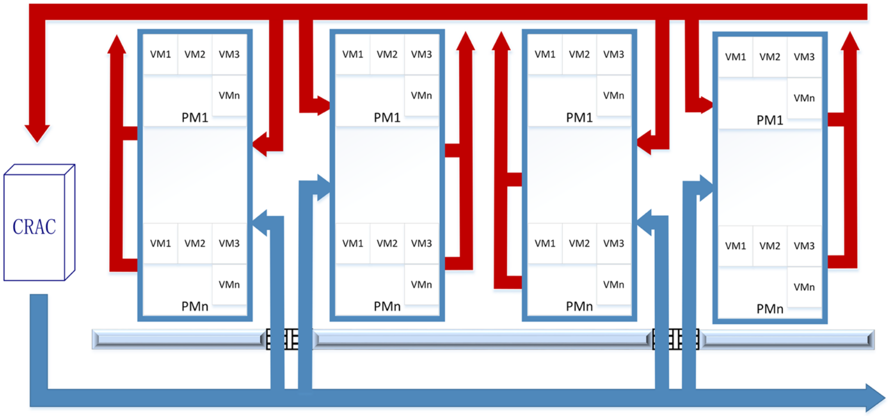


Fig. 1 An illustration of heat-recirculation phenomenon in a data center server room

temperature of cold air supply to maintain the host inlet temperature below the red line temperature. This phenomenon leads to more cold air energy consumption. This paper uses the low complexity heat-recirculation model proposed by Tang et al. [8]. The model calculates the equipment's power based on the data center's thermodynamic and physical characteristics.

In order to calculate the CPU temperature considering the data center's temperature, we should calculate the inlet temperature of the host. The inlet temperature of the host is mainly affected by the CRAC air supply temperature, server power consumption, and heat-recirculation effect. We define the inlet temperature of the host as the following linear function:

$$T_i^{\text{in}}(t) = T_{\text{sup}} + \sum_{j=1}^{m} d_{i,j} \times P_j(t) \tag{4}$$

In Eq. (4), $d_{i,j}$ represents the degree that the inlet temperature of the host h_i is affected by the host h_j. It is the number of i rows and j columns in the thermal distribution matrix D, which is the cooling supply temperature of CRAC. Equation (4) shows that the inlet temperature of the host is affected by its physical position and heat-recirculation effect.

The thermodynamic distribution matrix is defined as $D \equiv [(K - A^T K)^{-1} - K^{-1}]$, where $K = \text{diag}(K_1, K_2, ...K_m)$ is the thermodynamic coefficient matrix, and $A = a_{ij}$ denotes the heat from the outlet of each host to the inlet of other hosts.

CPU temperature modeling is the most important indicator of temperature modeling, the and RC model [8] is one of the mature methods for calculating CPU temperature. Its model can be expressed as:

$$T_i(t) = PR + T_i^{\text{in}} + (T_0 - PR - T_i^{\text{in}}) \times e^{-\frac{t}{RC}} \tag{5}$$

In Eq. (5), $T_i(t)$ is the CPU temperature of the host at time t. R and C are the server's thermal resistance and heat capacity, respectively. P is the energy consumption of the active host, and T_0 is the initial temperature of the CPU. The RC model assumes that the power and inlet temperature of the CPU is stable, and the stable CPU temperature is expressed by $PR + T_i^{\text{in}}$. At the same time t, it is also a function related to time t. With the increase of time, the CPU will continue approaching its stable temperature. Equation (5) shows the dynamic temperature change of a single host in a heat-recirculation environment.

3.3 Power model

Based on the models defined in the last section, we can formulate the computing and cooling system energy consumption given a cooling system supply temperature.

3.3.1 Computing system power model

A Cloud data center comprises heterogeneous servers with different physical capacities, power, and processing capabilities. The power consumption of the host is mainly determined by its CPU utilization. We use the following power consumption model to represent the power consumption of a single active host [8]:

$$P_i(t) = \begin{cases} P_i^{\text{idle}} + \sum_{j=1}^{V_i} U(\text{vm}_{i,k}(t)) \times P_i^{\text{dynamic}} & (V_i > 0) \\ 0 & (V_i = 0) \end{cases} \tag{6}$$

In Eq. (6), $P_i(t)$ represents the energy consumption of host h_i at time t ; P_i^{idle} represents energy consumption without load ; P_i^{dynamic} represents energy consumption at full load; $U(\text{vm}_{i,k}(t))$ represents the utilization rate of computing resources when the VM k runs in h_i at time t; V_i is the total number of VMs running in h_i. It is necessary to shut down the idle host ($V_i = 0$) to save unnecessary energy consumption.

3.3.2 Cooling system power model

CRAC is the primary cooling equipment of cloud data centers and accounts for most of the cooling energy cost [7]. The efficiency of CRAC is usually measured by calculating the power consumption ratio of the system to the cooling system, also known as the Coefficient of Performance (CoP):

$$\text{CoP}(T_{\text{sup}}) = \frac{P_{IT}}{P_{\text{cooling}}}, \tag{7}$$

where P_{cooling} is the power of the cooling system, P_{IT} is the power consumption of the IT system, and T_{sup} is the temperature of the CRAC's cold air supply.

A higher CoP value indicates higher cooling efficiency. Studies have shown that CoP positively correlates with the temperature of the cold supply air. We use the CoP determined by HP laboratory [31]:

$$\text{CoP}(T_{\text{sup}}) = 0.0068 \cdot T_{\text{sup}}^2 + 0.0008 \cdot T_{\text{sup}} + 0.458, \tag{8}$$

where T_{sup} is CRAC cold air supply temperature. Equation (8) shows that increasing the cooling supply temperature can improve the cooling system's efficiency.

3.4 Problem definition

The total energy consumption of a cloud data center is mainly composed of computing system energy consumption and cooling system energy consumption. The sum of the energy consumption of all servers is the energy consumption of the computing system, which can be expressed as:

$$P_{IT} = \sum_{t=0}^{T} \sum_{i=1}^{m} x_j P_i(t), \tag{9}$$

where $P_i(t)$ is the calculated energy consumption of host h from Eq. (6), and x_j is a binary variable. When h_i is active from time 0 to t, its value is 1; otherwise, it is 0. Timely shutting down the low-load host can save unnecessary overhead, so the key is adjusting the host workload allocation.

In order to maintain a reliable running state, the inlet temperature of each host needs to be kept below the threshold:

$$T_i^{\text{in}}(t) \le T_{\text{redline}} \tag{10}$$

From Eqs. (7) and (8), it can be seen that CoP is positively correlated with T_{sup}. Under the same calculation of energy consumption, higher T_{sup} can consume less cooling cost. Combined with Eqs. (5) and (10), it can be obtained:

$$T_{\text{sup}} \le T_{\text{redline}} - \sum_{j=1}^{m} d_{i,j} \times P_j(t) \tag{11}$$

The energy consumption of cooling system is defined as the ratio of calculated energy consumption to CoP:

$$P_{\text{cooling}} = \frac{P_{IT}}{\text{CoP}(T_{\text{sup}})} \tag{12}$$

It can be seen from Eq. (12) that in order to provide colder airflow, the cooling system needs to consume more energy to remove the heat brought by the host computation. The total energy consumption is expressed by Eq. (13):

$$P_{\text{total}} = P_{IT} + P_{\text{cooling}} \tag{13}$$

In this paper, the problem is formulated as a workload scheduling problem that aims to minimize the total energy consumption and migration cost under the constraints of server utilization, host CPU critical temperature, and server workload (represented by VMs).

$$\begin{aligned}
&\text{Minimize } P_{\text{total}} = P_{IT} + P_{\text{cooling}} = \sum_{t=0}^{t_{\text{total}}} \sum_{p=1}^{m} \left(1 + \frac{1}{\text{CoP}}\right) \cdot P_{IT} \\
&\text{Minimize } MC = \frac{1}{m} \sum_{i=1}^{m} \left(\frac{\text{Total number of VM}_i}{\text{Number of migration in VM}_i} \right) \\
&S.t.\ \forall i = 1, 2, .., m, \sum_{k=1}^{\text{VM}_{h_i}} \sum_{t=0}^{t_{\text{total}}} \text{Request}_{t}^{\text{vm}_k} \le \text{Capacity}^{h_i(t)} \\
&T_{\text{sup}} \le T_{\text{redline}} - \sum_{j=1}^{m} d_{i,j} \times P_j(t)
\end{aligned} \tag{14}$$

where VM_{h_i} is a collection of VMs running on the host h_i. The constraint conditions in Eq. (14) ensure that the CPU utilization and temperature of the host do not exceed the threshold due to the increase in workload, at the same time, ensure that the target host selected for the VM can meet the resource requirements of this VM.

4 Thermal-aware VMP strategy

In the data center, workloads submitted by different users are assigned to different VMs hosted by servers in the server room according to the allocation policy. During the operation of the server, a large amount of heat is released, leading to the heat-recirculation phenomenon and resulting in increased temperature of the entire room and thus affecting the inlet temperature of each server [8]. Excessive inlet temperature may affect the reliability and safety of the host. In order to keep the computer room at a relatively stable temperature, CRAC needs to reduce the supply temperature of cold air. However, it also leads to an increase in cooling energy consumption. In order to reduce the calculation and cooling energy consumption of the computer room, the placement strategy needs to schedule the VM to the target server reasonably and close or adjust the low-load server to sleep mode to save unnecessary energy consumption [30]. At the same time, it is necessary to actively prevent hotspots caused by excessive VM consolidation, which will cause server loss. In the process of VM migration, the system dynamically adjusts the CRAC cooling supply temperature according to the host CPU temperature detected at any time to prevent the waste of cooling energy. Based on this, appropriate regulation can be carried out in the VM placement stage to minimize the host inlet temperature and reduce the cooling cost.

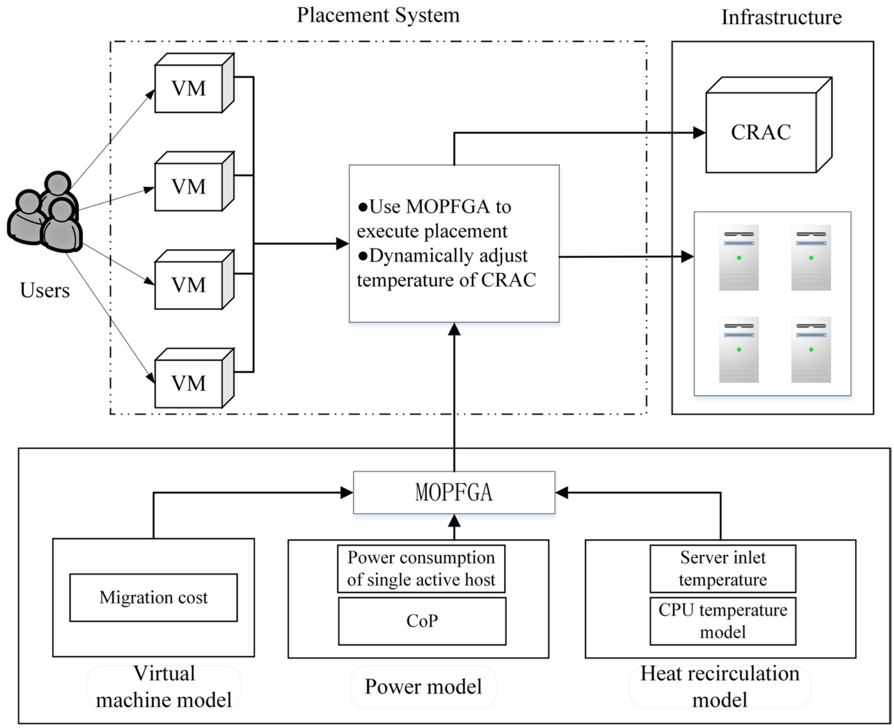


Fig. 2 An overview of the proposed thermal-aware VMP strategy

Figure 2 illustrates the overall workflow of the proposed thermal-aware VMP strategy. The process entails three roles, namely users, the placement system, and the infrastructure. Users submit requests to the deployed VM. The placement system is responsible for processing user requests, allocating computing resources for requests, and monitoring the CPU utilization and temperature of the host. At the same time, the VM, energy consumption, and heat-recirculation model provide parameters and constraints for the operation of the MOPFGA algorithm. Infrastructure includes computing equipment, cooling equipment, and other physical equipment. The system dynamically adjusts the supply temperature of CRAC to prevent the host temperature from exceeding the threshold. The procedure for placing a VM takes the following steps:

1. VM requests submission: Users submit VM requests to perform one or more tasks.
2. Host selection: The placement system uses MOPFGA to select the data center server to place the VM.
3. Placement decision execution: the execution of VM placement based on MOPFGA.

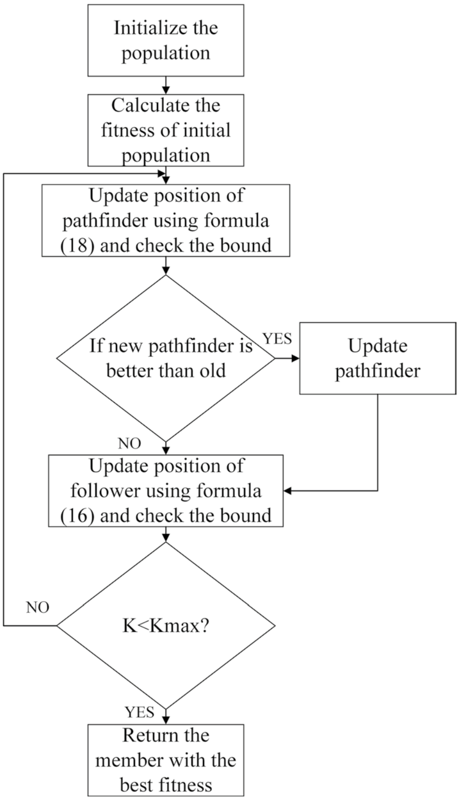
4. System resource information update: after placing the VM, the system updates the host and VM resource information in the data center.

In our thermal-aware VMP strategy, the placement system uses MOPFGA, a multi-objective VM placement algorithm based on MOPFA [11] and GA [12]. MOPFGA aims to minimize computational and cooling energy consumption while reducing the effect of heat-recirculation. This algorithm overcomes the characteristics of MOPFA that it quickly falls into local optimum and slow GA iteration and introduces the OBL (Opposition-based Learning) method [13] to increase the global search ability of the algorithm and reduce the probability that the algorithm is trapped in a local optimum. In order to study the MOPFGA algorithm, we first introduce the classic MOPFA algorithm [11].

4.1 Traditional MOPFA algorithm

The pathfinder algorithm [11] is inspired by the behavior of animal groups in nature. Usually, there is a pathfinder in an animal group as a minority of individuals who know information about food sources, hunting areas, and routes. On the contrary, other

Fig. 3 Process of MOPFA



individuals in the population are called followers. In addition to following the pathfinder, they are also affected by other individuals in the population. Figure 3 shows the process of the MOPFA algorithm. A swarm intelligence algorithm that shares search space information through pathfinder leadership and members' movement. For a population of size *pop*, each member x has its current location:

$$\text{Population}^K = (x_1^K, x_2^K, \ldots x_{\text{pop}}^K), \tag{15}$$

where K is the number of iterations of the algorithm. The movement of a follower of the population is random and affected by other members:

$$x_i^{K+1} = x_i^K + R_1 \cdot (x_j^K - x_i^K) + R_2 \cdot (x_p^K - x_i^K) + \epsilon \; i \geq 2 \tag{16}$$

where $R_1 = \alpha \cdot r_1$, $R_2 = \alpha \cdot r_2 . r_1$,$r_2$ are random numbers in [0,1]. α and β represent the degree of influence between adjacent members and set them to the random number in [1, 2, 11]. x_j represents the corresponding x_i in the previous iteration. ϵ guaranteed random movement of members:

$$\epsilon = (1 - \frac{K}{K_{\max}}) \cdot u_1 \cdot D_{ij} \tag{17}$$

In Eq. (17), u_1 is the random number in [–1,1]. $D_{ij} = \|x_i - x_j\|$ represents Euclidean distance between two members. There is always a pathfinder in the population, updated as follows:

$$x_p^{K+1} = x_p^K + 2 \cdot r_3 \cdot (x_p^K - x_p^{K-1}) + A, \tag{18}$$

where r_3 is the random number in [0, 1] and A is generated by Eq. (19):

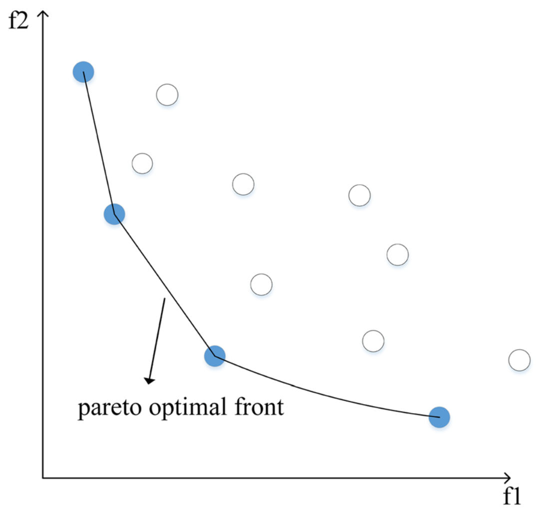


Fig. 4 Pareto optimal front

VM1	VM2	VM3	...	VMn
PM542	PM65	PM169	...	PM23

Fig. 5 Examples of population members

$$A = u_2 \cdot \frac{-2K}{K_{max}}, \tag{19}$$

where u_2 is a random number in [–1,1].

Instead of a single solution in single-objective problems, multi-objective algorithms include different solutions, the so-called Pareto Optimal Front. The Pareto optimal front is defined as the set of non-dominated solutions where each objective is considered equally good, as shown in Fig. 4. Compared with other meta-heuristic algorithms, MOPFGA can be seen that MOPFA can close to the true Pareto optimal front [11]. In addition, we have formulated VMP as a multi-objective optimization problem to minimize energy consumption and migration cost, so MOPFA is a suitable algorithm for VMP.

4.2 MOPFGA algorithm

On the one hand, we have formulated VMP as a multi-objective optimization problem to minimize energy consumption and migration cost, so we need to transform the traditional MOPFA algorithm into a multi-objective algorithm that can be applied to VMP. On the other hand, the traditional MOPFA algorithm has the defects of easy loss of population diversity and easy to fall into local optimum. It needs to be continuously improved to improve the versatility of the algorithm. Therefore, the proposed MOPFGA algorithm combines the MOPFA and the crossover and mutation operations of GA. Furthermore, this paper introduces the OBL strategy, which can improve the slow convergence defects of the traditional MOPFA and GA. In the scheduling process, MOPFGA migrates the VMs in the overloaded and low-load hosts to the selected target host to save unnecessary energy consumption.

Step 1. Solution encoding: initializing the solution to a string code is the first step in the optimization problem [29]. First, MOPFGA generates a group of population and a pathfinder randomly. All members of the population represent a solution. The solution is a sequence with the length of the number of VMs to be migrated. A code is randomly generated for each location representing the VM's target host. An example of the encoding format for the solution is shown in Fig. 5.

Step 2. Fitness calculation: Multi-objective optimization algorithms often use Pareto optimality to compare fitness. However, in VMP, there is usually no single solution but Pareto Optimal Front. The Pareto front is defined as the set of non-dominated solutions where each objective is considered equally good, as shown in Fig. 4. Calculating Pareto optimal front will make it difficult for the algorithm to obtain an individual optimal solution. In addition, computing the Pareto optimal frontier will significantly increase the algorithm's complexity. We calculate the fitness of each member by calculating the sum of the ratio of energy consumption and migration costs to their maximum:

$$\text{totalFitness} = \frac{\text{maxPowerFitness}}{\text{powerFitness}} + \frac{\text{maxMcFitness}}{\text{mcFitness}}, \tag{20}$$

where totalFitness is the total fitness corresponding to each member's solution. powerFitness is the fitness of energy consumption by calculating the increased energy consumption after allocation. maxPowerFitness is the maximum energy consumption fitness. mcFitness is the migration cost fitness by calculating the increased migration cost after allocation. maxMcFitness is the fitness of maximum energy consumption.

Step 3. Improved pathfinder update with OBL: As the number of iterations increases, the population members are surrounded by pathfinders, and the range of activities gets limited. As a result, traditional MOPFA tends to stop at local optima at a decreased convergence rate. This paper introduces the OBL strategy to improve the local optimization strategy of pathfinders. The principle is to search for the direction solution based on the current solution and to compete for the current solution with the reverse solution to leave a more favorable solution. The mathematical expressions are as follows:

$$b = \max(x_p^{K+1}), \quad a = \min(x_p^{K+1}) \tag{21}$$

$$\text{oppx}_p^{K+1} = \text{rand} \cdot (b + a) - x_p^{K+1} \tag{22}$$

In Eq. (21), b and a are the maximum and minimum values in the range of x_p^{K+1}, respectively; oppx_p^{K+1} is the reverse solution; rand is the random number in the range of [0, 1]. We use OBL to generate a new set of reverse candidate solutions, extending the range of candidate solutions for pathfinders. Therefore, the algorithm can find better new solutions on a broader search range to improve the convergence ability of the algorithm in the late iteration.

Step 4. Improved follower updating using GA: GA is used to expand its local search range and make its search more accurate when updating followers. It mainly includes two parts: crossover and mutation. Crossover exchanges partial sequences of two chromosomes (member's solution) to generate two new sub-chromosomes. The mutation operation mutates partial sequences of the two chromosomes. Finally, the better of the two sub-chromosomes are left as the updated followers.

Step 5. VM placement: After the MOPFGA iteration, the VM is assigned to the target host according to the solution corresponding to the members with the lowest fitness value.

The worst-case time complexity of MOPFGA is $O\big(K \cdot \big(\text{pop} \cdot \log(\text{pop}) + \text{Num}_{\text{vms}} \times \text{pop} + (F) \times \text{pop}\big)\big)$, where K is the number of iterations, pop is the number of population, Num_{vms} is the number of VMs to be migrated, and F is the cost of calculating the fitness (which is roughly equal to $\mathrm{Num}_{\mathrm{vms}}$).

4.3 Algorithmic detail

As shown in algorithm 1. First, we initialize the population randomly (line 1). Then we calculate the fitness of each initial member and find the initial Pathfinder (lines 2 to 3). After entering the iteration, we use the quick sort algorithm to sort the population members in ascending order according to their fitness (line 5). First, update the Pathfinder, use Eq. (18) to get the position of the new Pathfinder, and then use OBL to get the reverse solution. Compare the two to leave a better solution, and then compare the new solution with the old Pathfinder to leave a better solution as the new Pathfinder (lines 6 to 10). Traverse all followers, use Eq. (16) to get the position of the new follower, use GA's crossover and mutation operation (algorithm 2) to get the sub-solution, and compare the two to leave the better solution as the new follower (lines 11 to 14). Then traverse the population and find the optimal solution to replace the Pathfinder (line 15). Finally, MOPFGA returns the optimal members of the population (line 17).

Algorithm 1 *MOPFGA*

Input: VMs to migrate, host list, max number of iterations K_{max}
Output: VM-host mapping
1: Initialize the population
2: Calculate the fitness of initial population
3: Assign the best member as the pathfinder
4: **while** $K \leq K_{max}$ **do**
5: Sort the population by fitness
6: Use Eq. (18) to obtain the new position of pathfinder x_p^{K+1} and check the bound
7: Use Eq. (21) and Eq. (22) to obtain the opposite solution and keep the better one
8: **if** If new pathfinder is better than the old one **then**
9: Update the pathfinder
10: **end if**
11: **for** $i = 2$ to max size of population **do**
12: Use Eq. (16) to obtain the new position of followers x_i^{K+1} and check the bound
13: **CrossoverAndMutate**(x_p^{K+1}, x_i^{K+1})
14: **end for**
15: Traverse the population and find the optimal members to replace the pathfinder
16: **end while**
17: **Return** the optimal member of population

The details of GA crossover and mutation operations are shown in Algorithm 2.

Algorithm 2 CrossoverAndMutate

Input: x_p^{K+1}, x_i^{K+1}
Output: $newx_i^{K+1}$
Set the solution of x_p^{K+1} as $chromosome1$ and solution of x_i^{K+1} as $chromosome2$
Generate a random number τ_1
Crossover at the τ_1-th position of the two chromosomes
Generate random numbers τ_2 and τ_3
Set the τ_2-th position of chromosome1 as τ_3
Generate random numbers τ_4 and τ_5
Set the τ_4-th position of chromosome1 as τ_5
Compare the two chromosomes and assign the better one to $newx_i^{K+1}$
Return $newx_i^{K+1}$

5 Experiments and results

5.1 Experimental setup

We establish a simulation-based experiment environment similar to the actual data center to evaluate the effectiveness of the proposed VMP strategy. The cloud data center is set to contain 10 zones, each composed of 10 racks. The racks are arranged according to the layout of 5×2 grids, and each rack has ten servers. It is assumed that each zone is affected by the heat-recirculation effect, and the host temperature is affected by other hosts. The experiment uses the thermal distribution matrix used by Tang et al. [8]. The entire data center contains 1000 hosts. Each host is an IBM x3550 M3 machine with 8 processors and 4GB of RAM. The hosts are equipped with two types of processors: Intel Xeon X5670 (6 judges, 2.93 GHz, 12MB L3 Cache) processor and Intel Xeon X5675 (6 judges, 3.07 GHz, 12MB L3 Cache) processor. According to the instance type of Amazon EC2 [32], we set four types of single-core virtual machines, as shown in Table 2.

In order to evaluate the performance of dynamic migration, detailed settings of our experiments are listed in Table 3. We adopt data center room layout as per the work of Tang et al. We assume that the heat-recirculation effect exists in each region, and the thermal distribution matrix provided by Tang et al. [8] is used to simulate the thermal cycle effect in the region. This paper applies it to the experiment. According to ASHRAE[31], the CPU T_{redline} in Eq. (10) is set to 25°C. In Eq. (5), the heat capacity and resistance are set to 340 J/K and 0.34 K/W, respectively, and the initial CPU temperature is set to 318 K [33].

Our simulation-based experiment uses the real data center workload obtained by the PlanetLab system [34], which tracks the utilization data of multiple VMs

Table 2 VM types

VM size	Core	Processing speed (MIPS)	Ram (MB)	Band-width (Mbits/s)
Extra Large	1	870	870	100
Large	1	2000	1740	100
Micro	1	1000	1740	100
Nano	1	500	613	100

in more than 500 places around the world. In this paper, the actual workload trace with a sampling window of 10 days in the CoMon project is used to evaluate the proposed scheme. The workloads of all VMs are reported every 5 minutes in the dataset, as shown in Table 4. All algorithms are written in Java programming language, running on the Core i5-8500 CPU, 3.00GHz, 16GB RAM machine.

Table 3 Experimental parameters

Item		Value
Data center	Number of zones	20
	Number of racks in zone	10
	Number of hosts in rack	5
nServer	Heat capacity	340[J/K]
	Thermal resistance	0.34[K/W]
	Initial CPU temperature	318K
	Threshold of CPU temperature	25°C
Simulation	Simulation time	24hours
	Simulation platform	CloudSim V.4.0

Table 4 Workload data characteristics

Workload datasets	Date	Number of VMs	Mean(%)	SD(%)
CoMon project workload trace	03/03/2011	1052	12.31	17.09
	06/03/2011	898	11.44	16.83
	09/03/2011	1061	10.70	15.57
	22/03/2011	1516	9.26	12.78
	25/03/2011	1078	10.56	14.14
	03/04/2011	1463	12.39	16.55
	09/04/2011	1358	11.12	15.09
	11/04/2011	1233	11.56	15.07
	12/04/2011	1054	11.54	15.15
	20/04/2011	1033	10.43	15.21

5.2 Metrics

The experiment evaluates the efficiency of the proposed scheme based on six standard metrics: energy consumption, migration cost, SLAV, ESV, ESM, and number of host shutdowns.

Energy: This metric represents the energy consumption of each scheme in kWh.

Migration cost (MC): This metric represents the number of VM migrations during the experiment.

SLAV: This metric indicates the performance overhead caused by the dynamic consolidation of VMs. Meeting QoS requirements is significant for Cloud computing environments. Since the minimum throughput, maximum response time and bandwidth delivered by the system can vary depending on the application running. It is necessary to define a workload-independent metric $\text{SLA}_{\text{violation}}$.When the utilization rate of the host is close to 100%, the VM performance level on the host will be limited by the host capacity, which can be described as SLA violation time for each active host. When the utilization rate of the host is close to 100%, the VM performance level on the host will be limited by host capacity, which can be described as SLA violation time SLA_{TAH} for each active host. In addition, the consolidation of VMs will lead to performance degradation, which can be described as Performance Degradation due to Migration (PDM).

$$\text{SLA}_{\text{TAH}} = \frac{1}{N}\sum_{i=1}^{N}\frac{t_{si}}{t_{ai}} \quad \text{PDM} = \frac{1}{M}\sum_{j=1}^{M}\frac{C_{d_j}}{C_{r_j}} \tag{23}$$

$$\text{SLAV} = \text{SLA}_{\text{TAH}} \times \text{PDM} \tag{24}$$

In Eq. (23), the N is the number of hosts; t_{si} is the total time during which the h_i has experienced the utilization of 100% leading to an SLA violation; t_{ai} is the total of the h_i being in the active state; M is the number of VMs; C_{d_j} is the estimate of the performance degradation of the jth VM caused by migrations; C_{r_j} is the total CPU capacity (MIPS) requested by the jth VM during its lifetime. The overall SLA violation SLA for cloud infrastructure can be obtained by combining SLA_{TAH} and PDM, as shown in Eq. (24).

ESV (energy and SLA violations): This metric is the product of energy consumption and SLA violation rate. The goal is to compare the energy consumption and SLA violation rate of the proposed algorithm and the benchmark algorithm at the same time. If ESV is significantly reduced, the energy consumption and SLAV reach a balance.

$$\text{ESV} = \text{EC} \times \text{SLAV} \tag{25}$$

ESM (energy, SLA violations and migration): ESM is a comprehensive metric of SLA violation rate and VM migration cost. The goal is to simultaneously compare the proposed and baseline algorithms' energy consumption, SLA violation rate, and VM migration cost.

$$\mathrm{ESM} = \mathrm{EC} \times \mathrm{ESV} \times MC \tag{26}$$

The number of host shutdowns: This metric indicates the number of VM migrations during the experiment.

5.3 Baseline algorithms

We verify the effectiveness of MOPFGA by comparing it with several baseline algorithms. We implement the five state-of-the-art VM placement algorithms:

PPABFD: Mapetu et al. [6] improved the power-aware best fit reducing (PABFD) [35] algorithm based on the Pearson coefficient. Optimize the placement of VMs by calculating the Pearson coefficients of CPU, ram, and bandwidth.

ETAS: A dynamic VM placement algorithm was proposed by Ilager et al. [25]. This algorithm uses GRASP, a meta-heuristic online scheduling algorithm, to reduce energy consumption while preventing hotspots.

RACC: Ding et al. [19] proposed a VM consolidation framework based on resource utilization and heterogeneous host PPR. In the VM placement phase, RACC (Residual Available Computing Capacity) based technology detects overloaded hosts. This framework can effectively solve the trade-off between host computing overhead and performance.

IGA-POP: An improved permutation-based genetic algorithm (IGA-POP) proposed by Abohamama et al. [36] proposed a VM placement algorithm by balancing exploration and utilization in search space.

MOPFA: Metaheuristic algorithm for solving multi-objective problems proposed by Yapici et al. [11]. To prove the effectiveness of OBL, crossover, and mutation operations of MOPFGA, we compare the proposed algorithm with the traditional MOPFA.

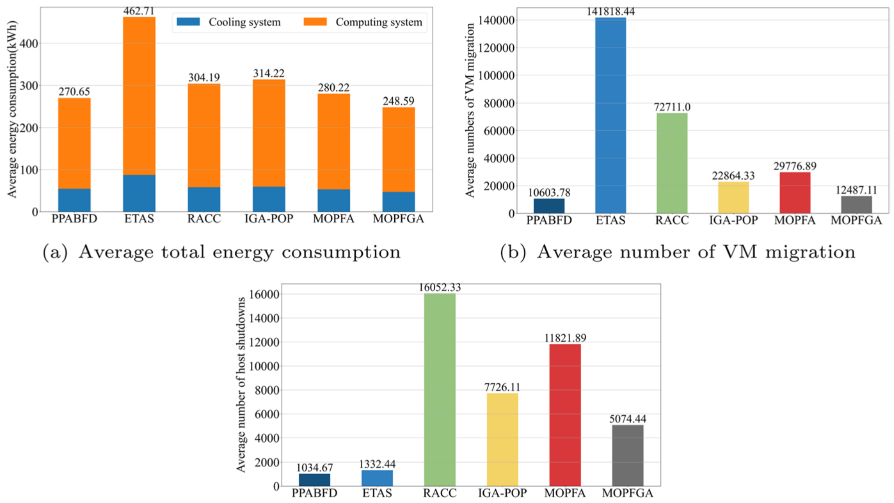
The setting of algorithm parameters and VM selection criterion is shown in Table 5. In addition, through experimental analysis, the CPU utilization threshold of all algorithms is set to 80%.

5.4 Experimental results and analysis

Figure 6 compares the average total energy consumption, the average number of MC, and the average number of closed hosts between MOPFGA and the baseline

Table 5 Algorithm parameters and VM selection criterion

Algorithm	Parameter	Value	VM selection criterion
MOPFGA and MOPF	Population size	50	Minimum Migration Time (MMT) [35]
	Number of iteration	300	
PPABFD	α	0.4	
	ϵ	0.1	
IGA-POP	Population size	50	
	Number of iteration	300	



(a) Average total energy consumption

(b) Average number of VM migration

(c) Average number of host shutdowns

Fig. 6 Average total energy consumption, average number of MC and average number of host shutdowns

algorithm in all workloads. As shown in Fig. 6a, it is observed that MOPFGA has a remarkable effect on limiting energy consumption, which reduces the total energy consumption by 23.84% on average compared with all the baseline algorithms. The cooling energy consumption decreases by 24.78% on average, and the calculated energy consumption decreases by 23.62% on average. ETAS consumes the most energy. It uses GRASP to improve the greedy selection of the target host and reduces the time complexity of selecting the host when the VM is placed. However, it also reduces the possibility of finding the optimal solution, increases a large number of VM migrations costs, and opens a large number of unnecessary hosts (Fig. 6b and c). Compared with PPABFD and RACC, they tend to choose the greedy strategy of the host with minor energy consumption, and the energy consumption of IGA-POP based on GA is slightly higher than that of the first two. RACC algorithm is more inclined to place VMs on high PPR hosts than traditional PABFD, which will cause VMs to concentrate on high PPR hosts and close many low PPR hosts, so RACC causes the most host closures and high energy consumption (Fig. 6a and c). PPABFD has the characteristics of a greedy selection of PABFD and uses the Pearson coefficient to reduce the probability of host utilization overload and indirectly reduce the energy consumption caused by VM migrations. In PPABFD, the host with a lower Pearson coefficient is more likely to become the target host of VM, which can reduce the probability of host overload, which is also the reason for less VM migration. However, the algorithm does not consider the thermal factor, making generating hotspots quickly (Fig. 6b and c). Compared with the baseline algorithm, MOPFA takes energy consumption and migration cost as the dual goals to reduce the excessive migrations of VMs while reducing the total energy consumption and the number of MC by 77.52% compared with all the baseline algorithms. On this basis, MOPFGA adds the operation of GA and OBL to make the search in the iteration more accurate. The results show that MOPFGA is better than the traditional MOPFA in each metric because of the combination of GA and OBL to improve the ability to search for the global optimum.

The SLAV, ESV, and ESM of MOPFGA are slightly higher than PPABFD and far lower than other baseline algorithms as shown in Table 6. The SLAV, ESV and ESM decreased by 96.799%, 98.210% and 98.962%, respectively. PPABFD calculates different CPU, RAM, and bandwidth to obtain the Pearson coefficient, which significantly avoids the resource conflict caused by VM allocation and obtains the best SLA violation rate level. However, the energy consumption and maximum cooling temperature of PPABFD are much higher than those of MOPFGA. In addition, the experimental results show that the allocation strategy of MOPFGA significantly reduces the SLA violation rate level and energy consumption caused by excessive VM consolidation by reducing the VM migrations cost. In particular, compared with the MOPFA results, it can be found that the improved algorithm can find the optimal global solution more accurately.

The average supply temperature of each algorithm at different times is shown in Fig. 7. Higher CRAC cooling supply temperature means saving more cooling energy [36]. Each increase in temperature can save about 2% to 5% of energy consumption [37]. Among them, the proposed MOPFGA and ETAS provide the highest cooling supply temperature, but ETAS performs more inefficiently in terms of energy consumption. Although PPABFD performs well in SLA, it needs a lower cooling supply temperature to prevent hotspots in the computer

Table 6 Average SLAV, ESV and ESM

SLAV($\times 10^{-5}$)	PPABFD	4.70354
	ETAS	1070.9352
	RACC	146.4901
	IGA-POP	36.724
	MOPFA	47.1506
	MOPFGA	8.35979
	Average improvement(%)	96.799
ESV($\times 10^{-3}$)	PPABFD	12.349101
	ETAS	4919.810467
	RACC	429.82569
	IGA-POP	112.191745
	MOPFA	130.865773
	MOPFGA	20.068973
	Average improvement(%)	98.210
ESM	PPABFD	3.375331202
	ETAS	2170.501065
	RACC	129.8087274
	IGA-POP	35.28042807
	MOPFA	37.42260659
	MOPFGA	4.935329674
	Average improvement(%)	98.962

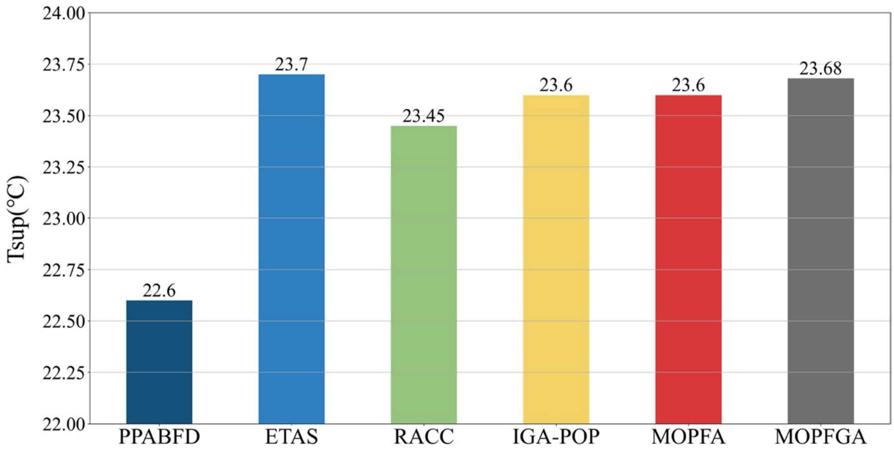
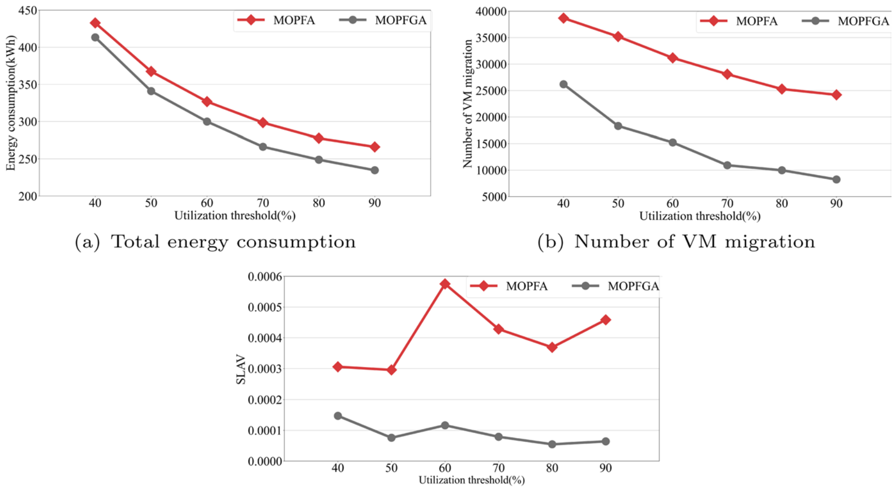


Fig. 7 Average CRAC cooling supply temperature of different algorithms

room. PPABFD is too negative for virtual machine migration to make the host temperature overload. In addition, RACC, IGA-POP, and MOPFA supply temperature slightly lower than MOPFGA. It can be seen that the proposed MOPFGA not only considers the computational energy consumption of the host but also reduces the possibility of hotspots by balancing the workload. Compared with all benchmark algorithms, MOPFGA increases the cooling supply temperature by 1.24% on average.

The proposed algorithm is affected by the utilization threshold level. In order to analyze the influence of different utilization thresholds and reflect the algorithm's improvement effect, we conducted a sensitivity analysis and found the best parameter setting. We take the CoMon project workload of 03/03/2011 as an example. The performance of MOPFA and MOPFGA under different CPU utilization thresholds is shown in Fig. 8. A smaller CPU utilization threshold requires more energy to run more hosts to complete the workload. At the same time, the lower CPU utilization makes the host can accommodate the number of VMs reduced, and VM resource requirements can not be met in a short time, forcing the number of VM migrations between hosts to increase. Therefore, a high threshold is needed to utilize data center resources to save energy efficiently. In addition, it can be found that the SLA violation rate level increases when the threshold reaches 90% because although the host can accommodate more VMs in CPU utilization, other resources (such as storage capacity, ram, bandwidth, etc.) cannot meet the needs of all VMs. Therefore, we set the experiment's CPU utilization threshold at 80%.

In conclusion, the MOPFGA not only reduces the computing energy consumption of the data center but also saves much cooling system energy consumption. MOPFGA increases the CRAC supply-air temperature to a higher level, with the server inlet temperature not exceeding the red line temperature. Also, avoid excessive VM consolidation to reduce the SLA violation rate.



(a) Total energy consumption

(b) Number of VM migration

(c) SLAV

Fig. 8 Comparison of total energy consumption, number of VM migrations and SLAV under different utilization thresholds

6 Conclusion

In this paper, we consider the VMP in a data center as a multi-objective optimization problem and propose a VMP strategy that aims at improving the energy efficiency of data centers by comprehensively considering the workload, computing resources, cooling overhead, and heat-recirculation effect. The strategy features a multi-objective VM placement algorithm MOPFGA developed to improve the data center's energy consumption and thermal efficiency while ensuring thermal constraints. As a combination of the classic MOPFA and GA, the proposed MOPFGA integrates OBL to prevent premature convergence to local optima and thus significantly improves the quality of VMP solutions. Extensive experiments on real-world workload data from the PlanetLab system show that MOPFGA outperforms MOPFA, PPABFD, ETAS, RACC, and IGA-POP regarding total energy consumption, SLA violation rate, and CRAC supply temperature, which is mainly attributed to the awareness of heat recirculation for our solution. As an insight, we also observe that frequent migration of VMs will increase the data center's energy cost and SLA violation rate.

On this basis, we plan to expand the model in the future further, such as using a more accurate fluid dynamics/heat transfer model (CFD/HT) model or a data-driven model to predict the thermal distribution of the computer room. At the same time, after some modifications, the proposed algorithm can be applied to different scenarios, such as fan speed and cooling system failure. We can add the joint optimization of VMP strategy and dynamic control cooling technology. According to the heat distribution of the cloud data center, the system can reduce the waste of cooling system resources by controlling the VM distribution and dynamically adjusting the fan speed of the host chassis. We will continue to study related issues in future work.

Author Contributions B. Liu, R. Chen, and J. Lin wrote the main body of the paper and carried out the experiments. W. Wu re-organized the paper structure and polished the writing. W. Lin and K. Li advised on the whole research. All authors reviewed the manuscript.

Funding This work is supported by the Key-Area Research and Development Program of Guangdong Province (2021B0101420002), National Natural Science Foundation of China (62072187), Guangdong Major Project of Basic and Applied Basic Research (2019B030302002), Guangdong Marine Economic Development Special Fund Project (GDNRC[2022]17), PCL Project (PCL2021A09), and Guangzhou Development Zone Science and Technology (2021GH10, 2020GH10).

Data Availability Statement We experimented with the PlanetLab workload data which is publicly available.

Declarations

Conflict of interest Not applicable.

Ethical approval Not applicable.

References

1. Lin W, Shi F, Wu W et al (2020) A taxonomy and survey of power models and power modeling for cloud servers[J]. ACM Comput Surv (CSUR) 53(5):1–41
2. Kamiyama N (2019) Virtual machine trading in public clouds[J]. IEEE Trans Netw Serv Manag 17(1):403–415
3. Teng F, Yu L, Li T, Deng D, Magoule's F (2017) Energy efficiency of VM consolidation in IAAS clouds. J Supercomput 73(2):782–809
4. Guo Z, Yao W, Wang D (2017) A virtual machine migration algorithm based on group selection in cloud data center[C]. In: IFIP International Conference on Network and Parallel Computing. Springer, Cham, pp 24-36
5. Buyya R, Beloglazov A, Abawajy J (2010) Energy-efficient management of data center resources for cloud computing: a vision, architectural elements, and open challenges[J]. arXiv preprint arXiv:1006.0308
6. Mapetu JPB, Kong L, Chen Z (2021) A dynamic VM consolidation approach based on load balancing using Pearson correlation in cloud computing[J]. J Supercomput 77(6):5840–5881
7. Zhang Q, Meng Z, Hong X et al (2021) A survey on data center cooling systems: technology, power consumption modeling and control strategy optimization. J Syst Archit. 119:102253
8. Tang Q, Gupta SKS, Varsamopoulos G (2008) Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: a cyber-physical approach. IEEE Trans Parallel Distrib Syst 19(11):1458–1472
9. Li J, Deng Y, Zhou Y et al (2022) Towards thermal-aware workload distribution in cloud data centers based on failure models[J]. IEEE Trans Comput. https://doi.org/10.1109/TC.2022.3158476
10. Feng H, Deng Y, Zhou Y et al (2021) Towards heat-recirculation-aware virtual machine placement in data centers[J]. IEEE Trans Netw Serv Manag 19:256–270
11. Yapici H, Cetinkaya N (2019) A new meta-heuristic optimizer: pathfinder algorithm[J]. Appl Soft Comput 78:545–568
12. Mirjalili S (2019) Genetic algorithm[M]. Evolutionary algorithms and neural networks. Springer, Cham, pp 43–55
13. Mahdavi S, Rahnamayan S, Deb K (2018) Opposition based learning: a literature review[J]. Swarm Evol Comput 39:1–23
14. Tao F, Li C, Liao TW et al (2015) BGM-BLA: a new algorithm for dynamic migration of virtual machines in cloud computing[J]. IEEE Trans Serv Comput 9(6):910–925
15. Mann ZÁ (2016) Multicore-aware virtual machine placement in cloud data centers[J]. IEEE Trans Comput 65(11):3357–3369

16. Regaieg R, Koubàa M, Osei-Opoku E et al. (2018) Multi-objective mixed integer linear programming model for vm placement to minimize resource wastage in a heterogeneous cloud provider data center[C]. In: 2018 10th International Conference on Ubiquitous and Future Networks (ICUFN). IEEE, pp 401-406
17. Xu H, Liu Y, Wei W et al (2019) Migration cost and energy-aware virtual machine consolidation under cloud environments considering remaining runtime[J]. Int J Parallel Program 47(3):481–501
18. Hariharan B, Siva R, Kaliraj S et al (2021) ABSO: an energy-efficient multi-objective VM consolidation using adaptive beetle swarm optimization on cloud environment[J]. J Ambient Intell Humaniz Comput. https://doi.org/10.1007/s12652-020-02740-2
19. Ding W, Luo F, Han L et al (2020) Adaptive virtual machine consolidation framework based on performance-to-power ratio in cloud data centers[J]. Future Gener Comput Syst 111:254–270
20. Sayadnavard MH, Haghighat AT, Rahmani AM (2022) A multi-objective approach for energy-efficient and reliable dynamic VM consolidation in cloud data centers[J]. Eng Sci Technol Int J 26:100995
21. Wang J, Gu H, Yu J et al (2022) Research on virtual machine consolidation strategy based on combined prediction and energy-aware in cloud computing platform[J]. J Cloud Comput 11(1):1–18
22. Shuja J, Gani A, Shamshirband S et al (2016) Sustainable cloud data centers: a survey of enabling techniques and technologies. Renew Sustain Energy Rev 62:195–214
23. Sun H, Stolf P, Pierson JM (2017) Spatio-temporal thermal-aware scheduling for homogeneous high-performance computing datacenters. Future Gener Comput Syst 71:157–170
24. Li X, Jiang X, Garraghan P et al (2018) Holistic energy and failure aware workload scheduling in cloud datacenters[J]. Future Gener Comput Syst 78:887–900
25. Ilager S, Ramamohanarao K, Buyya R (2019) ETAS: energy and thermal-aware dynamic virtual machine consolidation in cloud data center with proactive hotspot mitigation. Concurr Comput Pract Exp 31(17):e5221
26. Feng H, Deng Y, Li J (2021) A global-energy-aware virtual machine placement strategy for cloud data centers[J]. J Syst Archit 116:102048
27. Feng H, Deng Y, Yu L (2021) Modeling the failures of power-aware data centers by leveraging heat recirculation[J]. Concurr Comput Pract Exp 33(7):1–1
28. Xiao P, Ni Z, Liu D et al (2021) A power and thermal-aware virtual machine management framework based on machine learning[J]. Clust Comput 24(3):2231–2248
29. Aghasi A, Jamshidi K, Bohlooli A (2022) A thermal-aware energy-efficient virtual machine placement algorithm based on fuzzy controlled binary gravitational search algorithm (FC-BGSA)[J]. Clust Comput. https://doi.org/10.1007/s10586-021-03476-0
30. Li J, Deng Y, Zhou Y et al (2022) Towards thermal-aware workload distribution in cloud data centers based on failure models[J]. IEEE Trans Comput 72:586–599
31. Moore JD, Chase JS, Ranganathan P et al. (2005) Making scheduling" cool": temperature-aware workload placement in data centers[C]. In: USENIX Annual Technical Conference, General Track, pp 61-75
32. Humane P, Varshapriya JN (2015) Simulation of cloud infrastructure using CloudSim simulator: a practical approach for researchers. In: International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), Controls, Energyand Materials, IEEE, pp 207-211
33. ASHRAE (2018) American society of heating, refrigerating and air-conditioning engineers. http://tc0909.ashraetcs.org/
34. Park KS, Pai VS (2006) CoMon: a mostly-scalable monitoring system for planet lab. ACM Sigops Oper Syst Rev 40(1):65–74
35. Beloglazov A, Buyya R (2012) Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. Concurr Comput Pract Exp 24(13):1397–1420
36. Abohamama AS, Hamouda E (2020) A hybrid energy-aware virtual machine placement algorithm for cloud environments[J]. Expert Syst Appl 150:113306
37. El-Sayed N, Stefanovici IA, Amvrosiadis G, Hwang AA, Schroeder B (2012) Temperature management in data centers: why some (might) like it hot. ACM Sigmetr Perform Eval Rev 40(1):163–174

Authors and Affiliations

Bo Liu[1] · Rui Chen[1] · Weiwei Lin[2,3] · Wentai Wu[3] · Jianpeng Lin[2] · Keqin Li[4]

Bo Liu
liugubin@sohu.com

Rui Chen
2020022956@m.scnu.edu.cn

Jianpeng Lin
csjianpenglin@mail.scut.edu.cn

Keqin Li
lik@newpaltz.edu

1 School of Computer Science, South China Normal University, Guangzhou 510631, China

2 School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China

3 Peng Cheng Laboratory, Shenzhen 518000, China

4 Department of Computer Science, State University of New York, New Paltz, NY 12561, USA