Expert Systems With Applications 150 (2020) 113306

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A hybrid energy–Aware virtual machine placement algorithm for cloud environments

A.S. Abohamama[a,1,*], Eslam Hamouda[a,b,2]

[a] Assistant Professor, Computer Science Department, Mansoura University, Postal Code: 35516, Egypt
[b] Assistant Professor, Computer Science Department, Jouf University, Saudi Arabia

ARTICLE INFO

Article history:
Received 9 July 2019
Revised 26 January 2020
Accepted 10 February 2020
Available online 11 February 2020

Keywords:
Cloud computing
Server consolidation
Virtual machine placement
Permutation-based optimization

ABSTRACT

The high energy consumption of cloud data centers presents a significant challenge from both economic and environmental perspectives. Server consolidation using virtualization technology is widely used to reduce the energy consumption rates of data centers. Efficient Virtual Machine Placement (VMP) plays an important role in server consolidation technology. VMP is an NP-hard problem for which optimal solutions are not possible, even for small-scale data centers. In this paper, a hybrid VMP algorithm is proposed based on another proposed improved permutation-based genetic algorithm and multidimensional resource-aware best fit allocation strategy. The proposed VMP algorithm aims to improve the energy consumption rate of cloud data centers through minimizing the number of active servers that host Virtual Machines (VMs). Additionally, the proposed VMP algorithm attempts to achieve balanced usage of the multidimensional resources (CPU, RAM, and Bandwidth) of active servers, which in turn, reduces resource wastage. The performance of both proposed algorithms are validated through intensive experiments. The obtained results show that the proposed improved permutation-based genetic algorithm outperforms several other permutation-based algorithms on two classical problems (the Traveling Salesman Problem and the Flow Shop Scheduling Problem) using various standard datasets. Additionally, this study shows that the proposed hybrid VMP algorithm has promising energy saving and resource wastage performance compared to other heuristics and metaheuristics. Moreover, this study reveals that the proposed VMP algorithm achieves a balanced usage of the multidimensional resources of active servers while others cannot.

1. Introduction

Cloud computing is a well-known computing model that hosts and delivers a wide range of services through the internet. Cloud providers mainly offer three types of services: infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS). Recently, many enterprises relied on cloud infrastructure rather than in-house infrastructure to benefit from the advantages offered by cloud environments such as using a pay-as-you-go pricing model, eliminating maintenance burdens, and achieving on-demand scalability (Abohamama, Alrahmawy & Elsoud, 2018). Data centers are used in cloud environments to provide cloud services. However, these data centers consume large amounts of energy in their operations. In the U.S., the annual energy consumption of data centers was 91 billion kWh and it is expected to increase to 140 billion kWh by 2020 (Alharbi et al., 2019). Another study revealed that the energy consumption of data centers around the world was approximately 1.1–1.3% of the worldwide total and the rate was expected to increase to 8% by 2020 (Liu et al., 2018). Such a rapid increase in the energy consumption of data centers has become not only a critical economic issue but also a critical environmental issue. According to Amazon's studies about its data centers, the energy costs are approximately 42% of the total operational costs (Gao, Guan, Qi, Hou & Liu, 2013; Tang & Pan, 2015). Another motivation for energy conservation is the current debate on climate change. It is estimated that running servers contribute 0.5% of the global CO_2 emissions (Speitkamp et al., 2010). Hence, reducing the energy consumption of data centers without sacrificing the quality of the offered services is an emerging research area.

Data centers are usually equipped with large numbers of physical servers. Almost 60% of the total energy consumption in a data

* Corresponding author: Assistant Professor, Computer Science Department, Mansoura University, Postal Code: 35516, Egypt

E-mail addresses: abohamama@mans.edu.eg (A.S. Abohamama), eslam_foad@mans.edu.eg (E. Hamouda).

[1,2] Present address: 60 El-Gomhoreya Street, University of Mansoura, Faculty of Computers and Information Sciences, Mansoura, 35516, Egypt.

[2] Jouf University, College of Computer and Information Sciences, 2014, Saudi Arabia.

https://doi.org/10.1016/j.eswa.2020.113306
0957-4174/

center comes from the IT infrastructure, which is dominated by Physical Machines (PMs). Therefore, minimizing the number of active PMs in a data center will highly improve the energy consumption rate. Virtualization is an enabling technology in which cloud computing resources are provided to customers in the form of an unlimited number of virtual machines (VMs) based on a set of service level agreements (SLAs) established between cloud providers and customers (Abohamama, Alrahmawy, Elsoud & Hamza, 2018). Virtualization plays a crucial role in achieving both high server utilization and energy efficiency since several VMs can be allocated to the same physical server. In a virtualized cloud environment, the number of required active servers depends on the placement of the Virtual Machines (VMs). Hence, using an efficient Virtual Machine Placement (VMP) algorithm can have a great influence on a data center's power consumption.

VMP algorithms attempt to find the optimal allocation of VMs to PMs that achieves their design objectives. Many design objectives have been considered in the literature such as optimizing the power consumption, improving the resource utilization, minimizing SLA violations, etc. The VMP problem can formulated as a variant of the bin-packing problem, which is an NP-hard optimization problem. Therefore, several metaheuristic techniques such as the Firefly Algorithm (FA), Glowworm Swarm Optimization (GSO), and biogeography-based optimization (BBO) have been employed to generate efficient solutions within a reasonable time (Yan, Zhang, Xu & Zhang, 2018).

However, despite several VMP algorithms (Abdel-Basset, Abdle-Fatah & Sangaiah, 2018; Alharbi et al., 2019; Liu et al., 2018) having been proposed to address the energy consumption optimization problem in cloud environments, they do guarantee the balanced usage of multidimensional resources among active PMs. These algorithms may assign different amounts of residual resources for each resource type of each PM. In anticipation of future requests, the resources left on each PM should be balanced along the different dimensions. Otherwise, unbalanced residual resources may prevent any further VM placement, thus wasting computing resources (Zheng et al., 2016).

The static VMP problem can be formulated as a permutation-based optimization problem. The order suggested by each permutation can significantly affect the performance of the VM placement algorithm. Hence, an efficient permutation-based optimization algorithm is needed. One contribution of this paper is proposing a new permutation-based genetic algorithm called the Improved Genetic Algorithm for Permutation-based Optimization Problems (IGA-POP). The IGA-POP can efficiently solve permutation-based problems by balancing the exploration and exploitation in the search space. Its performance has been validated against several permutation-based optimization algorithms over two well-known classical problems. The main contribution of this paper is proposing a hybrid VMP algorithm that depends on the proposed IGA-POP and multidimensional resource-aware best fit (BF) allocation strategy. The proposed VMP algorithm aims to improve the energy consumption rate of data centers through minimizing the number of active physical servers and switching off the idle servers. Moreover, it aims to achieve the balanced usage of the multidimensional resources (CPU, RAM, and Bandwidth) of active servers, which in turn, reduces the resource wastage rate. The contributions of this study are summarized as follows:

– Proposing a new permutation-based genetic algorithm (IGA-POP) that can effectively address the permutation-based problems by balancing the exploration and exploitation in the search space,
– Proposing a hybrid VMP algorithm based on the proposed permutation-based genetic algorithm and a multidimension resource-aware best fit (BF) allocation strategy, and
– Improving the power consumption and resource wastage in cloud environments through minimizing the number of physical machines used to host the VMs and balancing the usage of different resources (CPU, RAM, and Bandwidth) within each used server.

In this paper, the proposed algorithm is used effectively to address the VMP problem in cloud environments. However, it can be employed to solve many problems in the scope of expert and intelligent systems. Expert Systems (ESs) are a branch of AI that attempt to use human expertise to mimic human experts when solving problems by reasoning about the knowledge. The intervention of soft computing tools (techniques) in expert systems has greatly improved their performance (Omisore, Samuel & Atajeromavwo, 2017). The proposed algorithm can be effectively employed in permutation-based expert systems such as production scheduling (flow shop, job shop, and open shop problems) (Kanet & Adelsberger, 1987; Sauve & Collinot, 1987; Xu, Wang, Wang & Liu, 2014) in manufacturing environments. Additionally, it can be employed in Home Health Care (HHC) systems that deliver medical, paramedical and social services to patients in their homes rather than in a hospital. Caregiver routing, which outlines the order in which each caregiver will visit the patients assigned to him/her, is an important decision in planning such systems (Liu, Xie, Augusto & Rodriguez, 2013; López-Santana & Méndez-Giraldo, 2016). Moreover, the proposed algorithm can be used to build a location allocation model for health care facility planning (Shariff, Moin & Omar, 2012).

The remaining sections of this paper are arranged as follows. Section 2 presents some recent studies in the field. Section 3 presents the problem formulation and the proposed algorithms. Section 4 includes the conducted experiments and results analysis. Eventually, the paper is concluded and the future work is given in Section 5.

2. Literature review

VMP is one of the cloud computing challenges that affect many aspects of cloud environments. Therefore, many studies have been conducted to optimize the VM placement among the available physical servers. Various objectives have been considered across different studies such as power consumption minimization, network traffic minimization, utilization rate maximization, quality of service maximization, economical revenue maximization, etc. Various methods have been proposed to address the VMP problem including deterministic algorithms (Chaisiri, Lee & Niyato, 2009; Alicherry & Lakshman, 2013; Dang & Hermenier, 2013) and more intelligent metaheuristic algorithms (Abdel-Basset et al., 2018; Gao et al., 2013; Liu, Sui & Li, 2016).

First-Fit Decreasing (FFD) and its variations such as Best-Fit Decreasing (BFD) are heuristic algorithms that were employed early on to address the VMP problem. In these heuristics, the VMP problem has been formulated as a classic NP-hard bin packing problem. These algorithms are simple and have low computational complexity compared to optimization algorithms. However, the solutions generated by these algorithms are low quality since they do not consider the different objectives that need to be optimized (Alharbi et al., 2019). In addition, the resulting solutions do not achieve the balanced usage of the different resources within the physical machines as our proposed work does. Additionally, the VMP problem was addressed as a linear programming (LP) problem. Chaisiri, Lee and Niyato (2009) employed stochastic integer programming to form an efficient virtual machine placement algorithm. The objective of the proposed algorithm was to minimize the total resource provision costs in cloud environments under different payment plans. However, the proposed algorithm did not address optimizing the energy consumption of the physical machines inside cloud data centers, which can highly influence the costs of the provisioned resources. Speitkamp et al. (2010) presented linear programming formulations to address server consolidation problems. In addition, they designed extension constraints for assigning VMs to a certain set of physical servers that have some unique attribute. They aimed at minimizing both the energy and hardware costs. The proposed work is suitable for data centers with up to 600 physical servers. However, it is not applicable for larger problems with more than 600 servers.

Furthermore, several metaheuristic algorithms have been effectively utilized to address the VMP problem. Evolutionary computing (EC) algorithms such as the genetic algorithm (GA) have been adopted in many works. For example, Adamuthe, Pandharpatte and Thampi (2013) used a non-dominated sorting genetic algorithm-II (NSGA-II) to solve the VMP problem, which was formulated as a multiobjective optimization problem. The proposed work aimed at maximizing both the profits and load balance among the physical servers and minimizing the resource wastage. However, the proposed algorithm was evaluated on a small-sized problem with up to 60 VMs. It is not clear whether this is still effective in a real scenario with hundreds of VMs.

Wang, Liu, Zheng, Sun and Yang (2013) designed an improved GA to solve the VMP problem, aiming at maximizing the resource utilization, balancing the multidimensional resources and minimizing the communication traffic. The results obtained from the conducted simulations show that the proposed approach is better than a number of optimization algorithms regarding the design objectives. However, this work did not address the energy consumption optimization problem, which could have important economic and environmental effects. A modified GA with a fuzzy multiobjective evaluation was presented in Xu and Fortes (2010) to address the VMP problem. The proposed work aimed at minimizing the resource wastage, energy consumption and thermal dissipation costs. The obtained results have shown the superiority of the proposed approach against several bin-packing algorithms and single-objective approaches. However, the proposed work ignored the bandwidth as an important resource in today's world in which a large amount of information has be moved to/from data centers. Foo, Goh, Lim, Zhan and Li (2015) used the GA to optimize a neural network to forecast the energy consumption in cloud data centers. The conducted experiments show that the proposed approach achieves promising results regarding both the forecasting accuracy and speed. However, the proposed work has nothing to do with optimizing the energy consumption and balancing the multidimensional resources of cloud physical machines.

Additionally, Swarm Intelligence (SI) algorithms have been heavily used to solve the VMP problem. Gao et al. (2013) used ant colony optimization (ACO) to minimize the power consumption and resource wastage through efficient VM placement. The proposed algorithm attempted to reduce the number of active servers, which in turn, decreased the amount of consumed energy. The obtained results show that the proposed approach is competitive and even better than other promising approaches to the problem. Liu et al. (2018) used ACO hybridized with order exchange and migration (OEM) local search techniques to reduce the number of active servers needed to host VMs. The results show that the proposed technique outperforms conventional heuristic and other evolutionary based approaches. However, the works presented by Gao et al. (2013) and Liu et al. (2018) both characterized the VMs only in terms of the CPU and memory usage while the bandwidth was ignored. In Wang et al. (2013), energy-aware particle swarm optimization (PSO) was used to solve the VMP problem. The proposed approach formulated the energy consumption of physical servers based on their CPU utilization. The obtained results show that the proposed approach significantly reduced the energy consumption compared to a number of heuristics. However, it did not achieve the balanced usage of multidimensional resources. In addition, the performance of the proposed approach has not been compared to those of other optimization algorithms to assure its superiority to other approaches. Ali and Lee (2014) proposed a virtual machine placement algorithm based on biogeography-based optimization (BBO) (Simon, 2008). The proposed algorithm aimed at reducing the energy consumption in cloud data centers. The obtained results show that the proposed algorithm outperforms the GA regarding energy consumption. However, the proposed work did not attempt to achieve the balanced usage of the multidimensional resources of the physical machines. Additionally, the bandwidth has not been considered in characterizing VMs and physical machines.

In Abdel-Basset et al. (2018), a bandwidth-aware VMP algorithm was proposed based on the improved whale optimization algorithm (WOA) hybridized with a new bandwidth allocation policy (BWAP). The obtained results show that the proposed algorithm outperforms a number of heuristics and meta-heuristics. However, the proposed work only focused on optimizing the bandwidth while other important resources such as the CPU and memory usage have not been considered. Additionally, the power consumption optimization problem was not addressed by their work. Finally, Alresheedi, Lu, Elaziz and Ewees (2019) proposed a hybrid multiobjectives VM placement algorithm based on the salp swarm algorithm (SSA) and sine-cosine algorithm (SCA). The proposed algorithm aimed at optimizing the mean time before host shutdown (MTBHS), the power consumption, and SLA violations. The proposed algorithm has been compared to several meta-heuristics and obtained results supporting its superiority. However, the bandwidth has not been used when describing VMs and physical machines. In addition, the balanced usage of the multidimensional resources in physical servers was not guaranteed.

3. The proposed methodology

3.1. Problem formulation

In this study, the VMP problem is addressed as a variable sized bin packing problem (VSBPP) with non-divisible items Friesen & Langston, 1986). The VSBPP can be summed up as follows: given a set of indivisible items with certain weights and a set of bins with variable sizes (or types), pack all items into a number of bins such that the sum of the costs or the wasted space of the used bins is minimized. In this study, both virtual machines (VMs) and physical machines (PHs) are represented by the three most representative resources: CPU, memory, and network bandwidth. Assume that there are N VMs and M PHs and the total demands of the VMs are less than the total capacities of the PHs. Each VM must be assigned to exactly one physical machine (Eqs. (2) and (4)). Each physical machine must have enough resources for the assigned virtual machines (Eqs. (5), (6), and (7). v_{cpu_i}, v_{mem_i} and v_{BW_i} denote the CPU, memory, and network bandwidth demands of VM_i, respectively. Likewise, p_{cpu_j}, p_{mem_j} and p_{BW_j} denote the CPU, memory, and network bandwidth capacities of PH_j, respectively. The VMP problem can be formulated as a VSBPP as illustrated below.

$$\textbf{\textit{Minimize}} \sum_{j=1}^{M} c_j y_j \tag{1}$$

Subject to

$$x_{ij} = \begin{cases} 1, & if\ server\ PH_j\ is\ allocated\ to\ VM_i \\ 0, & Otherwise \end{cases} \tag{2}$$

$$y_j = \begin{cases} 1, & if\ \sum_{i=1}^{N} x_{ij} \geq 1 \\ 0, & Otherwise \end{cases} \tag{3}$$

Algorithm 1 IGA-POP Algorithm.

Input: Population size(p), permutation size (l), and the maximum number of iterations (t_{max})
Output: The best permutation X_{best}.
Generate p permutations each of size l and save them as population Pop_0.
Initialize r_1 and r_2
For $t = 0$ to $t_{max} - 1$
  Compute the fitness of each permutation X in population Pop_t.
  Select the M best permutations (Pop_t^*).
  For $k = 1$ to p
    Update r_1.
    If ($r_1 < 0.5$) **Then**
    // Mutation
      Mutate X_k by randomly selecting one of the mutation operators and generate the new permutation X_k'.
      Add the mutated permutation X_k' to the new population Pop_{t+1}.
    Else
    //Crossover
      Select a permutation X_r from Pop_t^* using the roulette wheel selection algorithm.
      Apply the crossover operation between X_k and X_r with a copy ratio r_2 to generate X_c.
      Add the fittest permutation among X_k, X_r, and X_c to the new population Pop_{t+1}.
    End
  End
  Update r_2 using Eq. (8).
End
Return the best permutation in the last population, X_{best}.

$$\sum_{j=1}^{M} x_{ij} = 1 \tag{4}$$

$$\forall j, \sum_{i=1}^{N} v_{cpu_i} \times x_{ij} \le p_{cpu_j} \tag{5}$$

$$\forall j, \sum_{i=1}^{N} v_{mem_i} \times x_{ij} \le p_{mem_j} \tag{6}$$

$$\forall j, \sum_{i=1}^{N} v_{BW_i} \times x_{ij} \le p_{BW_j} \tag{7}$$

where y_j is a binary variable that indicates whether PH_j contains virtual machines or not, c_j is the costs or the wasted space of using physical machine PH_j, x_{ij} is a binary variable that indicates if VM_i is assigned to PH_j or not, N is the total number of virtual machines, M is the total number of physical machines, $i \in \{1, 2, \ldots, N\}$, and $j \in \{1, 2, \ldots, M\}$.

3.2. Improved genetic algorithm for permutation-based optimization problems

The effectiveness of the genetic algorithm (GA) has been proved in many optimization problems. However, it suffers from many issues when applied to permutation-based optimization problems. Hence, a modified version of the GA called the Improved Genetic Algorithm for Permutation-based Optimization Problems (IGA-POP) is introduced to improve its effectiveness at solving permutation-based optimization problems. The pseudocode of the IGA-POP algorithm is shown by Algorithm 1. The first step of the IGA-POP algorithm is to create an initial population of feasible solutions (permutations) (Pop_0). For each iteration, the fitness of each solution is computed and the best set of them (Pop_t^*) is selected. Based on the value of a uniformly distributed random variable r_1 (where r_1 is bounded by the interval [0, 1]), each solution is mutated or crossed over with another solution selected from Pop_t^* using the roulette wheel selection algorithm. The role of the variable r_1 is to balance the exploration and the exploitation for the IGA-POP. This process continues until the stopping condition is satisfied and the best solution in the last population (X_{best}) is returned.

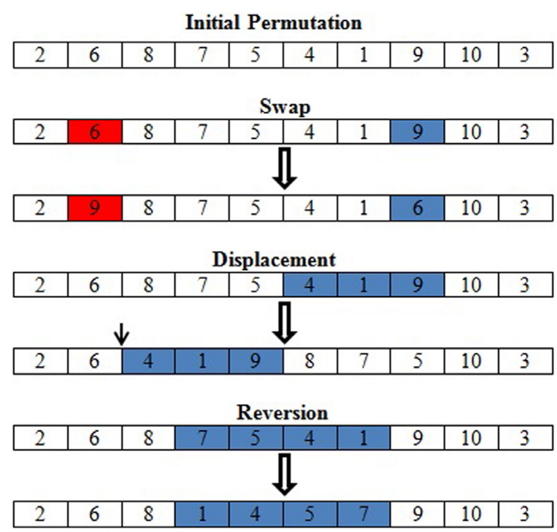

Fig. 1. Mutation operators.

3.2.1. Mutation phase in the IGA-POP algorithm

The mutation operator is applied on the current permutation if the value of r_1 is less than 0.5. The mutation phase in the IGA-POP is performed by randomly selecting one of the following operators: swap, displacement, or reversion (See Fig. 1) (Abdel-Basset et al., 2018). The swap operator randomly selects two positions from the current permutation and swaps their contents. The displacement operator clips a random subset of contiguous positions and inserts it in a random place within the permutation. The reversion operator clips a random subset of contiguous positions and reverses its contents.

3.2.2. Crossover phase in the IGA-POP algorithm

The ordered crossover operator is utilized in the IGA-POP to produce valid permutations. This operator is applied to the current permutation if the value of r_1 is greater than or equal to 0.5 and is performed between the current permutation (X_k) and another permutation (X_r) that is randomly selected from Pop_t^* using the roulette wheel selection algorithm. The result of the crossover operator is a single permutation (X_c) that contains a substring copied from parent X_r of a length determined by the copy ratio r_2, while

Algorithm 2 The proposed VMP algorithm.

Input: Population size(p), the maximum number of iterations (t_{max}), a list of VMs (l_{VM}), and a list of servers(l_{PH}).
Output: The best permutation X_{best}.
Generate p permutations each of size $|l_{VM}|$ and save them as population Pop_0.
Initialize r_1 and r_2
For $t = 0$ to $t_{max} - 1$
  For $k = 1$ to p
    //Virtual machine placement using best strategy
    For each VM_i, $i \in \{1, 2, \ldots, N\}$
      For eachPH_j, $j \in \{1, 2, \ldots, M\}$
      If the available resource of PH_j is suitable for VM_i **Then**
        Compute the resource wastage (W_j) of PH_j after assigning VM_i to PH_j Eq. (11).
      End
    End
    Assign VM_i to PH_j with the minimum resource wastage W_j.
    End
    Compute the fitness of each permutation X in the population Pop_t (Eq. (9)).
  End
  Determine the M best solutions (Pop_t^*) from the population Pop_t.
  For $k = 1$ to p
    Update r_1.
    If ($r_1 < 0.5$) **Then**
    // Mutation
      Mutate $\mathrm{X_k}$ by randomly selecting one of the mutation operators and generate the new permutation $\mathrm{X_k'}$.
      Add the mutated permutation $\mathrm{X_k'}$ to the new population Pop_{t+1}.
    Else
    //Crossover
    Select a permutation $\mathrm{X_r}$ from $\mathrm{Pop_t^*}$ using roulette wheel selection algorithm.
    Apply the crossover operation between $\mathrm{X_k}$ and $\mathrm{X_r}$ with a copy ratio $\mathrm{r_2}$ to generate $\mathrm{X_c}$.
    Add the fittest permutation among $\mathrm{X_k}, \mathrm{X_r}$, and $\mathrm{X_c}$to the new population Pop_{t+1}.
    End
  End
  Update r_2 using Eq. (8)x.
End
Return the permutation with the best allocation from the last population, X_{best}

the remaining parts are copied from parent X_k. The value of the copy ratio r_2 is updated using Eq. (8). The value of r_2 linearly increases from 0.5 to 1 through the iterations of the IGA-POP, i.e., X_c contains the half of the contents of X_r in the first iteration and it is identical to X_r in the last iteration. The proposed crossover operator is designed to balance the exploration and exploitation for the IGA-POP. After applying the crossover operator, the fitness values of X_k, X_r and X_c are computed and the fittest permutation is added to the new population.

$$r_2 = \left[1.5 + 0.5*\left(\frac{2*t - (t_{max} + 1)}{t_{max} - 1}\right)\right] \Big/ 2 \tag{8}$$

where t is the current iteration and t_{max} is the maximum number of iterations.

3.3. Hybrid virtual machine placement algorithm based on the IGA-POP and best fit

In this section, a hybrid VMP algorithm is proposed based on the IGA-POP algorithm and a variation of the best fit (BF) algorithm. The pseudocode of the proposed algorithm is shown in Algorithm 2. Given N virtual machines and M physical machines, the role of the IGA-POP is to suggest different permutations for the VMs that need to be assigned to the available physical machines. The objective of the fitness function of the IGA-POP is to minimize the total power consumption of the used physical machines, which consequently reduces the total costs of the cloud providers. The used fitness function is formulated as follows:

$$\textbf{\textit{Miminize}}\, f(x) = \left[\sum_{j=1}^{M} y_j \times \left(\left(P_j^{busy} - P_j^{idle}\right) \times U_j^{cpu} + P_j^{idle}\right)\right] \tag{9}$$

where $f(x)$ is the total power consumption of the used physical machines, y_j is a binary variable that indicates if PH_j contains virtual machines or not, P_j^{busy} is the maximum power consumption of physical machine PH_j, P_j^{idle} is the minimum power consumption of physical machine PH_j (as suggested in Beloglazov, Abawajy and Buyya (2012), $P_j^{idle} \approx 0.6 * P_j^{busy}$), and U_j^{cpu} is the CPU utilization ratio of physical machine PH_j. U_j^{cpu} is formulated as follows:

$$U_j^{cpu} = \frac{\sum_{i=1}^{N} x_{ij} \times v_{cpu_i}}{p_{cpu_j}} \tag{10}$$

where x_{ij} is a binary variable that indicates if VM_i is assigned to PH_j or not, v_{cpu_i} is the CPU demands of virtual machine VM_i, and p_{cpu_j} is the CPU capacity of physical machine PH_j.

For each permutation produced by the IGA-POP, the role of the BF allocation strategy is to assign the VMs to the available PHs in the submitted order. Given a VM that needs to be assigned, the BF algorithm looks for a PH that can provide the resources (CPU, memory, and network bandwidth) needed by the VM. The BF strategy selects the physical machine with the minimum resource wastage after assigning the current VM. To fully utilize the multi-dimensional resources, the following equation is used to calculate the potential costs of wasted resources:

$$W_j = \frac{\left|2L_j^{cpu} - L_j^{mem} - L_j^{BW}\right| + \varepsilon}{U_j^{cpu} + U_j^{mem} + U_j^{BW}} \tag{11}$$

where W_j is the resource wastage of physical machine PH_j. L_j^{cpu}, L_j^{mem} andL_j^{BW} denote the normalized remaining CPU, memory, and bandwidth of physical machine PH_j, respectively. U_j^{cpu}, U_j^{mem} and U_j^{BW} represent the normalized CPU, memory, and bandwidth usage of physical machine PH_j, respectively. ε is a very small positive real number and its value is set to 0.0001. The main idea of Eq. (11) is to effectively use the multidimensional resources and balance the remaining resources on each physical machine along different dimensions. U_j^{cpu} is previously defined by Eq. (10), while the

Table 1
Average distance and standard deviation for each algorithm.

Algorithms	Datasets					
	bays29	berlin52	burma14	ch130	dantzig42	eil51
IDEA-ICE	3269.1 283.7	15,813.6 797.7	3739.2 166.2	32,330.9 1373.2	947.7 77.4	998.1 65.6
EHBSA_{WT}	2020.0 0.0	7542.0 0.0	3323.0 0.0	6318.7 39.8	699.0 0.0	426.0 0.0
EHBSA_{WO}	2022.5 3.5	7584.6 68.2	3323.0 0.0	8654.9 45.5	701.8 3.4	428.4 0.8
NHBSA_{WO}	2177.8 87.9	9195.8 268.8	3345.2 25.9	9221.4 324.1	854.0 31.6	518.0 17.6
NHBSA_{WT}	2321.8 23.0	11,729.7 217.9	3323.0 0.0	20,787.5225.8	1052.9 19.9	663.5 7.8
$\text{REDA}_{\text{UMDA}}$	4064.9 66.5	21,875.9 504.2	3521.1 102.1	37,791.3 401.1	1750.0 286.2	1234.7 13.3
$\text{REDA}_{\text{MIMIC}}$	3316.1 122.9	15,850.3 757.6	4109.8 76.2	39,310.6 310.0	705.0 6.8	893.4 39.6
OmeGA	3353.5 159.6	18,624.0 671.4	3753.9 4 194.1	42,247.8 428.7	1393.6 53.1	1062.1 34.1
IGA-POP	1880.70 49.93	7519.62 171.37	2680.18 90.44	6657.76 163.69	659.95 13.21	428.4 6.17

remaining terms are defined as follows:

$$L_j^{cpu} = \frac{p_{cpu_j} - \left(\sum_{i=1}^{N} x_{ij} \times v_{cpu_i}\right)}{p_{cpu_j}} \tag{12}$$

$$L_j^{mem} = \frac{p_{mem_j} - \left(\sum_{i=1}^{N} x_{ij} \times v_{mem_i}\right)}{p_{mem_j}} \tag{13}$$

$$L_j^{BW} = \frac{p_{BW_j} - \left(\sum_{i=1}^{N} x_{ij} \times v_{BW_i}\right)}{p_{BW_j}} \tag{14}$$

$$U_j^{mem} = \frac{\sum_{i=1}^{N} x_{ij} \times v_{mem_i}}{p_{mem_j}} \tag{15}$$

$$U_j^{BW} = \frac{\sum_{i=1}^{N} x_{ij} \times v_{BW_i}}{p_{BW_j}} \tag{16}$$

where v_{cpu_i}, v_{mem_i} and v_{BW_i} denote the CPU, memory, and network bandwidth demands of VM_i, respectively. p_{cpu_j}, p_{mem_j} and p_{BW_j} denote the CPU, memory, and network bandwidth capacities of PH_j, respectively. x_{ij} is a binary variable that indicates if VM_i is assigned to PH_j or not.

4. Experimental results

This section aims to evaluate the performance of the proposed algorithms and compare them with other related algorithms. Two experiments have been conducted. The first empirical experiment is conducted to evaluate the proposed IGA-POP algorithm on a set of well-known classical permutation-based problems. The second experiment is conducted to evaluate the proposed hybrid VMP algorithm. Due to the possible statistical fluctuations of the stochastic algorithms, ten runs have been applied for each experiment and the average results are reported. Meanwhile, the deterministic algorithms have been conducted once each.

4.1. Experiment I: IGA-POP evaluation

This experiment is designed to compare the proposed IGA-POP algorithm with related algorithms applied to permutation-based optimization problems. The following set of algorithms are selected for comparison: IDEA-ICE (Bosman & Thierens, 2001), EHBSA-WT (Tsutsui, 2002), EHBSA-WO (Tsutsui, 2002), NHBSA-WO (Tsutsui, Pelikan & Goldberg, 2006), NHBSA-WT (Tsutsui et al., 2006), REDA-UMDA (Romero & Larrañaga, 2009), REDA-MIMIC (Romero & Larrañaga, 2009), and OmeGA (Knjazew & Goldberg, 2002) Two classical permutation-based optimization problems, namely, the Traveling Salesman Problem (TSP) and Flow Shop Scheduling Problem (FSSP), have been used to evaluate the different algorithms. To provide a fair comparison, the same parameter settings are used for the different algorithms. The population size is fixed to 10 × problem size and the number of iterations is fixed to 100 × problem size.

4.1.1. Traveling salesman problem

The TSP is a well-known permutation-based problem that aims to find the shortest route among a set of n cities, where each city should be visited only once. The solution is represented by a permutation sequence $\{c_1, c_2, \ldots, c_n\}$, where $c_i \in [1\ldots n]$. Each sequence represents a possible route. For a problem with n cities, the objective function f is formulated as follows:

$$\textbf{\textit{minimize}}\ f(c_1, c_2, \ldots, c_n) = \sum_{i=1}^{n-1} \mathrm{d}(c_i, c_{i+1}) \tag{17}$$

subject to $c_i \neq c_j, \forall\, \mathrm{i} \neq \mathrm{j}$. where d is the distance between any two given cities. The route with the minimum cost value f is the optimum solution for the TSP problem. The experiment is executed using TSP benchmark test sets (Reinelt, 1991). We selected the TSP datasets with the following labels: bays29, berlin52, burma14, ch130, dantzig42, and eil51, where the number in each dataset label denotes the number of cities. Table 1 shows the average route distance (1st line) and standard deviation (2nd line) for each algorithm.

As shown in Table 1, the proposed IGA-POP algorithm produces comparable results and it outperforms other algorithms in most cases. Moreover, the proposed algorithm has relatively stable performance.

4.1.2. Flow shop scheduling problem

The flow shop scheduling problem (FSSP) is another classical permutation-based optimization problem. Given k jobs and a set of n machines, each job k_i consists of n threads. All jobs arrive at time zero and should be processed by machines in the same given order. Moreover, the dependencies between threads should be maintained. The FSSP aims to schedule all jobs to the given machines while minimizing the completion time (makespan) of all jobs. The solution is represented by a permutation sequence $\{j_1, j_2, \ldots, j_k\}$, where $j_i \in [1\ldots k]$. Each sequence represents the order in which the jobs will be processed by machines. For a problem with k jobs and n machines, the objective function f is formulated as follows:

$$\textbf{\textit{Minimize}}\, f(j_1, j_2, \ldots, j_n) = \mathrm{C}(\mathrm{j_k}, \mathrm{n}) \tag{18}$$

subject to: $j_u \neq j_v, \forall\, \mathrm{u} \neq \mathrm{v}$. where $\mathrm{C(j_k,n)}$ is the completion time of the last job. The permutation with the minimum value f is the optimum solution. The experiment is executed using the FSSP benchmark test sets (Taillard, 1993). We selected the FSSP datasets with the following labels: tai20 × 5_0, tai20 × 5_1, tai20 × 10_0, tai20 × 10_1, tai50 × 10_0, and tai50 × 10_1. For each dataset label "tai a × $\mathrm{b_c}$", a denotes the number of jobs, b denotes the number of machines, and c refers the index of the instance from each file. Table 2 shows the average makespan in seconds (1st line) and standard deviation (2nd line) for each applied algorithm.

Table 2
Average distance and standard deviation for each algorithm.

Algorithms	Dataset					
	tai 20 × 5_0	tai 20 × 5_1	tai 20 × 10_0	tai 20 × 10_0	tai 50 × 10_0	tai 50 × 10_0
IDEA-ICE	1303.1 7.3	1371.2 4.4	1677.8 15.2	1760.5 14.1	3245.2 28.8	3130.4 21.9
$EHBSA_{WT}$	1281.8 4.6	1359.7 0.4	1590.4 4.2	1673.6 4.4	3095.8 9.0	2967.9 9.5
$EHBSA_{WO}$	1296.0 1.2	1365.7 0.7	1606.0 11.8	1710.2 5.6	3265.5 12.3	3124.2 11.8
$NHBSA_{WO}$	1297.0 0.0	1363.2 3.4	1599.5 8.9	1678.3 1.7	3126.0 0.0	3009.8 8.1
$NHBSA_{WT}$	1294.2 4.5	1362.7 3.3	1591.2 2.8	1672.5 5.0	3103.0 8.2	2978.2 5.0
$REDA_{UMDA}$	1297.0 0.0	1375.5 7.0	1675.6 14.5	1764.5 13.3	3336.7 24.4	3194.4 11.9
$REDA_{MIMIC}$	1313.6 6.7	1409.7 6.4	1706.6 10.7	1805.5 13.5	3378.4 18.28	3339.2 12.4
OmeGA	1310.4 9.4	1372.7 6.0	1690.0 26.0	1763.3 24.6	3487.0 23.2	3440.0 0.0
IGA-POP	1278.0 1.4	1359.6 0.5	1599.60 6.0	1684.20 5.9	3083.40 21.4	2964.60 18.5

Table 3
Physical Servers' Configuration.

Type	CPU (GHz)	RAM (GB)	Bandwidth (Bps)	Power (P^{busy})
PH_1	1	1	{10,000,15,000,20,000}	105 W
PH_2	2	3	{15,000,20,000,25,000}	125 W
PH_3	4	4	{20,000,25,000,30,000}	140 W

Table 4
Virtual Machines' Requirements.

Type	CPU (GHz)	RAM (GB)	Bandwidth (Bps)
VM_1	0.5	1	3000:10,000
VM_2	1	2	5000:15,000
VM_3	2	2	5000:20,000

As shown form Table 2, the proposed IGA-POP algorithm outperforms the other comparison methods in most cases and produces results that are too close to the optimal solution in the remaining cases.

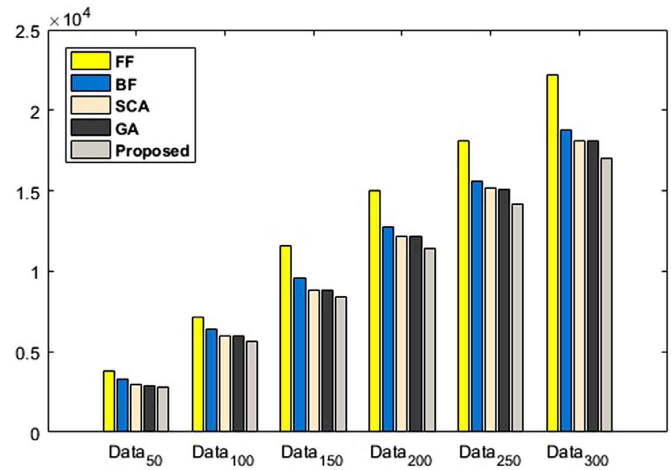

Fig. 2. Power Consumption of the Comparison Algorithms (in Watts).

4.2. Experiment II: evaluating the proposed VMP algorithm

This set of experiments is dedicated to evaluating the proposed hybrid VMP algorithm. For comparison, two classes of algorithms are used: deterministic algorithms such as the First Fit (FT) and Best Fit (BF), and stochastic optimization algorithms such as the Sine-Cosine Optimization Algorithm (SCA) (Mirjalili, 2016) and the Genetic Algorithm (GA). The BF algorithm used in the experiment considers the multidimension resources including the CPU, memory, and network bandwidth. In the GA, the crossover rate, mutation rate, and elitism count are set to 0.8, 0.01, and 2, respectively. In the SCA, the control parameter r_1 linearly decreases from 2 to 0. To conduct a fair comparison among the comparison stochastic algorithms, the population size is fixed to 4 × problem size and the number of iterations is fixed to 200.

Experiments are performed to simulate a data center having heterogeneous physical machines with three different types. The parameter settings for the physical machines are shown by Table 3. Virtual machines are generated based on the assumption that none of the VMs requires more resources than can be offered by a single physical machine.

Similarly, we assumed three different types of VMs with resource requirements, as illustrated in Table 4. Six datasets with different numbers of VMs are randomly generated for the experiments, and the labels of the generated datasets are set to the following: $data_{50}$, $data_{100}$, $data_{150}$, $data_{200}$, $data_{250}$, and $data_{300}$. The number in each dataset label denotes the number of VMs included in the dataset. For each independent experiment, the number of physical machines is set to be equal to the number of virtual machines, which reflects the worst case scenario.

The first experiment is conducted to investigate the optimality of the proposed hybrid VMP algorithm. Small test cases are used in this experiment in order to enumerate all possible solutions to obtain the optimum one. Datasets of sizes 5, 6, 7, 8, and 10 VMs are used. In each scenario, the total number of possible permutations equals n! where n is the number of virtual machines. The performance of the different algorithms is compared to the optimal solution using the power consumption in watt. The obtained results are shown in Table 5.

Based on Table 5, it is noticed that both the proposed hybrid VMP algorithm and the SCA produced the optimum solution using the different test cases while the GA failed to produce the optimum solution in most cases. On the other side, FF and BF algorithms have the worst performance compared to the optimum solution.

Figs. 2 and 3 show the obtained results of the different algorithms using the large datasets. Fig. 2 illustrates the power consumption measured in watts. It is observed that the larger the number of VMs is, the more power that is consumed by the PHs for all applied algorithms. We can also observe that the proposed VMP algorithm produces the minimum power consumption among all algorithms in all cases. Moreover, it is observed that the amount of power saved by the proposed VMP algorithm is noticeable for large datasets. In addition, the FF algorithm is the worst one in terms of the amount of consumed power.

Based on Fig. 3, it is noticed that the proposed hybrid VMP algorithm produces the best placement for the given VMs in terms of the number of used physical machines, which in turn, reduces the amount of consumed power. The good results of the proposed VMP algorithm are mainly obtained due to the effective optimization performed by the IGA-POP algorithm keeping the balance

Table 5
Optimality study for the different algorithms in terms of the power consumption (measured in Watt).

No. of VMs	Algorithms					
	FF	BF	SCA	GA	Proposed	Optimum
5	436.0	364.5	353.5	353.5	353.5	353.5
6	456.0	520.0	352.0	357.0	352.0	352.0
7	579.0	562.0	455.0	455.0	455.0	455.0
8	596.0	667.0	490.0	562.8	490.0	490.0
10	602.5	700.5	588.0	590.8	588.0	588.0

Table 6
Average utilization ratio for each algorithm.

Datasets	Metric	Algorithms				
		FF	BF	SCA	GA	Proposed
$Data_{50}$	CPU	0.80±0	0.76±0	0.77±0.03	0.76±0.02	0.77±0.02
	RAM	0.84±0	0.90±0	0.96±0.03	0.96±0.02	0.99±0.01
	BW	0.73±0	0.78±0	0.86±0.01	0.88±0.08	0.92±0.02
$Data_{100}$	CPU	0.83±0	0.83±0	0.80±0.02	0.79±0.01	0.77±0.02
	RAM	0.88±0	0.97±0	0.98±0.01	0.97±0.003	0.99±0.01
	BW	0.78±0	0.82±0	0.84±0.01	0.84±0.01	0.86±0.01
$Data_{150}$	CPU	0.84±0	0.84±0	0.79±0.01	0.79±0.01	0.77±0.01
	RAM	0.85±0	0.97±0	0.98±0.01	0.98±0.01	0.99±0.002
	BW	0.75±0	0.82±0	0.84±0.01	0.84±0.01	0.88±0.01
$Data_{200}$	CPU	0.83±0	0.85±0	0.81±0.003	0.82±0.01	0.78±0.01
	RAM	0.86±0	0.98±0	0.97±0.003	0.98±0.01	0.99±0.004
	BW	0.78±0	0.85±0	0.86±0.004	0.86±0.004	0.90±0.004
$Data_{250}$	CPU	0.82±0	0.83±0	0.81±0.01	0.82±0.01	0.78±0.01
	RAM	0.87±0	0.98±0	0.98±0.01	0.98±0.01	0.99±0.01
	BW	0.77±0	0.84±0	0.84±0.01	0.85±0.01	0.88±0.01
$Data_{300}$	CPU	0.85±0	0.83±0	0.82±0.01	0.82±0.003	0.78±0.01
	RAM	0.88±0	0.98±0	0.98±0.01	0.98±0.003	0.99±0.001
	BW	0.77±0	0.82±0	0.84±0.01	0.84±0.01	0.87±0.001

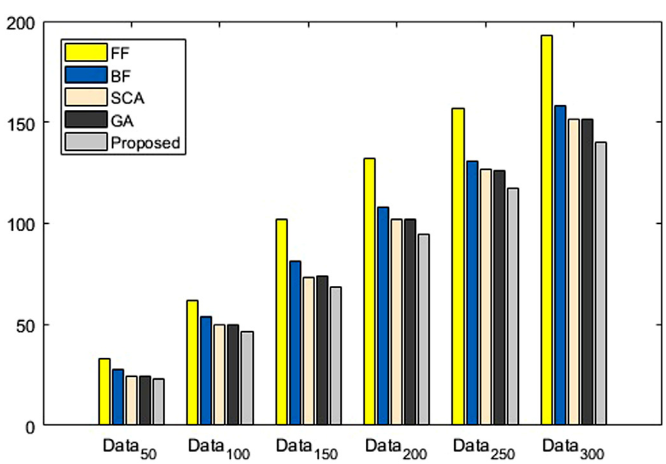

Fig. 3. Number of Physical Machines for the Comparison Algorithms.

between the exploration and exploitation. Another explanation for these good results is balancing the utilization rate for the different resources within the physical machines using the multidimension resource-aware best fit allocation strategy. Furthermore, the FF strategy is also the worst among the different comparison methods.

To evaluate the convergence of the proposed VMP algorithm compared to other stochastic algorithms, another experiment is conducted. Fig. 4 shows the convergence curves for the GA, SCA, and the proposed VMP algorithm for the "$data_{50}$" instance. Based on Fig. 4, it is noticed that the proposed VMP algorithm has the fastest convergence compared to the other algorithms. Moreover, it produces the minimum power consumption compared to other algorithms.

Resource utilization is another key objective for virtual machines' placement. Table 6 shows the average normalized utilization rates with respect to the CPU, RAM, and Bandwidth (BW) of the different algorithms. The CPU utilization rate, the memory utilization rate, and bandwidth utilization rate are calculated using Eqs. (10), (15), and (16), respectively.

Based on Table 6, we can observe that the proposed VMP algorithm produces an efficient utilization with respect to the different resources. The applied resource-aware best fit (BF) allocation strategy assigns the VMs to the available physical machines while considering the efficient and balanced utilization of the different resources. Additionally, it is noticed that some algorithms achieve better utilization with respect to the CPU usage while the proposed VMP algorithm achieves the balanced utilization of the different resources.

The experimental results show that the efficient utilization of the multidimensional resources of active servers can play an important role in improving the power consumption of cloud data centers. Unbalanced residual resources along different resource dimensions would prevent active servers from hosting more VMs. This situation requires activating more servers to accommodate upcoming VM placement requests, which in turn, consume more energy.

Finally, the elapsed run time of each algorithm is measured and the obtained results are shown in Table 7. All experiments have been conducted using an Intel (R) Core (TM) i7 (2.27) GHz machine with 4.00GB RAM and the Windows 8 operating system.

Based on Table 7, it is observed that increasing the number of VMs increases the run time for all algorithms. It is observed that the SCA, GA, and the proposed VMP algorithm take significantly greater time compared to FF and BF. These results are rational where the SCA, GA, and the proposed VMP algorithm are iterative algorithms, while FF and BF are non-iterative algorithms.

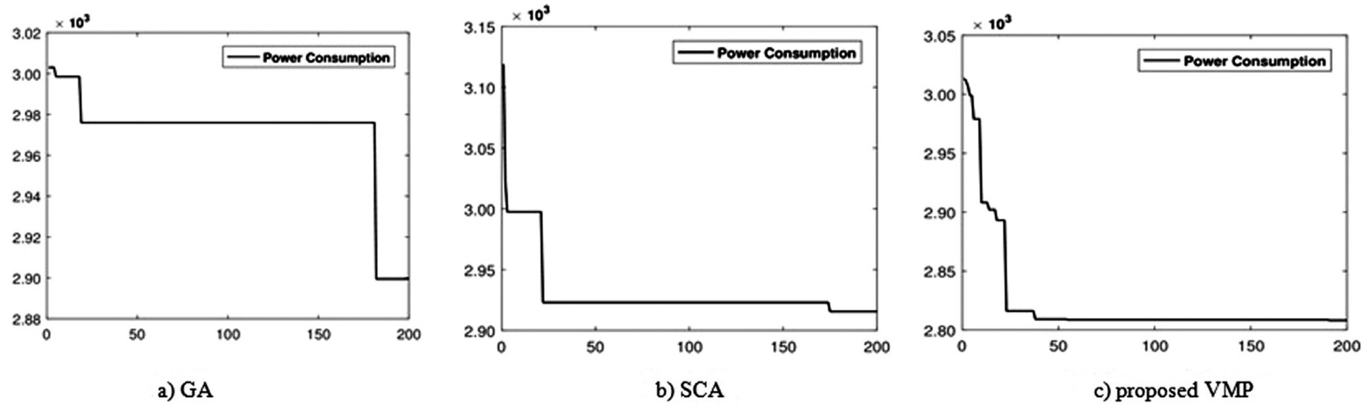

Fig. 4. Convergence of the different algorithms for the "$data_{50}$" instance.

Table 7
Elapsed run time for each algorithm (in seconds).

Dataset	Algorithms				
	FF	BF	SCA	GA	Proposed
$Data_{50}$	0.005	0.003	67.01	78.60	107.25
$Data_{100}$	0.005	0.005	292.55	327.77	433.54
$Data_{150}$	0.009	0.010	717.14	755.15	1101.49
$Data_{200}$	0.012	0.013	1311.64	1383.56	2077.50
$Data_{250}$	0.017	0.014	2237.92	2267.36	2788.09
$Data_{300}$	0.023	0.020	3283.52	3226.25	3527.73

Moreover, the SCA takes less time compared to both the GA and the proposed VMP algorithm.

5. Conclusion

Efficient VM placement in PMs can effectively reduce the energy consumption of cloud data centers. In this study, a hybrid VMP algorithm is mainly presented to improve the energy consumption of cloud data centers through minimizing the number of active servers. The proposed VMP algorithm combines an improved permutation-based GA (IGA-POP) and resource-aware best fit strategy. The algorithm reduced the number of active servers via an efficient balanced usage of the multidimensional resources of active servers, which allows them accommodate future VMs placement requests without the need to activate other servers. The performance of the proposed IGA-POP has been evaluated against several permutation-based algorithms on two well-known problems (TSP and FSSP). The obtained results using a number of datasets showed that the IGA-POP is better than other permutation-based algorithms in most cases and achieved competitive performance in the remaining cases. In addition, the performance of the proposed VMP algorithm has been evaluated against a number of heuristics and metaheuristics under different scenarios. The experimental results showed that the proposed VMP algorithm is better than other algorithms in terms of the energy savings and number of active servers. Moreover, the proposed VMP algorithm achieved better utilization of the multidimensional resources (CPU, RAM, and Bandwidth) of the active servers through balancing the residual resources along the different dimensions.

For future work, the proposed hybrid VMP algorithm can be used to perform cloud task scheduling to achieve a number of objectives such as minimizing the makespan, maximizing the load balance among the different VMs, and maximizing the resource utilization of the used VMs. Additionally, the proposed VMP algorithm can be adapted to consider constrained optimization such as allocating VMs to a specific set of physical servers that contain some unique attributes. Additionally, violations of the SLA may happen due to the high CPU utilization or high memory utilization of physical servers. Hence, the proposed VMP algorithm may be adapted to avoid the overutilization that causes VM performance degradation through adopting a live VM migration strategy. Moreover, using a VM migration strategy can allow the VMP algorithm to cope with the dynamic requirements of users and applications. In addition, the proposed algorithm can applied to emerging real-world applications such as production scheduling, home health care systems, and health care facility planning.

Declaration of Competing Interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Credit authorship contribution statement

A.S. Abohamama: Conceptualization, Methodology, Data curation, Writing - original draft, Writing - review & editing, Investigation, Project administration. **Eslam Hamouda:** Conceptualization, Methodology, Software, Validation, Formal analysis, Writing - original draft, Investigation, Resources.

References

Abdel-Basset, M., Abdle-Fatah, L., & Sangaiah, A. K. (2018). *An improved lévy based whale optimization algorithm for bandwidth-efficient virtual machine placement in cloud computing environment*. Cluster Computing. doi:10.1007/s10586-018-1769-z.

Abohamama, A. S., Alrahmawy, M. F., & Elsoud, M. A. (2018). Improving the dependability of cloud environment for hosting real time applications. *Ain Shams Engineering Journal, 9*(4), 3335–3346.

Abohamama, A. S., Alrahmawy, M. F., Elsoud, M. A., & Hamza, T. T. (2018). Swarm intelligence based fault-tolerant real-time cloud scheduler. *Mansoura Journal for Computers and Information Sciences, 14*(1), 1–10.

Adamuthe, A. C., Pandharpatte, R. M., & Thampi, G. T. (2013, November). Multiobjective virtual machine placement in cloud environment. In *2013 International Conference on Cloud & Ubiquitous Computing & Emerging Technologies* (pp. 8–13). IEEE.

Alharbi, F., Tian, Y. C., Tang, M., Zhang, W. Z., Peng, C., & Fei, M. (2019). An ant colony system for energy-efficient dynamic virtual machine placement in data centers. *Expert Systems with Applications, 120*, 228–238.

Ali, H. M., & Lee, D. C. (2014, December). A biogeography-based optimization algorithm for energy efficient virtual machine placement. In *2014 IEEE Symposium on Swarm Intelligence* (pp. 1–6). IEEE.

Alicherry, M., & Lakshman, T. V. (2013,). Optimizing data access latencies in cloud systems by intelligent virtual machine placement. In *2013 Proceedings IEEE INFOCOM* (pp. 647–655). IEEE.

Alresheedi, S. S., Lu, S., Elaziz, M. A., & Ewees, A. A. (2019). Improved multiobjective salp swarm optimization for virtual machine placement in cloud computing. *Human-centric Computing and Information Sciences, 9*(1), 15. doi:10.1186/s13673-019-0174-9.

Beloglazov, A., Abawajy, J., & Buyya, R. (2012). Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future generation computer systems, 28*(5), 755–768.

Bosman, P. A., & Thierens, D. (2001, July). Crossing the road to efficient IDEAs for permutation problems. In *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation* (pp. 219–226). Morgan Kaufmann Publishers Inc.

Chaisiri, S., Lee, B. S., & Niyato, D. (2009, December). Optimal virtual machine placement across multiple cloud providers. In *2009 IEEE Asia-Pacific Services Computing Conference (APSCC)* (pp. 103–110). IEEE.

Chaisiri, S., Lee, B. S., & Niyato, D. (2009, December). Optimal virtual machine placement across multiple cloud providers. In *2009 IEEE Asia-Pacific Services Computing Conference (APSCC)* (pp. 103–110). IEEE.

Dang, H. T., & Hermenier, F. (2013, November). Higher sla satisfaction in datacenters with continuous vm placement constraints. In *Proceedings of the 9th Workshop on Hot Topics in Dependable Systems* (p. 1). ACM.

Foo, Y. W., Goh, C., Lim, H. C., Zhan, Z. H., & Li, Y. (2015, October). Evolutionary neural network based energy consumption forecast for cloud computing. In *2015 International Conference on Cloud Computing Research and Innovation (ICCCRI)* (pp. 53–64). IEEE.

Friesen, D. K., & Langston, M. A. (1986). Variable sized bin packing. *SIAM journal on computing, 15*(1), 222–230.

Gao, Y., Guan, H., Qi, Z., Hou, Y., & Liu, L. (2013). A multi-objective ant colony system algorithm for virtual machine placement in cloud computing. *Journal of Computer and System Sciences, 79*(8), 1230–1242.

Kanet, J. J., & Adelsberger, H. H. (1987). Expert systems in production scheduling. *European Journal of Operational Research, 29*(1), 51–59.

Knjazew, D., & Goldberg, D. E. (2000, July). OMEGA-Ordering messy GA: Solving permutation problems with the fast messy genetic algorithm and random keys. In Proceedings of Genetic and Evolutionary Computation Conference (pp. 181–188).

Liu, D., Sui, X., & Li, L. (2016, August). An energy-efficient virtual machine placement algorithm in cloud data center. In *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)* (pp. 719–723). IEEE.

Liu, R., Xie, X., Augusto, V., & Rodriguez, C. (2013). Heuristic algorithms for a vehicle routing problem with simultaneous delivery and pickup and time windows in home health care. *European Journal of Operational Research, 230*(3), 475–486.

Liu, X. F., Zhan, Z. H., Deng, J. D., Li, Y., Gu, T., & Zhang, J. (2018). An energy efficient ant colony system for virtual machine placement in cloud computing. *IEEE Transactions on Evolutionary Computation, 22*(1), 113–128.

López-Santana, E. R., & Méndez-Giraldo, G. A. (2016, September). A knowledge-based expert system for scheduling in services systems. In *Workshop on Engineering Applications* (pp. 212–224). Springer.

Mirjalili, S. (2016). SCA: A sine cosine algorithm for solving optimization problems. Knowledge-Based Systems*, 96*, 120–133.

Omisore, M. O., Samuel, O. W., & Atajeromavwo, E. J. (2017). A genetic-neuro-fuzzy inferential model for diagnosis of tuberculosis. *Applied Computing and Informatics, 13*(1), 27–37.

Reinelt, G. (1991). TSPLIB—A traveling salesman problem library. *ORSA journal on computing, 3*(4), 376–384.

Romero, T., & Larrañaga, P. (2009). Triangulation of bayesian networks with recursive estimation of distribution algorithms. *International Journal of Approximate Reasoning, 50*(3), 472–484.

Sauve, B., & Collinot, A. (1987). An expert system for scheduling in a flexible manufacturing system. *Robotics and Computer-Integrated Manufacturing, 3*(2), 229–233.

Shariff, S. R., Moin, N. H., & Omar, M. (2012). Location allocation modeling for healthcare facility planning in malaysia. *Computers & Industrial Engineering, 62*(4), 1000–1010.

Simon, D. (2008). Biogeography-based optimization. *IEEE transactions on evolutionary computation 12*, 702–713.

Speitkamp, B., & Bichler, M. (2010). A mathematical programming approach for server consolidation problems in virtualized data centers. *IEEE Transactions on services computing, 3*(4), 266–278.

Taillard, E. (1993). Benchmarks for basic scheduling problems. *european journal of operational research, 64*(2), 278–285.

Tang, M., & Pan, S. (2015). A hybrid genetic algorithm for the energy-efficient virtual machine placement problem in data centers. *Neural Processing Letters, 41*(2), 211–221.

Tsutsui, S. (2002, September). Probabilistic model-building genetic algorithms in permutation representation domain using edge histogram. In *International Conference on Parallel Problem Solving from Nature* (pp. 224–233). Springer.

Tsutsui, S., Pelikan, M., & Goldberg, D. E. (2006). Node histogram vs. edge histogram: A comparison of pmbgas in permutation domains. In *MEDAL Report* (p. 2006009).

Wang, S., Gu, H., & Wu, G. (2013, July). A new approach to multi-objective virtual machine placement in virtualized data center. In *2013 IEEE Eighth International Conference on Networking, Architecture and Storage* (pp. 331–335). IEEE.

Wang, S., Liu, Z., Zheng, Z., Sun, Q., & Yang, F. (2013, December). Particle swarm optimization for energy-aware virtual machine placement optimization in virtualized data centers. In *2013 International Conference on Parallel and Distributed Systems* (pp. 102–109). IEEE.

Xu, J., & Fortes, J. A. (2010, December). Multi-objective virtual machine placement in virtualized data center environments. In *2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing* (pp. 179–188). IEEE.

Xu, Y., Wang, L., Wang, S., & Liu, M. (2014). An effective hybrid immune algorithm for solving the distributed permutation flow-shop scheduling problem. *Engineering Optimization, 46*(9), 1269–1283.

Yan, J., Zhang, H., Xu, H., & Zhang, Z. (2018). Discrete PSO-based workload optimization in virtual machine placement. *Personal and Ubiquitous Computing, 22*(3), 589–596.

Zheng, Q., Li, R., Li, X., Shah, N., Zhang, J., & Tian, F. , Chao, K. M. , & (2016). Virtual machine consolidated placement based on multi-objective biogeography-based optimization. *Future Generation Computer Systems, 54*, 95–122.