Decision Science Letters 13 (2024) ***–***

Contents lists available at GrowingScience

Decision Science Letters

homepage: www.GrowingScience.com/dsl

Simple and efficient duelist algorithm variations for energy-aware virtual machine placement in cloud data centers

Amol Adamuthe[a*] and Vrushabh D. Kupwade[a]

[a]*Department of Information Technology, Rajarambapu Institute of Technology, Shivaji University, Sakhrale, MS – 415414, India*

C H R O N I C L E

Article history:
Received: August 17, 2023
Received in revised format: November 20, 2023
Accepted: March 28, 2024
Available online: March 28, 2024

Keywords:
Energy efficiency in cloud
Virtual machine placement
Duelist algorithm

A B S T R A C T

This research presents a novel approach to address the Virtual Machine Placement Problem (VMPP) in cloud data centers with the aim of minimizing energy consumption. The main contributions of this study are threefold. Firstly, a Duelist Algorithm specifically designed for VMPP, which introduces a unique concept of duelists combined with optimization techniques. The algorithm aims to strike a balance between exploration and exploitation in the search space, leading to more effective resource allocation and energy-efficient cloud data center management. Secondly, enhance the performance of the Duelist Algorithm by reducing the number of algorithm-specific parameters. This simplifies the implementation process and increases the algorithm's adaptability to various real-world problems, making it more user-friendly and robust. Lastly, conduct a comprehensive comparison of the Duelist Algorithm with the widely used Hybrid Harmony Search Algorithm (HS+SA+LS) in terms of energy consumption and overall efficiency. The experimental results demonstrate that the Duelist Algorithm consistently outperforms the Hybrid Harmony Search Algorithm, achieving remarkable improvements in both best and mean fitness values. Additionally, the Duelist Algorithm exhibits lower standard deviation values, indicating more stable and consistent performance. The findings of this research validate the effectiveness of the proposed Duelist Algorithm in minimizing energy consumption and optimizing cloud resource allocation. The reduction of algorithm-specific parameters further contributes to its versatility and simplicity.

1. Introduction

Cloud computing enables the immediate provision of computing resources on demand, encompassing software, data storage, computing power, networking, and databases. This service model includes different types of cloud services, such as Infrastructure as a Service (IaaS), Software as a Service (SaaS), and Platform as a Service (PaaS) (Buyya et al., 2010). Both data center providers and end-users benefit from cloud computing. Providers can offer computer resources to a large number of consumers, while users can access these resources at lower costs compared to maintaining private infrastructure. The primary goal of cloud providers is to maximize profits, which involves minimizing the deployment of computer resources. Data centers constitute the fundamental infrastructure of cloud computing, housing IT equipment for data storage, processing, and communication. To meet user demands, data centers operate continuously with numerous active hosts or servers, networking equipment, and storage devices, resulting in significant energy consumption (Shuja et al., 2016). Extensive research has been undertaken to address the issue of data center power consumption, with collaborative studies involving the United States. In 2010, data center energy consumption stood at 91 billion kWh, with a projected increase to 140 billion kWh by 2020 (Alharbi et al., 2019). Currently, data centers account for approximately 1.1-1.3 percent of the total global energy consumption, a proportion expected to rise to 8% by 2020. The rapid growth in energy consumption by data centers raises significant economic and environmental concerns, as data center servers contribute to 0.5 percent of global CO2 emissions. Consequently, there is a growing focus on research aimed at reducing data center power usage and

* Corresponding author.
E-mail address: amol.admuthe@gmail.com (A. Adamuthe)

doi: 10.5267/dsl.2024.3.004

addressing challenges related to the optimal placement of virtual machines within cloud infrastructure.

The major IT infrastructure for cloud computing is the data center, comprising IT equipment for data processing, storage, and communication. Data centers continuously operate with a multitude of active hosts, servers, networking equipment, and storage devices, resulting in significant energy usage (Shuja et al., 2016). Currently, data centers consume approximately 3% of the world's electricity, with predictions indicating a rise to 4% by 2030. Hyperscale operations consume 20 to 50 MW of power annually, equivalent to powering 37,000 households. The increasing energy consumption in data centers has led to more scrutiny from governmental authorities (Energy Efficiency Predictions for Data Centres in 2023). Since 2015, the electricity used in data centers in Ireland has more than tripled and accounted for 14% of all electricity used by 2021. The data center industry overview indicates that 10% of global IT organizations will go server-less before 2023, and data centers are projected to consume 20% of the total energy by 2025, with a significant spending of $222 billion on IT data centers in 2023 (15 Crucial Data Center Statistics to Know in 2023). According to the International Energy Agency (IEA), data center workload significantly increased by 260% from 2015 to 2021, with energy usage rising by 10-60% during the same period (IEA, 2022). The energy consumption in cloud data centers remains a critical environmental concern, as the operation of data center servers contributes to 0.5% of the world's $CO2$ emissions.

As a consequence, there has been a notable increase in research focused on reducing power consumption within data centers and addressing the challenge of Virtual Machine Placement (VMP) within cloud infrastructure. Various strategies to curtail energy consumption in data centers have been explored by researchers, with one particularly effective approach involving the implementation of virtualization technology for server or physical machine (PM) management. Virtual machine placement, alternatively known as server consolidation, encompasses the mapping of multiple virtual machines (VMs) onto physical machines (PMs), enabling the sharing of resources such as CPU, storage, bandwidth, and memory. This tactic serves to optimize resource utilization and assumes a pivotal role in data center management (Speitkamp & Bichler, 2010). However, despite the evident advantages of virtual machine placement, substantial challenges remain to be addressed. Of primary concern is the task of determining the most optimal allocation or arrangement of VMs onto physical machines, a task guided by specific design objectives geared toward minimizing energy consumption within data centers. The achievement of this optimization presents a critical research quandary that necessitates thorough consideration and continued exploration within the research community. The principal aim of virtual machine placement algorithms is to secure an optimal distribution of VMs across PMs while concurrently satisfying designated design objectives (Speitkamp & Bichler, 2010). The initial concept entailed the mapping of VMs onto a select number of energy-efficient active servers, thereby allowing inactive or underutilized hosts to be deactivated (Tang & Pan, 2015). Such an approach can yield energy savings of up to 66% of total consumption (Chen et al., 2008). Given the intricate nature of cloud workloads, characterized by their dynamic and unpredictable attributes, coupled with the numerous constraints inherent in VM placement on physical hosts, the VMP problem stands as a complex optimization challenge within cloud computing. The fundamental objective revolves around efficient management of physical machine utilization, aimed at reducing the overall count of active PMs within data centers (Speitkamp & Bichler, 2010).

Usmani & Singh (2016 conducted a comprehensive survey on the VMP problem, providing valuable insights into different approaches employed by researchers to tackle the issue. One essential classification is based on resource types, with several studies primarily focusing on the criticality of CPU resources for physical machines (Batista et al., 2007; Breitgand & Epstein, 2011). However, the problem is extended in some studies to consider additional resources like memory and bandwidth (Van et al., 2010; Zhu et al., 2017). Another classification is based on the considered set of virtual machines, with some studies addressing the placement of all virtual machines in the data center simultaneously (Beloglazov et al., 2012; Biran et al., 2012), while others concentrate on a single virtual machine or a set of VMs belonging to the same application (Breitgand & Epstein, 2011; Jayasinghe et al., 2011). The objectives pursued in VMP studies also vary, with the majority of works focusing on optimizing energy consumption (Ghribi et al., 2013; Beloglazov et al., 2012), but employing different energy models. Some studies consider the number of active physical machines as the primary factor influencing energy consumption (Beloglazov & Buyya, 2010), while others aim to reduce overloaded physical equipment to mitigate performance loss. Several projects also incorporate the cost of virtual machine migration into their research (Wood et al., 2009).

Researchers have proposed various methods to address the VMP problem effectively. A comprehensive review of virtual machine placement methods in a cloud environment using metaheuristic algorithms was provided by (Alsadie, 2022). The review discussed the advantages and disadvantages of various methods, emphasizing the need for further research in this area. Deterministic algorithms have been explored, such as those introduced by Chaisiri et al. (2009), Alicherry & Lakshman (2013), and Dang & Hermenier (2013), which aim to find optimal solutions based on predefined criteria. Additionally, more intelligent metaheuristic algorithms, like those proposed by Gao et al. (2013) and Abdel-Basset et al. (2019), have gained attention in optimizing VM placement while considering multiple objectives and resource constraints. In the pursuit of power efficiency challenges, the Virtual Machine Placement Framework towards the Power Efficiency of Sustainable Cloud Environment (MV-PESC) technique is recommended. Furthermore, the FPNSO algorithm, introduced by (Singh et al., 2023), has shown significant improvements in power consumption, carbon emissions, and resource utilization in cloud data centers, using a combination of Flower Pollination Optimization (FPO) and NSGA-II. To enhance the energy efficiency of

VMP in cloud data centers, (Wu et al., 2012) presented a new approach that reduced the computational complexity of the Genetic Algorithm (GA) by employing a new data structure and an alternative fitness function, leading to faster execution times. Moreover, the importance of efficient virtual machine placement in cloud computing, with the aim of reducing energy consumption and increasing resource usage, was discussed in (Pushpa & Siddappa, 2022). The authors introduced an enhanced Variation of the ABCSO algorithm that integrated the capabilities of Artificial Bee Colony (ABC) and Cat Swarm Optimization (CSO) for improved results. Additionally, the Ant Colony Optimization (ACO) algorithm proposed by (Xing et al., 2022) takes traffic-awareness into account to enhance overall system performance and improve energy efficiency, offering a promising approach for effective resource allocation and management in cloud computing.

The Duelist algorithm is an evolutionary algorithm that operates based on population dynamics, where each individual in the population is referred to as a duelist. In this algorithm, duelist individuals engage in fights with each other to determine winners and losers. Both winners and losers adopt different strategies for self-improvement. Winners learn from their past mistakes, using them as opportunities for growth, while losers learn from the winners, attempting to incorporate some of their successful traits. Through numerous improvements and duels, certain duelists emerge as the best solutions to the given set of challenges (Biyanto et al., 2016a; Biyanto et al., 2015).

The research gap identified in the context of the proposed modifications in the Duelist Algorithm is as follows.

- Balancing Exploitation and Exploration: One of the key challenges in optimization algorithms is finding the right balance between exploitation and exploration. Exploitation involves exploiting the current best solutions to converge towards a local optimum, while exploration involves exploring new regions of the search space to potentially find a global optimum. The research gap in this context is to develop modifications in the Duelist Algorithm that can effectively balance exploitation and exploration, allowing the algorithm to efficiently explore the search space while converging towards optimal solutions.
- Improving Energy Efficiency of Cloud Data Centers: Cloud data centers are critical infrastructures that consume substantial amounts of energy. Optimizing energy efficiency in cloud data centers is a significant concern, as it can lead to cost savings and environmental benefits. The research gap here lies in exploring how the proposed modifications in the Duelist Algorithm can be tailored and extended to address the specific challenges of optimizing energy efficiency in cloud data centers. By incorporating energy-related constraints and objectives into the algorithm, it may be possible to design a more efficient and sustainable cloud resource allocation approach.
- Reducing Algorithm-Specific Parameters: Optimization algorithms often require the tuning of various parameters to achieve optimal performance for specific problem domains. However, having a large number of algorithm-specific parameters can make the algorithm complex to implement and tune. The research gap is to investigate methods to reduce the number of algorithm-specific parameters in the modified Duelist Algorithm while maintaining or even improving its performance. By developing a more parameter-efficient Variation, the algorithm can be more easily applied to various real-world problems without extensive parameter tuning.

The main contributions of the research can be summarized as follows.

- Designing a Duelist Algorithm for VMPP: The primary contribution of this research is the design and development of a novel Duelist Algorithm specifically tailored to address the Virtual Machine Placement Problem (VMPP) in cloud data centers. The VMPP is a crucial optimization problem that aims to minimize energy consumption while efficiently allocating virtual machines to physical machines. By devising a Duelist Algorithm that is well-suited for this specific problem domain, the research contributes to the advancement of optimization techniques in cloud computing.
- Reducing Algorithm-Specific Parameters: Another significant contribution is the reduction of algorithm-specific parameters in the proposed Duelist Algorithm. Many optimization algorithms require the tuning of numerous parameters to achieve optimal performance, making them complex and time-consuming to implement and calibrate. By reducing the number of algorithm-specific parameters in the Duelist Algorithm, the research simplifies its implementation and makes it more accessible to practitioners and researchers alike. This reduction in parameters enhances the algorithm's efficiency and applicability to a wider range of real-world problems.
- Performance Comparison with Hybrid Harmony Search Algorithm: The research further contributes by conducting a comprehensive performance comparison between the proposed Duelist Algorithm and the Hybrid Harmony Search Algorithm (HS+SA+LS), which is a well-known and widely used optimization technique. By comparing the performance of both algorithms on the VMPP, the research provides valuable insights into the strengths and weaknesses of each approach. This comparison helps identify the relative advantages of the Duelist Algorithm over the Hybrid Harmony Search Algorithm in terms of energy consumption optimization and overall efficiency.

Section 2 presents a virtual machine placement problem with objectives, assumptions and constraints. Duelist algorithm, proposed variations, solution representation and fitness function is presented in section 3. Section 4 is about experimental details, results and discussion. Section 5 is about the conclusion.

2. Virtual Machine Placement Problem

The Virtual Machine Placement (VMP) problem is a crucial challenge in the field of cloud computing, as it directly impacts the overall efficiency and performance of cloud environments. In cloud computing, a data center hosts a large number of physical machines (PMs), and on these PMs, multiple Virtual Machines (VMs) are mounted. The efficient allocation of these VMs onto the available PMs is the key objective of the VMP problem. The primary goal of the VMP problem is to achieve optimal resource utilization while minimizing operational costs and meeting the performance requirements of the applications running on the VMs. This involves finding a suitable mapping of VMs to PMs that maximizes resource utilization, minimizes the number of active physical servers, and balances the workload across the data center.

In this particular research study, the primary focus is on reducing energy consumption in cloud data centers by optimizing the allocation of CPU resources. The VMP problem is addressed using a bin packing problem formulation, which is a well-known optimization problem. In the context of VMP, the bin packing problem involves assigning VMs (items) to PMs (bins) in a way that minimizes the number of PMs used while ensuring that the resource requirements of each VM are met by its corresponding PM.

The study formulates the VMP problem as an energy optimization problem, with the aim of finding a configuration that minimizes the total energy consumption of the data center. The constraints of the problem ensure that each VM is assigned to a single PM, and each PM has sufficient resources to accommodate its assigned VM. In this particular study, the primary objective is to reduce energy consumption in cloud data centers, with a specific focus on considering CPU as a promising resource for optimization. The VMP problem is addressed using a bin packing problem formulation, which has been previously described in a relevant work (Abohamama & Hamouda, 2020). The formulation involves assigning a set of VMs to a set of Physical Machines (PMs) in a cloud data center while adhering to certain constraints. The main goal is to achieve a reduction in the total energy usage of the data center.

The formulation of the VMP problem entails the following constraints:

- All virtual machines must be allocated to a physical machine, ensuring that each VM has a dedicated PM.
- Only one physical machine should be allocated to each virtual machine, ensuring a one-to-one VM-to-PM mapping.
- Physical machines must possess sufficient resources to accommodate the assigned VMs, ensuring that the resource requirements of each VM are met by its respective PM.

The given equations describe a mathematical model for minimizing the total energy consumption of a data center due to physical machines (PMs) hosting virtual machines (VMs). The objective function f(p) represents the total energy consumed by the data center. The model takes into account the energy consumption of each PM based on its CPU utilization and whether or not it contains a VM. The fitness function is given as follows,

$$min\ f(p) = \sum_{j=1}^{Q} \alpha_j \times \left\{ \left(P\_max_j^{busy} - P\_min_j^{idle} \right) \times P_{ut}{}_j^{cpu} + P\_min_j^{idle} \right\} \tag{1}$$

$$P_{ut}{}_j^{cpu} = \sum_{i=1}^{V} \beta_{ij} \times \frac{CPU\_V_i}{CPU\_P_j} \tag{2}$$

where,

i € {1,2,3,……,V} and j € {1,2,3,……,Q}
$f(p)$ is total energy consumed by data center due to PMs
α_j show whether or not j[th] physical machine contains virtual machine
$P\_max_j^{busy}$ maximum energy consumption of PM
$P\_min_j^{idle}$ minimum energy consumption of PM
$P_{ut}{}_j^{cpu}$ is CPU utilization ratio of j[th] physical machine
β_{ij} is a binary that shows whether or not a virtual machine is assigned to a physical machine
CPU_V_i is virtual machine's CPU demand
CPU_P_j is physical machine's CPU capacity

3. Methodology for Solving VMP Problem

This section presents the proposed solution representation, initialization and fitness function for virtual machine placement problem.

3.1 Duelist Algorithm

The Duelist algorithm offers a novel approach that draws inspiration from human combative behaviour and learning capabilities to tackle these optimization problems effectively. The iterative nature of the Duelist algorithm allows it to refine the pool of duelists through competitive learning and adaptation, making it a powerful optimization tool for various computer engineering tasks. The flowchart of the fundamental Duelist algorithm is illustrated in Fig. 1.

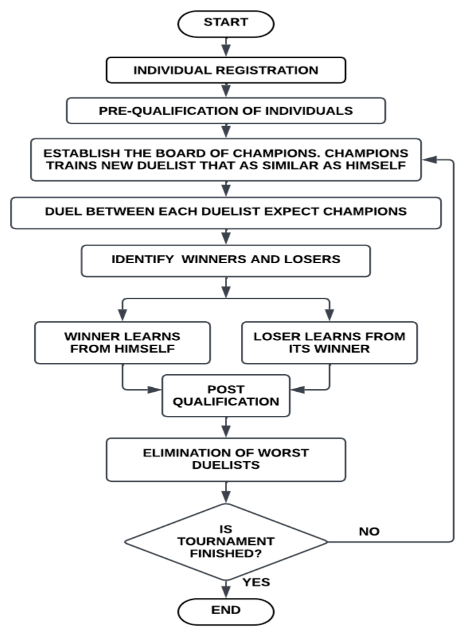

Fig. 1. Flowchart of basic Duelist algorithm (Biyanto et al., 2016a)

The Duelist algorithm has been successfully applied to various real-world problems across different domains. Some examples include solving the refinery crude preheat train cleaning scheduling (Biyanto et al., 2016b), optimizing oil production in CO_2 enhanced oil recovery and steam injection in enhanced oil recovery (Biyanto et al., 2017a), enhancing the dimensional surface quality and material removal rate in turning processes (LASHIN et al., 2020) and optimizing energy efficiency and conservation in green building design (Biyanto et al., 2017b).

Steps of the algorithm are,

- Pre-Qualification of Duelists: Before entering the dueling phase, every registered duelist undergoes a pre-qualification test to assess their skillset, which is calculated using the fitness function. The fitness function evaluates how well each duelist's assigned VM-to-PM assignment satisfies the problem's constraints and objectives. It is crucial for every duelist to meet the specific problem's requirements to be considered for further competition. The fitness of each duelist is determined based on Equation 1, which quantifies their performance in the given problem context.
- Establishing the Champion's Board: The board of champions is formed based on the optimal fitness value achieved by the duelists. For this study, the duelists with the minimum fitness values are selected to be part of the champion's board. A total of five champions are chosen to join this elite group. Each champion possesses an exceptional VM-to-PM assignment, and they play a significant role in guiding the subsequent steps of the algorithm. These champions are given the responsibility to train new duelists, passing on their expertise and capabilities to the next generation of participants. As the game progresses, the champions will eventually be replaced by the new duelists they train, ensuring a continuous flow of competitive candidates.
- Duel Between Each Duelist: The dueling phase involves random matchups between the duelists. In these duels, each duelist utilizes their fighting skills, represented by their assigned VM-to-PM assignment, and an element of luck, determined by a random function. The combination of their skillset and luck will decide the outcome of the duel. Duelist A prevails if the combined effect of their fighting ability and luck surpasses that of duelist B, and vice versa.

Introducing randomness helps the algorithm to avoid being trapped in local optima and enables it to explore various configurations more effectively.

- Duelist's Improvement: Following each duel, the duelists are categorized as either winners or losers based on their performances. The learning process takes place, where the losers receive training from the winners. In this context, learning means that the losers may adopt a portion of the winner's array or skillset, enhancing their own abilities. On the other hand, the winners undergo self-improvement through mutation. This process involves advancing and refining their skillset, introducing variations that may lead to more advanced solutions. The combination of learning and mutation allows the duelists to continually evolve and adapt in the competitive environment.
- Elimination: As new duelists join the competition, the total number of participants may exceed a predefined limit. To maintain a constant pool of duelists and prevent excessive computational burden, an elimination process is carried out. Based on their individual dueling abilities, the weakest performing duelists are eliminated from the competition. The duelist with the poorest performance in a duel is the one to be eliminated. This ensures that only the strongest and most competitive duelists continue to participate in the algorithm's iterative progression.

In the basic the Duelist algorithm, the following parameters play a crucial role in shaping the algorithm's behaviour and performance.

- Luck: The luck parameter represents the influence of randomness in the dueling process. It determines the extent to which luck affects the outcome of the duel between two duelists. A higher luck value introduces more randomness, allowing the algorithm to explore a wider range of solutions. On the other hand, a lower luck value reduces randomness and makes the algorithm more deterministic, focusing on exploiting promising regions.
- Mutation: The mutation parameter controls the rate at which the skillset of a winning duelist undergoes changes or mutations. Mutation introduces small variations in the skillset, allowing the algorithm to explore nearby solutions. A higher mutation rate promotes greater exploration, potentially aiding in escaping local optima. Conversely, a lower mutation rate encourages exploitation of known good solutions.
- Innovation (Winner Learning Probability): In the basic Duelist algorithm, the innovation parameter, represented as the winner learning probability, determines the likelihood that a winner will learn from its own skillset and improve. A higher innovation probability encourages winners to explore their skillsets, potentially leading to more advanced solutions and increased diversity.
- Learning (Loser Learning Probability): The learning parameter, represented as the loser learning probability, controls the probability that a loser will adopt a portion of the winner's skillset during the learning process. A higher learning probability enables greater knowledge transfer from winners to losers, promoting convergence towards better solutions.
- Number of Champions: The number of champions parameter determines the size of the champion's board. It specifies the number of best-performing duelists that are considered champions and contribute to training new duelists. A larger board size may increase the diversity of the trained duelists, potentially improving exploration capabilities.

3.2 Proposed Duelist Algorithm Variations

In the different variations of the Duelist algorithm, certain parameters may be modified or removed altogether, as described in the specific variations. The innovation and learning probabilities, along with the number of champions, continue to be important parameters in these variations as they affect the learning and adaptation mechanisms of the algorithm. Depending on the problem characteristics and optimization goals, tuning these parameters is crucial to achieving good performance and convergence properties in the Duelist algorithm.

Fig. 2 and Fig. 3 present the proposed variations 1 and 2 of Duslist algorithms. The modifications proposed in variation 1 are as follows.

- Removed Luck Factor: The luck factor, which was previously used to introduce randomness in the dueling process, has been eliminated. In this variation, the outcomes of duels are solely determined by the fighting abilities and skillsets of the duelists. By removing the luck factor, the algorithm's behavior becomes more deterministic, and the exploration of the search space becomes less random. This modification may lead to a more focused search and potentially a faster convergence to optimal or near-optimal solutions.
- Eliminated Champions Training Duelists that are Themselves: In the Duelist algorithm, champions were responsible for training new duelists, who inherited the same capabilities as the champions. In variation 1, this mechanism has been removed. Champions no longer train new duelists that are similar to themselves. Instead, new duelists are generated through other means or strategies. This change can diversify the population of duelists, introducing more variations and novel solutions in each generation. It may also reduce the risk of premature convergence by avoiding an over-reliance on a specific set of champion solutions.

By incorporating these modifications, variation 1 of the Duelist algorithm aims to explore the search space more efficiently, enhance diversity among the duelists, and potentially improve the algorithm's ability to find better solutions in complex optimization problems. However, the effectiveness of these changes depends on the specific problem being addressed and the characteristics of the fitness landscape. Extensive experimentation and comparative analysis with the basic Duelist algorithm are required to assess the performance of variation 1 and its suitability for different optimization tasks.

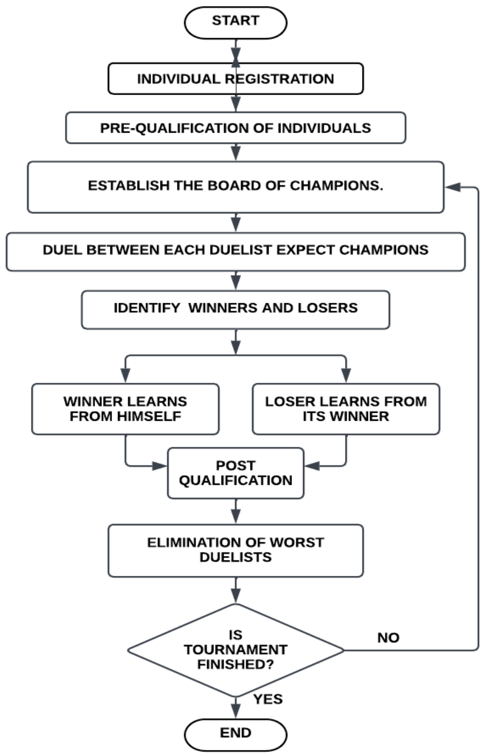

Fig. 2. Proposed variation 1 of DA

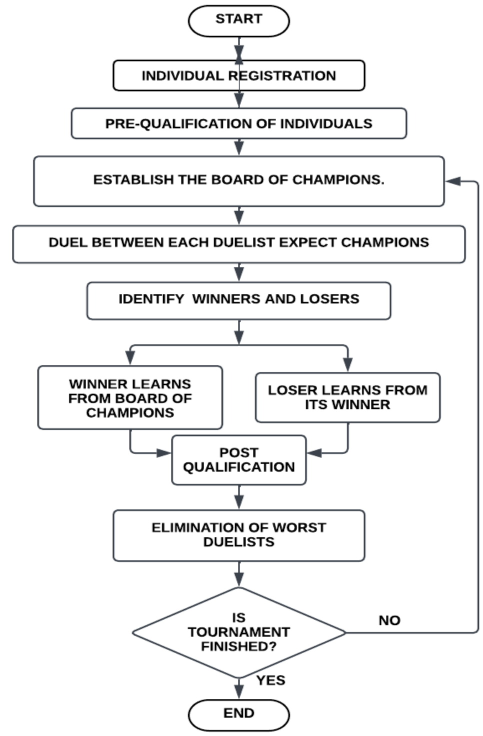

Fig. 3. Proposed variation 2 of DA

In variation 2 of the Duelist algorithm, three significant modifications have been introduced, building upon the changes made in Variation 1:

- Removed Luck Factor: Similar to variation 1, the luck factor has been eliminated from the dueling process in variation 2. The outcomes of the duels are solely determined by the fighting abilities and skillsets of the duelists, making the algorithm more deterministic in its search behaviour.
- Eliminated Champions Training Duelists that are Themselves: As in variation 1, the mechanism where champions train new duelists that are similar to themselves has been removed. New duelists are now generated using other methods or strategies to promote diversity and exploration in the population.
- Winner Learns from Board of Champions (Group of Champions): In variation 2, after each duel, the winner gains an opportunity to learn not just from a single champion but from a group of champions known as the "Board of Champions". This board consists of the top-performing duelists from previous iterations. The winner incorporates knowledge from this diverse group of champions to enhance its own skillset and abilities. This learning from multiple champions allows the algorithm to leverage the collective wisdom and best practices of successful solutions, potentially leading to better convergence and improved performance.
- The introduction of this learning mechanism from the Board of Champions further enhances the Duelist algorithm's adaptability and ability to explore promising regions of the search space. It can help in efficiently capturing valuable knowledge from the champions and propagate it to other duelists, fostering the discovery of better solutions in the subsequent iterations.

By combining these modifications, variation 2 of the Duelist algorithm aims to strike a balance between exploration and exploitation, promoting diversity among duelists, and leveraging the knowledge of successful solutions from the Board of Champions to drive the algorithm towards more promising regions of the solution space. As with any algorithmic modification, the performance of variation 2 would require rigorous testing and evaluation on a variety of optimization problems to assess its effectiveness and efficiency compared to previous variations.

3.3 Solution Representation and initialization

The Duelist algorithm represents a solution to the assignment of Virtual Machines (VMs) to Physical Machines (PMs) using a 1D array. The length of the array is equal to the number of VMs, and each element of the array corresponds to a VM, indicating which PM it is assigned to. Each VM is allocated to one of the PMs. In the given representation (figure 4), there are seven VMs (VM1 to VM7) and three PMs (PM1, PM2, and PM3). The array represents the assignment of VMs to PMs. For example, in the given configuration:

- PM1 contains VM2, VM3, and VM6.
- PM2 contains VM1 and VM4.
- PM3 contains VM5 and VM7.

The array elements show the VM-to-PM assignments, indicating which VM is hosted on which PM. This solution representation allows the Duelist algorithm to explore different configurations of VM-PM assignments and optimize the assignment to achieve specific objectives, such as minimizing resource utilization, balancing the workload, or optimizing power consumption in virtualized environments.

PM_2	PM_1	PM_1	PM_2	PM_3	PM_1	PM_3
VM_1	VM_2	VM_3	VM_4	VM_5	VM_6	VM_7

Fig. 4. Solution representation

In the initialization phase of the Duelist algorithm for the scenario with 50 VMs and 35 PMs, an array is generated randomly to represent the initial assignment of VMs to PMs. The size of this array is 50, where each element corresponds to a VM and indicates the PM on which the VM is mounted. As the initialization is done randomly, the assignment of VMs to PMs is not optimized at this point and serves as the starting point for the Duelist algorithm's iterative process.

4. Dataset, Results and Discussion

This section provides the simulation details necessary for conducting experiments to assess the performance of algorithms in solving the virtual machine placement problem. Various datasets have been utilized for this purpose, and the duelist algorithm along with its proposed variations have been tested using 15 different datasets. These datasets were generated using a Python program, which produces datasets with diverse characteristics, including different numbers of virtual machines (VMs) and physical machines (PMs). The datasets are classified as small, medium, and large based on the number of VMs and PMs they contain, with the maximum number of PMs being 450 and the maximum number of VMs being 600. The minimum number of PMs is 35, and the minimum number of VMs is 50. The CPU values for virtual machines are generated within the range of 50 to 150, while adhering to the restriction that the VM CPU values must be smaller than the available PM CPU capacities. Two Python programs were developed to generate these datasets. The first program generates inputs randomly, creating random CPU values for virtual machines. This method produces five datasets. The remaining ten datasets are generated by the second program, which generates inputs following a normal distribution, thus creating VM CPU values distributed normally. Some parameter values were fixed according to the established standards.

Table 1
Experimental scenario and Dataset description

Dataset	Number of		CPU usage of (in MIPS)		PM CPU at busy state	Category of Dataset	Method of generation
	PM	VM	PM	VM			
1	35	50				Small	
2	40	100				Small	
3	50	120				Small	Random
4	70	150				Small	
5	100	200				Medium	
6	150	200				Medium	
7	150	300				Medium	
8	200	300	1000 (MIPS)	50 to 150 (MIPS)	250 (MIPS)	Medium	
9	250	300				Medium	
10	290	400				Medium	Normal
11	300	450				Medium	
12	350	500				Large	
13	400	500				Large	
14	450	600				Large	
15	500	700				Large	

Table 1 presents the specifications for virtual machines and physical machines along with their corresponding values, with CPU measured in Million Instructions per Second (MIPS) and provides details of the datasets used in the experiments. A total of 15 datasets were employed to evaluate the performance of the algorithms. These datasets are categorized as small, medium, and large based on the number of VMs and PMs they contain. The table also indicates the method of dataset generation, specifying whether the dataset was randomly generated or generated using normal distribution.

For the experiment, the algorithm is tested using a combination of heterogeneous virtual machines and homogeneous physical machines. The input to the algorithm includes the number of VMs, CPU values of the virtual machines, the number of physical machines, CPU values of the physical machines, and CPU values at the busy state of the physical machines. The duelist algorithm and its variations are implemented using the 'Python' programming language. The execution of the algorithm programs takes place on a system equipped with an Intel(R) Core(TM) i5-6300U CPU - 2.50GHz - 2.40 GHz and 8 GB RAM.

4.1 Results Comparison

To assess the performance of the algorithms, this study evaluates the best, mean, and standard deviation values. The best fitness value represents the optimal result obtained from ten program executions. The mean value is the average fitness value of the population obtained at the 100th iteration, and the standard deviation is the statistical measure of the variability within that population. In reference (Suganthan et al., 2005), six matrices are provided to measure the effectiveness of the algorithms, including the success rate (SR), the number of function evaluations (NFEs), the convergence graph, the improvement rate, and the acceleration rate (AR). The following equations are used to calculate these results. Table 2 presents the results of basic Duelist algorithm and proposed two variations.

$$Success\ rate\ = \frac{number\ of\ times\ value\ obtained\ is\ greater\ than\ average\ of\ \ all\ execution}{number\ of\ times\ algorithm\ executed} \tag{3}$$

$$NFEs = Population\ \times Number\ of\ generations\ needed\ for\ convergence \tag{4}$$

$$Acceleration\ rate\ = \frac{NFE\ of\ proposed\ variation}{NFE\ of\ basic\ DA} \tag{5}$$

$$Improvement\ = \left(NEF_{other} - NEF_{proposed}\right) \times \frac{100}{NEF_{other}} \tag{6}$$

- Throughout all datasets, Duelist Variation 2 consistently showcased superior performance when compared to Duelist Variation 1, as evidenced by its higher fitness values for both the best and mean solutions. The algorithm consistently achieved better optimization results, indicating its ability to find solutions with higher fitness scores. Additionally, Duelist Variation 2 demonstrated improved average performance across all datasets, suggesting that it was consistently more effective in converging towards better solutions compared to Duelist Variation 1.
- In addition to better fitness values, Duelist Variation 2 demonstrated another advantage over Duelist Variation 1. It consistently exhibited lower standard deviation values across all datasets. A lower standard deviation implies more stable and consistent performance throughout the optimization process. This indicates that Duelist Variation 2 was not only capable of achieving higher fitness scores but also demonstrated a more reliable and consistent performance in generating solutions, reducing the variability in its results compared to Duelist Variation 1.
- The observed performance differences between Duelist Variation 2 and Duelist Variation 1 can be attributed to the specific modifications made in Duelist Variation 2. By eliminating the luck factor and introducing learning from the Board of Champions, Duelist Variation 2 has incorporated new mechanisms to guide the optimization process. These modifications have proven to be effective in achieving better and more reliable optimization outcomes compared to Duelist Variation 1. The removal of luck reduces randomness and potential biases in the optimization, while learning from the Board of Champions likely allows Duelist Variation 2 to benefit from the accumulated knowledge and experience of the champions, leading to more informed and effective search strategies. Overall, these improvements in Duelist Variation 2 have contributed to its superiority over Duelist Variation 1 in terms of optimization performance.
- Duelist Variation 2 consistently outperformed the Basic Duelist Algorithm and Duelist Variation 1 in terms of improvement percentage. This trend suggests that Duelist Variation 2 possesses enhanced optimization capabilities, enabling it to achieve better results in terms of solution quality or convergence towards the optimal solution.
- In specific datasets (e.g., 1, 2, 4, 6, 9, 11, 13 and 15), Duelist Variation 2 exhibited remarkable improvements, exceeding 50% in comparison to the Basic Duelist Algorithm and Duelist Variation 1. These datasets indicate scenarios where Duelist Variation 2 was particularly effective in finding significantly better solutions than the other algorithms. The noteworthy improvements signify the algorithm's ability to efficiently explore and exploit the search space, leading to substantial enhancements in optimization results.
- This statement highlights a specific observation in Dataset 14, where Duelist Variation 2 showed only a marginal

improvement of 2% over the Basic Duelist Algorithm and Duelist Variation 1. In this particular case, Duelist Variation 2's performance was relatively close to the other algorithms, resulting in a minor enhancement in comparison. This suggests that there might be certain scenarios or characteristics in Dataset 14 where Duelist Variation 2's optimization approach had limited impact, leading to a small improvement percentage compared to Duelist Variation 1. Further analysis may be needed to understand the factors contributing to this minimal improvement and whether it is a consistent pattern across multiple datasets.

Table 2
Best, mean, standard deviation on obtained results

Dataset		Duelist	Variation 1	Variation 2	Improvements V1 (in %)	Improvements V2 (in %)
1	Best	2681	2081	1931	42.50	58.60
	Mean	2960	2331	2078		
	SD	56	73	21		
2	Best	5023	4423	3823	32.66	65.30
	Mean	5578	4702	3971		
	SD	77	56	14		
3	Best	6600	5400	4500	4	32
	Mean	7282	5713	4648		
	SD	77	101	14		
4	Best	8861	7361	5711	10	59
	Mean	9737	7958	5955		
	SD	58	149	75		
5	Best	12747	11098	8247	34.54	45.45
	Mean	13776	11515	8392		
	SD	76	137	39		
6	Best	17951	16001	11801	35.16	68.13
	Mean	19163	16928	12312		
	SD	117	243	137		
7	Best	20175	18225	13575	17	45
	Mean	21324	19284	13827		
	SD	90	253	73		
8	Best	23025	21525	14775	11	47
	Mean	24540	22272	15034		
	SD	153	266	87		
9	Best	25275	22275	15525	27	68.13
	Mean	27126	23100	16053		
	SD	172	256	109		
10	Best	32880	21930	19980	4	52
	Mean	34498	22401	20296		
	SD	64	145	99		
11	Best	35699	24299	22499	43	50
	Mean	37472	24542	22751		
	SD	189	81	76		
12	Best	40883	29934	26183	6	42
	Mean	42839	30689	26361		
	SD	103	244	62		
13	Best	43133	29484	26183	15	58
	Mean	45496	29896	26454		
	SD	74	149	62		
14	Best	50857	38407	32707	12	22
	Mean	53116	39368	33007		
	SD	155	284	108		
15	Best	48981	38181	**33381**	3	61
	Mean	50974	38711	33625		
	SD	87	170	83		

Table 3 presents the results of success rate of basic Duelist algorithm and proposed two variations. Duelist Variation 2 consistently demonstrated better success rates across most datasets compared to Duelist and Duelist Variation 1. The higher success rates of Duelist Variation 2 indicate its ability to efficiently explore the solution space and find feasible solutions within the constraints of the problem. For some datasets, Duelist Variation 2 achieved a perfect success rate of 100%, indicating its reliability in finding optimal or near-optimal solutions for those particular problem instances. The results suggest that Duelist Variation 2's modifications, such as the removal of the luck factor and the introduction of learning from the Board of Champions, have improved its ability to find feasible solutions with higher success rates. In Table 4, the given results provide a comparison of the number of function evaluations required by the Duelist Algorithm and its two Variations (Variation 1 and Variation 2) to reach their respective base fitness values in different datasets. Additionally, the acceleration rates for each Variation are calculated to understand the efficiency of the algorithms in converging towards the optimal solution.

The given results provide a comparative analysis of the performance of three algorithms, namely the Duelist Algorithm (DA), Variation 1 and Variation 2 across multiple datasets. Each dataset is associated with a specific base fitness value, and the algorithms' efficiency is evaluated based on the number of function evaluations required to reach this value. Additionally, the acceleration rates of Variation 1 and Variation 2 compared to the Duelist Algorithm are calculated, indicating their convergence speed towards the base value.

Table 3
Average, best value and success rate

Dataset	Category of dataset	Duelist			Duelist Variation 1			Duelist Variation 2		
		Average	Best	SR	Average	Best	SR	Average	Best	SR
1	Small	2801	2681	60	2141	2081	60	**1961**	**1931**	80
2	Small	5203	5023	60	4573	4423	80	**3823**	**3823**	100
3	Small	6690	6600	40	5790	5400	60	**4740**	**4500**	60
4	Small	8951	8861	40	8441	7361	60	**6131**	**5711**	60
5	Medium	13017	13017	40	11997	11098	40	**8697**	**8247**	60
6	Medium	18101	17951	60	16541	16001	80	**12251**	**11801**	80
7	Medium	20265	20175	40	18855	18225	40	**14265**	**13575**	40
8	Medium	23226	23025	40	21775	21525	60	**15825**	**14775**	60
9	Medium	25575	25225	60	23565	22275	60	**15855**	**15525**	60
10	Medium	32850	32580	60	22170	21930	60	**20570**	**19980**	60
11	Medium	35849	35699	40	24299	25469	60	**22829**	**22499**	80
12	Large	40973	40883	40	28673	27683	60	**26453**	**26183**	60
13	Large	43343	43133	40	32663	29484	40	**27293**	**26183**	80
14	Large	51307	50857	60	39457	38407	60	**33907**	**32707**	80
15	Large	49192	48981	60	38181	39891.4	60	**34281**	**33381**	60

Table 4
Function Evaluation and Acceleration rate

Dataset	Base fitness value	Number of Function Evaluations for Duelist	Number of Function Evaluations for Variation 1	Number of Function Evaluations for Variation 2	Acceleration rate V1	Acceleration rate V2
1	2681	8700	5000	3600	1.74	2.41
2	5023	9800	6600	3400	1.48	2.88
3	6600	5000	4800	3400	1.04	1.47
4	8861	6800	6100	2800	1.11	2.42
5	13017	5500	3600	3000	1.52	1.83
6	17951	9100	5900	2900	1.54	3.13
7	20175	4700	3900	2600	1.20	1.80
8	23025	6700	6000	3600	1.11	1.86
9	25225	9100	6700	2900	1.35	3.13
10	33030	5000	4800	2400	1.04	2.08
11	35699	9300	5300	2100	1.75	4.42
12	40883	6300	5900	3700	1.06	1.70
13	43133	7600	6500	3200	1.16	2.37
14	50857	6000	5300	4700	1.13	1.27
15	48981	7800	7600	3100	1.02	2.52

In general, all Variations of the Duelist Algorithm demonstrated improvements over the base fitness values in each dataset, which indicates their effectiveness in optimizing the solutions. However, Duelist Variation 2 consistently outperformed both the Duelist Algorithm and Variation 1, achieving lower fitness values in most datasets.

The acceleration rates of Variation 2 are generally higher than those of Variation 1, suggesting that Variation 2 converges more efficiently towards the base fitness value. This improvement in acceleration rate signifies Variation 2's enhanced optimization capabilities and efficiency, making it a more reliable and effective optimization algorithm.

Dataset-specific analysis revealed interesting observations. In several datasets (e.g., Datasets 6, 9, 11, and 13), Duelist Variation 2 demonstrated remarkable improvements, converging towards the optimal solution significantly faster than the other Variations. However, there were instances where Variation 1 performed marginally better, as seen in Dataset 14.

- Performance Comparison: The comparison between Duelist Variation 1 and Variation 2 shows that both algorithms achieve improvements over the base value in all datasets. This indicates that both Variations of the Duelist Algorithm are effective in optimizing the given problems. However, it is observed that Duelist Variation 2 consistently outperforms Variation 1. Lower fitness values indicate better solutions, signifying that Duelist Variation 2 is more capable of finding optimal or near-optimal solutions compared to Variation 1.
- Acceleration Rate: The acceleration rate is a measure of how quickly the algorithm converges towards the base value, which represents the optimal solution. In this context, a higher acceleration rate indicates that the algorithm converges more efficiently and requires fewer function evaluations to reach the base value. The analysis shows that Duelist Variation 2 exhibits a higher acceleration rate compared to Variation 1 for most datasets. This indicates that Duelist

Variation 2 is more efficient in converging towards optimal solutions than Variation 1. The higher acceleration rate of Variation 2 suggests that it can reach a good solution with fewer function evaluations, making it more time and resource-efficient.

- Dataset-specific Analysis: The analysis also considers the performance of the two Variations of the Duelist Algorithm in specific datasets. For datasets 2, 5, 6, 9, 11, 13, and 15, Duelist Variation 2 displays significantly higher acceleration rates compared to Variation 1. This indicates that Variation 2 is particularly effective in these datasets, as it converges faster towards the optimal solution. However, in dataset 14, Duelist Variation 2 shows a slightly lower acceleration rate compared to Variation 1. This suggests that Variation 1 performs marginally better in this specific dataset. These findings imply that the effectiveness of each Variation may vary depending on the characteristics of the dataset and the nature of the optimization problem.

Table 5

Comparison of Hybrid HSA and Variation 2 of DA

Dataset	Fitness value of Hybrid HS+SA+LS (Adamuthe, A., & Kagwade, S. (2022)	Fitness value of Variation 2	Iterations
1	2381	**1931**	100
2	4475	**3823**	
3	5845	**4500**	
4	7452	**5711**	
5	10145	**8247**	
6	13235	**11801**	
7	62210	**13575**	
8	11775	**9775**	
9	15525	**15525**	
10	21002	**19980**	200
11	24006	**22499**	
12	28569	**26183**	
13	28230	**26183**	
14	34001	**32707**	
15	**32122**	33381	

In Table 5, The given results represent the fitness values of two different optimization algorithms, namely the Hybrid Algorithm (HS+SA+LS) and Variation 2, across multiple datasets in table 5. Each dataset is associated with a specific dataset number, and the fitness values represent the quality of the solutions obtained by the algorithms for each dataset. In general, both the Hybrid Algorithm and Variation 2 demonstrate improvements over the fitness values across all datasets, indicating their effectiveness in optimizing the solutions. The fitness values obtained by Variation 2 are consistently lower than those obtained by the Hybrid Algorithm, suggesting that Variation 2 achieves better optimization results in most cases. The results suggest that Variation 2 is a more efficient and effective optimization algorithm compared to the Hybrid Algorithm. It consistently produces better fitness values and better-quality solutions for a diverse range of datasets. Dataset-specific analysis reveals that Variation 2 significantly outperforms the Hybrid Algorithm in most datasets. For example, in Datasets 1, 2, 3, 4, 5, 6, 7,8,9, 10, 11, 12, 13 and 14, the fitness values of Variation 2 are notably lower than those of the Hybrid Algorithm, indicating its enhanced optimization capabilities in these datasets. Overall, the results demonstrate that Variation 2 is a superior optimization algorithm compared to the Hybrid Algorithm. Its ability to consistently achieve lower fitness values across various datasets indicates its robustness and efficiency in finding better solutions. The modifications made in Variation 2 have evidently resulted in improved optimization performance, making it a more reliable choice for solving optimization problems.

4.2 Convergence analysis

The convergence behaviour of different algorithms is a crucial aspect in assessing their effectiveness. In this context, Figs. (5-10) provide insight into the convergence curves of the Duelist algorithm and its proposed variations across datasets of varying sizes. Fig. 5 and Fig. 6 illustrate the convergence curves for two small datasets. These curves depict how the fitness values of the algorithms evolve over iterations. Similarly, Fig. 7 and Fig. 8 showcase the convergence curves for two medium-sized datasets, while Fig. 9 and Fig. 10 present the convergence curves for two large datasets. These visualizations offer a comparative understanding of the algorithms' convergence tendencies across datasets with diverse complexities. A notable observation from these results is that the proposed variations of the Duelist algorithm consistently outperform the basic Duelist algorithm in terms of convergence speed. The basic Duelist algorithm often encounters challenges in escaping local optima, resulting in slower convergence rates. In contrast, the proposed variations demonstrate the ability to avoid such pitfalls and maintain steady progress towards optimal solutions. This is particularly evident across all six datasets, regardless of their sizes. The convergence curves serve as a valuable means of gauging algorithm performance, providing insights into their ability to explore and exploit the solution space effectively. The superiority of the proposed variations in achieving faster convergence is indicative of their enhanced capability to adapt, learn, and optimize solutions in a variety of scenarios. This observation underscores the significance of introducing novel approaches, such as the proposed variations, to enhance the efficiency and effectiveness of optimization algorithms like the Duelist algorithm.

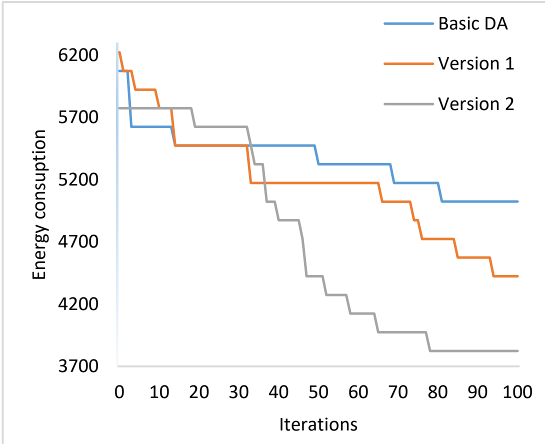

Fig. 5. Convergence graph for Dataset 2

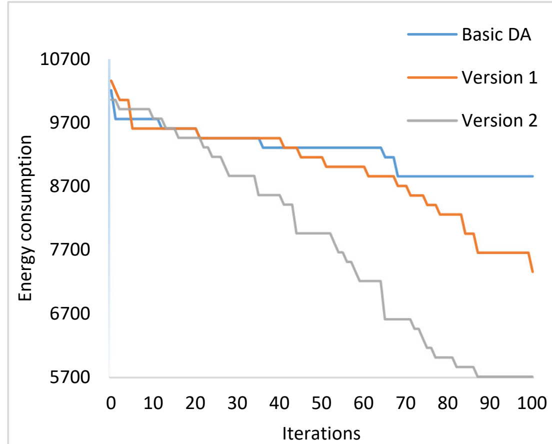

Fig. 6. Convergence graph for Dataset 4

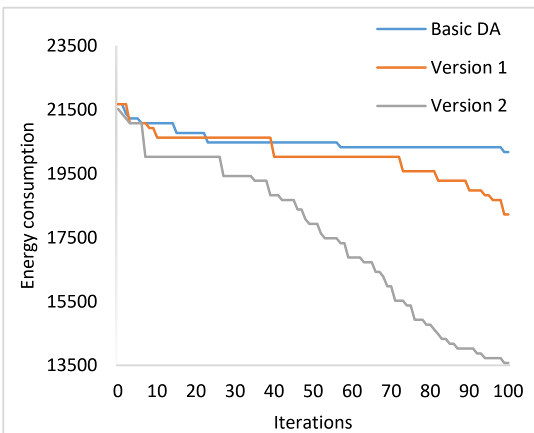

Fig. 7. Convergence graph for Dataset 7

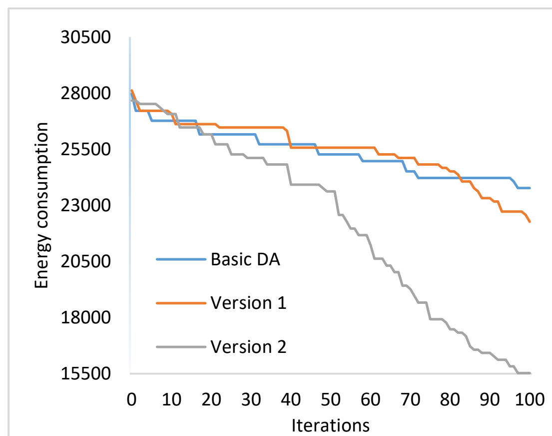

Fig. 8. Convergence graph for Dataset 9

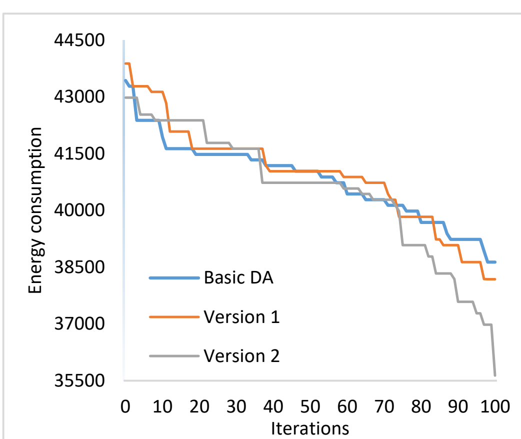

Fig. 9. Convergence graph for Dataset 12

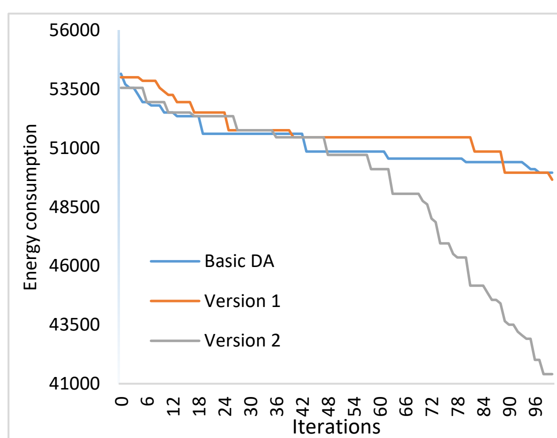

Fig. 10. Convergence graph for Dataset 14

5. Conclusions

The paper has presented virtual machine placement problem in cloud computing with the objective of minimization of energy consumption in cloud data centers. The proposed algorithms, namely Duelist Variation 1 and Duelist Variation 2, have been rigorously tested and compared with the Basic Duelist Algorithm using various datasets.

Firstly, the Duelist Variation 1 introduces an improvement over the Basic Duelist Algorithm by removing the luck factor and allowing champions to train new duelists. This modification shows positive results, as Duelist Variation 1 consistently demonstrates improvements in fitness values compared to the Basic Duelist Algorithm across most datasets. However, it is evident that there is room for further enhancement. Secondly, the more advanced Duelist Variation 2 goes a step further by removing the luck factor, allowing removal of champions to train new duelists stage, and introducing learning from the Board of Champions. This additional learning capability proves to be highly beneficial, as Duelist Variation 2 consistently outperforms both the Basic Duelist Algorithm and Duelist Variation 1 in terms of fitness values. It exhibits higher acceleration rates and converges faster towards optimal fitness values for a majority of datasets.

The comparison of the acceleration rates clearly indicates that Duelist Variation 2 is more efficient in finding better solutions with a reduced number of function evaluations. This efficiency is particularly evident in datasets 1, 2, 4, 6, 9, 11, 13 and 15, where Duelist Variation 2 exhibits remarkable improvements of over 50% compared to the Basic Duelist Algorithm. However, it is essential to consider that the performance of the algorithms may vary slightly depending on the specific dataset. Nonetheless, overall, Duelist Variation 2 demonstrates enhanced optimization capabilities, consistent improvements, and more stable performance compared to the other Variations. The proposed algorithms, particularly Duelist Variation 2, showcase substantial improvements over the Basic Duelist Algorithm and Duelist Variation 1. They exhibit higher acceleration rates, converge towards better fitness values, and demonstrate enhanced optimization capabilities. The modifications made in Duelist Variation 2, such as the removal of the luck factor and the introduction of winner learning from the Board of Champions, have proven to be effective in achieving these superior results. The experimental findings validate the suitability and reliability of Duelist Variation 2 for a wide range of optimization tasks. As a more advanced and efficient algorithm, it presents a valuable contribution to the field of optimization, providing researchers and practitioners with a powerful tool for solving complex optimization problems

References

15 Crucial Data Center Statistics to Know in 2023. (n.d.). Techjury. Retrieved from https://techjury.net/blog/data-center-statistics/

Abdel-Basset, M., Abdle-Fatah, L., & Sangaiah, A. K. (2019). An improved Lévy based whale optimization algorithm for bandwidth-efficient virtual machine placement in a cloud computing environment. *Cluster Computing*, *22*(4), 8319-8334.

Abohamama, A. S., & Hamouda, E. (2020). A hybrid energy–aware virtual machine placement algorithm for cloud environments. *Expert Systems with Applications*, *150*, 113306.

Adamuthe, A., & Kagwade, S. (2022). Hybrid and adaptive harmony search algorithm for optimizing energy efficiency in VMP problem in cloud environment. *Decision Science Letters*, *11*(2), 113-126.

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.

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

Alsadie, D. (2022). Virtual Machine Placement Methods using Metaheuristic Algorithms in a Cloud Environment-A Comprehensive Review. *International Journal of Computer Science & Network Security*, *22*(4), 147-158.

Batista, D. M., Da Fonseca, N. L., & Miyazawa, F. K. (2007, March). A set of schedulers for grid networks. In *Proceedings of the 2007 ACM symposium on Applied computing* (pp. 209-213).

Beloglazov, A., & Buyya, R. (2010, May). Energy efficient allocation of virtual machines in cloud data centers. In *2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing* (pp. 577-578). IEEE.

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

Biran, O., Corradi, A., Fanelli, M., Foschini, L., Nus, A., Raz, D., & Silvera, E. (2012, May). A stable network-aware vm placement for cloud systems. In *2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012)* (pp. 498-506). IEEE.

Biyanto, T. R., Fibrianto, H. Y., Nugroho, G., Hatta, A. M., Listijorini, E., Budiati, T., & Huda, H. (2016a). Duelist algorithm: an algorithm inspired by how duelist improve their capabilities in a duel. In *Advances in Swarm Intelligence: 7th International Conference, ICSI 2016, Bali, Indonesia, June 25-30, 2016, Proceedings, Part I 7* (pp. 39-47). Springer International Publishing.

Biyanto, T. R., Irawan, S., Ginting, H. J., & Fitri, A. I. (2017a). Operating Conditions Optimization of Steam Injection in Enhanced Oil Recovery Using Duelist Algorithm. *International Journal of Industrial and Manufacturing Engineering*, *11*(2), 272-277.

Biyanto, T. R., Matradji, Syamsi, M. N., Fibrianto, H. Y., Afdanny, N., Rahman, A. H., ... & Putra, Y. A. (2017b, November). Optimization of energy efficiency and conservation in green building design using Duelist, Killer-Whale and Rain-Water Algorithms. In *IOP conference series: materials science and engineering* (Vol. 267, No. 1, p. 012036). IOP Publishing.

Biyanto, T. R., Ramasamy, M., Jameran, A. B., & Fibrianto, H. Y. (2016b). Thermal and hydraulic impacts consideration in refinery crude preheat train cleaning scheduling using recent stochastic optimization methods. *Applied Thermal Engineering*, *108*, 1436-1450.

Breitgand, D., & Epstein, A. (2011, May). SLA-aware placement of multi-virtual machine elastic services in compute clouds. In *12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops* (pp. 161-168). IEEE.

Buyya, R., Broberg, J., & Goscinski, A. M. (Eds.). (2010). *Cloud computing: Principles and paradigms*. John Wiley & Sons.

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.

Chen, G., He, W., Liu, J., Nath, S., Rigas, L., Xiao, L., & Zhao, F. (2008, April). Energy-Aware Server Provisioning and Load Dispatching for Connection-Intensive Internet Services. In *NSDI* (Vol. 8, pp. 337-350).

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* (pp. 1-6).

Energy Efficiency Predictions for Data Centres in 2023. (2022, December 30). Data Centre Magazine. Retrieved from https://www.datacentremagazine.com/articles/efficiency-to-loom-large-for-data-centre-industry-in-2023

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.

Ghribi, C., Hadji, M., & Zeghlache, D. (2013, May). Energy-efficient vm scheduling for cloud data centers: Exact allocation and migration algorithms. In *2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing* (pp. 671-678). IEEE.

IEA. (2022). Data Centres and Data Transmission Networks. IEA, Paris. Retrieved from https://www.iea.org/reports/data-centres-and-data-transmission-networks

Jayasinghe, D., Pu, C., Eilam, T., Steinder, M., Whally, I., & Snible, E. (2011, July). Improving performance and availability of services hosted on iaas clouds with structural constraint-aware virtual machine placement. In *2011 IEEE International Conference on Services Computing* (pp. 72-79). IEEE.

LASHIN, M. M., GAAFER, A. M., & Al Nemer, G. N. (2020). Optimization of dimensional, surface quality and material removal rate in turning using response surface methodology and duelist algorithm. *International Journal of Mechanical and Production Engineering Research and Development*, *10*(1), 499-514.

Pushpa, R., & Siddappa, M. (2022, January). Adaptive Hybrid Optimization Based Virtual Machine Placement in Cloud Computing. In *2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)* (pp. 1-9). IEEE.

Ruki Biyanto, T., Yernias Fibrianto, H., Nugroho, G., Listijorini, E., Budiati, T., & Huda, H. (2015). Duelist Algorithm: An Algorithm Inspired by How Duelist Improve Their Capabilities in a Duel. *arXiv e-prints*, arXiv-1512.

Shuja, J., Gani, A., Shamshirband, S., Ahmad, R. W., & Bilal, K. (2016). Sustainable cloud data centers: a survey of enabling techniques and technologies. *Renewable and Sustainable Energy Reviews*, *62*, 195-214.

Singh, A. K., Swain, S. R., Saxena, D., & Lee, C. N. (2023). A bio-inspired virtual machine placement toward sustainable cloud resource management. *IEEE Systems Journal*.

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.

Suganthan, P. N., Hansen, N., Liang, J. J., Deb, K., Chen, Y. P., Auger, A., & Tiwari, S. (2005). Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. *KanGAL report*, *2005005*(2005), 2005.

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

Usmani, Z., & Singh, S. (2016). A survey of virtual machine placement techniques in a cloud data center. *Procedia Computer Science*, *78*, 491-498.

Van den Bossche, R., Vanmechelen, K., & Broeckhove, J. (2010, July). Cost-optimal scheduling in hybrid iaas clouds for deadline constrained workloads. In *2010 IEEE 3rd international conference on cloud computing* (pp. 228-235). IEEE.

Wood, T., Shenoy, P., Venkataramani, A., & Yousif, M. (2009). Sandpiper: Black-box and gray-box resource management for virtual machines. *Computer Networks*, *53*(17), 2923-2938.

Wu, G., Tang, M., Tian, Y. C., & Li, W. (2012). Energy-efficient virtual machine placement in data centers by genetic algorithm. In *Neural Information Processing: 19th International Conference, ICONIP 2012, Doha, Qatar, November 12-15, 2012, Proceedings, Part III 19* (pp. 315-323). Springer Berlin Heidelberg.

Xing, H., Zhu, J., Qu, R., Dai, P., Luo, S., & Iqbal, M. A. (2022). An ACO for energy-efficient and traffic-aware virtual machine placement in cloud computing. *Swarm and Evolutionary Computation*, *68*, 101012.

Zhu, W., Zhuang, Y., & Zhang, L. (2017). A three-dimensional virtual resource scheduling method for energy saving in cloud computing. *Future Generation Computer Systems*, *69*, 66-74.