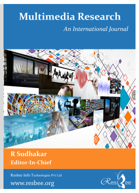
Vol.3 No.1 Jan 2020

Multimedia Research

Received 1 September, Revised 3 November, Accepted 25 December



Optimal Container Resource Allocation Using Hybrid SA-MFO Algorithm in Cloud Architecture

VhatkarKapilNetaji
V.J.T.I.
Mumbai, Maharashtra, India
vhatkarkapilnetaji@gmail.com

Bhole G P
VIT,
Mumbai, Maharashtra, India

Abstract: Owing to the merits of container practice such as easier and more rapid consumption, superior portability, and limited overheads, it can be extensively installed over the cloud architecture. Then, a suitable architecture solution is proposed to develop the applications, which are produced using the microservice expansion model. Thus far, numerous research works have determined on resolving the open problems in container management and automation. In reality, for cloud providers, container resource allocation is considered as the main knothole as it directly influences the system performance and resource utilization. In this way, this work initiates a novel optimized container resource allocation framework by developing a novel optimization theory. Here, a novel hybrid approach is proposed such as, SA and MFO that is the hybridization of Simulated Annealing (SA) and Moth Flame Optimization Algorithm (MFOA) to create the prospect of optimal container resource allocation. In addition, the solution of optimized resource allocation is inclined with the modeling of a novel objective model which contemplates system failure, threshold distance, total network distance, and balanced cluster use, correspondingly. At last, the performance of the proposed approach is evaluated over other existing approaches and exhibits the performance of the proposed model.

Keywords: Resource Allocation; Cloud Computing; Microservices; Container Allocation; Optimization

Nomenclature

Abbreviation	Definition
QoS	Quality of Service
IoT	Internet of Things
HAPSO	Hybrid Adaptive Particle Swarm Optimization
HDFS	Hadoop Distributed File System
GA	Genetic Algorithm
FCC	Fog and Cloud Computing
ACO	Ant Colony Optimization
FNs	Fog Nodes
NOMA	Non-Orthogonal Multiple Access
ADMM	Alternating Direction Method of Multipliers
NN	Neural Network
CRAN	Cloud Radio Access Network
VMs	Virtual Machines
BBU	Base Band Unit
ICAM	Incentive-Compatible Auction Mechanism
BPA	Bandwidth Power Allocation
SD	Standard deviation
EARA	Energy-Aware Resource Allocation
MINLP	Mixed-Integer Nonlinear Programming

1. Introduction

The prompt applications and advancements of information technologies have transported an innovative transformation to contemporary product advancements [2]. By the exploit of powerful software packages, high-performance computers, and competent network services, the advancement and design procedure is accelerated and the quality of the product is enhanced in the meantime. Since the design model is performed further frequently using temporally and geographically distributed design teams concerning

https://doi.org/10.46253/j.mr.v3i1.a2

diverse roles like domain experts, modelers, verification and validation experts and end-users of diverse conditions, complex product advancements turn out to be more and more incorporated and collaborative. Therefore, for multiple users, the requirement of rising collaborative working platforms to distribute heterogeneous resources, and demeanor design tasks in a distributed and collaborative environment, is increased [1].

By means of the introduction of the IoT and the 5G epoch, conventional cloud computing applications were not capable to meet the requirements of low latency, high bandwidth on the edge side [9]. Famous instances comprise connected vehicles [11], streaming media [10], and the smart grid [12]. In addition, the number of mobile devices is rising [13]. To tackle the latency problem of the existing Cloud Computing platforms, the edge computing infrastructures were presented. At the edge of the network, edge computing indicates the enable applications permitting computation to be done, on upstream data in support of IoT services and downstream data in support of cloud services. In the IoT era, edge computing is considered a crucial infrastructure. Edge cloud computing is indicated as the edge cloud. On top of edge infrastructure, it constructs cloud computing platforms on the basis of the core capabilities of edge computing and cloud computing. To comprehend cloud-edge collaboration the edge cloud computing integrates edge computing and cloud computing.

For several years, to hold diverse computational technologies and convene the needs of users, cloud computing was developed. Service subscribers have the ability to use the resources that are controlled as a shared pool based on the demands of their users in cloud computing. Nevertheless, resources in the cloud are physically located distant from users, worsening to hold up low-latency technologies. For that reason, to enhance QoS for users and support the dynamic scalability, competent in-network processing, it is essential to pull the resources nearer to users and adapt the model of computing paradigms.

Generally, microservices is come out as a novel architecture, wherein huge and complex software technologies are collected of little one or further services. It can be installed in parallel with each other. These services are loosely coupled with one another [14] [15]. Each of these microservices charges dependable for carrying out merely one task competently. In cloud computing, microservices are extremely helpful for technologies. In cloud computing, the exploit of microservices permits raising the fame of the cloud [16] [17]. The employ of microservices presents additional alternatives and choices to independently develop the service. Recently, it comes out to accelerate the procedure of growth in web and mobile technologies.

The main contribution of this paper is to implement a novel optimized container resource allocation approach using system performance. Hence, a novel Hybrid SA-MFOA approach is proposed for the resource allocation of the optimal container. Moreover, the performance of the developed SA-MFOA approach is subsequently evaluated with the other existing methods regarding the modeled cost function.

2. Literature Review

In 2019, Chunlin Li et al [1] presented a data migration technique and an adaptive allocation of resource approach. For the adaptive resource allocation, the prediction approach offers the foundation of the edge cloud cluster. Moreover, to decide the allocation of resource scheme, the adaptive allocation of resource approach was exploited for the edge cloud cluster with the minimum service cost. The data migration technique assures the dependability of data and attains the cluster load balancing.

In 2017, Sadip Midya et al [2], developed a three-tier architecture comprising vehicular cloud, roadside cloudlet, and centralized cloud. Moreover, they had presented a task scheduling and an optimized allocation of resource technique to competently provide a huge number of task requests inward from on-road users when controlling enhanced QoS. Between the three cloud layers, these task requests were optimally mapped to cloud resources. Moreover, by exploiting the developed HAPSO technique that was an amalgamation of the GA and Adaptive PSO the optimization process was performed.

In 2016, K.P.N Jayasena et al [3], analyzed a new performance of multimedia big data distribution and data analyzing. Here, the developed architecture comprises three layers like platform layer, service layer, and infrastructure layer. By exploiting a MapReduce framework running on an HDFS and the media processing libraries Xuggler, the platform layer of the system was designed and implemented. In this manner, the developed system minimizes the transcoding time for huge numbers of data into definite formats based upon the user needs. Moreover, it offers flexible multimedia write/record interface and can model large scale multimedia big data analytics technologies on the basis of the Hadoop cloud platform. In addition, they had presented the ACO technique in the infrastructure layer for competent resource allocation.

In 2019, Yiming Liu et al [4], developed an incorporated FCC algorithm, whereas users can offload a sequence of applications to near FNs or cloud centers considerately. However, because of the constrained storage, computing, and radio resources, how to do resource allocation to attain a best and stable performance was a significant issue. Moreover, they have focused on multiple resource allocation issues in a broad system that comprises of multi-FN, multi-user, and a cloud center to address the aforesaid issue. Additionally, to minimize offloading latency of transmission and let go of the constraint of inadequate radio resource, NOMA, that enables multiple users to broadcast data to the similar FN for offloading tasks on the similar spectrum resource, was developed into the presented FCC algorithm. Moreover, they had decoupled the novel issue and convert it into a convex issue. At last, they had developed the ADMM on the basis of the techniques to resolve the optimization issue in an effectual manner and distributed manner.

In 2017, Gongzhuang Peng et al [5], developed a systematic model for evaluating, monitoring, and enhancing the system performance. In particular, a radial basis function NN was developed to convert the experimentation tasks with abstract explanations into the precise requirements of a resource regarding their qualities and quantities. Moreover, a new mathematical framework was developed to symbolize the complex allocation of resource procedures in a multi-tenant computing environment. It was done by taking into consideration of total computational cost.

In 2018, Ayman Younis [6], proposed a new resource allocation solution which optimizes the energy utilization of a CRAN. Initially, an energy utilization framework which exemplifies the calculation energy of the BBU pool was developed on the basis of the empirical outcomes gathered from a programmable C-RAN testbed. Subsequently, the resource allocation issue was divided into two subproblems such as the BPA and the BBU EARA. The BPA that is the initial cast by means of MINLP and after that reformulated as a convex issue, aspires at allocating a possible bandwidth and power to provide all users when meeting their QoS requirements.

In 2017, Fuhong Lin et al [7], worked on vehicular fog computing and it aspires to minimize serving time by assigning the obtainable bandwidth to four types of services. Moreover, a usefulness model was modeled in accordance with the serving aforesaid approaches and it was resolved by a two-step algorithm. For the initial step, all the sub-optimal solutions were given on the basis of the Lagrangian approach. An optimal solution selection procedure was analyzed and presented for the second step.

In 2016, A-Long Jin et al [8], developed an ICAM for resource trading among cloudlets as service providers and mobile devices as service users. Moreover, to assure the service demands of mobile devices and decide the pricing, ICAM can efficiently assign cloudlets. Both numerical results and theoretical analysis demonstrate that ICAM assures preferred properties regarding the budget balance, individual rationality, reliability (incentive compatibility) for both sellers and buyers and computational effectiveness.

3. Optimal Container Allocation Using the Hybrid Model

Consider a set of the application S_A then improved the pattern on the basis of the microservices. The description of all these applications al_x is done by exploiting the stack of microservices and the user request count us_x. Here, the microservices stack is represented as the series of microservices and the interoperability between them, via that the requirement of the application is applied. The enterprise of this interoperability association is occurred, only if the outcomes from the other microservices are utilized using these exacting microservices. Similarly, the microservices stack is modeled as a directed graph. The microservices n_y are indicated as nodes and the link between them is shown as edges. An edge links the two nodes $(n_{provider}, n_{consumer})_{pro/con}$ while the destination microservice $n_{consumer}$ consumes the result of novel microservice $n_{provider}$. Fig 1 demonstrates the proposed optimal container resource allocation model.

For each microservice, the exemplification has done with a tuple (t_y, ms_y, r_y, f_y), in that the threshold level for resource utilization is shown as t_y besides in that demeans the performance of service and add the bottleneck of application al_x is produced using microservices; the number of requests on microservice, which are necessary for satisfying a single application user request us_x is shown as ms_y; the resource on computation which inspired on fulfilling single microservice request is defined as r_y; and the micro service's failure rate is stated as f_y. us_x and ms_y values are on the basis of the system workload, and the r_y and f_y values are based on the microservice implementation.

In the system, the execution of each microservice is shortened as more than one container cn_z, and that is explained as $n_y \equiv c]n_z$. The container count is on the basis of the micro service's scale level sl_y.

Between the containers, the resource utilization of microservice is uniformly distributed; therefore the computation of container's computational resource utilization r_z is stated as per the eq. (1).

$$r_z = \frac{us_x \times ms_y \times r_y}{sl_y} \tag{1}$$

The Container Set (CS) is estimated in the container cluster encompassing a physical machine set py_u. $ac(cn_z) = py_u$ is the correlation that represents the container allocation cn_z to the py_u physical machine. The physical machine that assigns the container of a microservice n_y is shortened by the scheme $ac(n_y) = py_u$ whilst $cn_z \equiv n_y$ $ac(cn_z) = py_u$.

The tuple which exemplifies each physical machine is stated as cay_u and f_u, that shows the node's failure rate and computational capacity correspondingly. The most important system constraint is that the sum of computational resources exploited by the containers assigned to py_u must be lesser than the computational ability of this physical node. At last, the physical machines and physical network Ntw are interrelated, in that the paths between the nodes $py_u, py_{u'}$ are considered by the network distance dis_{py_u, py'_u}.

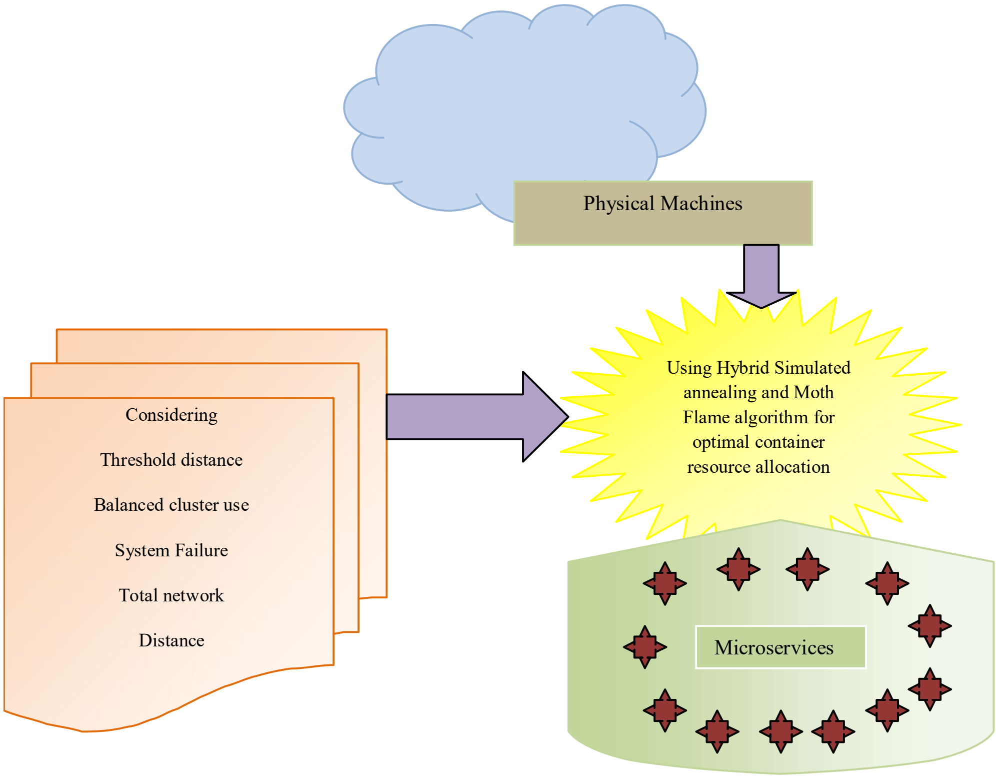


Fig. 1. *Block diagram of the proposed model for optimal container resource allocation*

3.1. Objective Function

The objective of this proposed method in cloud computing is to decide the optimal container allocation regarding a new proposed objective model that is stated in eq. (2). Eq. (2) represents the integrated modeling of four most important concerned objectives stated in equations such as eq. (3) (4) (6) and (8), correspondingly.

$$obj\ fun = \min(T_D + B_C + S_F + TN_D) \tag{2}$$

The first objective is regarding the container workload, which evades the bottleneck's availability. While the container is under the exploited circumstance, the resource utilization of the container is smaller than the resource threshold of microservice, which tends to greater scalability level in microservice. Likewise, if the microservice request count is greater, subsequently the resource utilization is greater than the resource threshold of microservice, which tends to small scalability level. Therefore, a metric is stated called threshold distance T_D and which is stated in eq. (3).

$$T_D = \sum_{\forall n_y} \left| \frac{us_x \times ms_y \times r_y}{sl_y} - t_y \right| \tag{3}$$

The next objective is regarding assisting the potential admission and novel application stipulation using balanced utilize of clusters B_C. If the physical node practice creates uniform distribution in the cluster, subsequently the cluster is appeared to be balanced. The S.D of the resource practice percentage of the physical nodes is installed to estimate the cluster balance, and it is defined in eq. (4).

$$B_C = \eta\left(PY_{usage}^{py_u} \quad ,\text{if } \exists m_y \quad \left| ac(n_y) = py_u \right.\right) \tag{4}$$

Where

$$PY_{usage}^{py_u} = \frac{\sum_{n_y} \frac{us_x \times ms_y \times r_y}{sl_y}}{cay_u} \tag{5}$$

$$\forall \qquad n_y \left| ac(n_y) = py_u \right.$$

The third objective is regarding the reliability of the application by exploiting a container's leveled distribution through the nodes of the cluster. The system's failure rate S_F is exploited to estimate the system's reliability and which is arithmetically stated in eq. (6).

$$S_F = \sum_{\forall n_y} \text{ServiceFailure}(n_y) \tag{6}$$

Where

$$\text{ServiceFailure}(n_y) =$$

$$\prod_{\forall py_u | ac(n_y) = py_u} \left(f_u + \prod_{\forall py_u | ac(n_y) = py_u} f_y \right) \tag{7}$$

The fourth objective assigns the intercommunication overhead by related microservices with a physical machine encompassing small network distances. Here, the mean value of the distance between the microservice and the containers (Total Network Distance) TN_D replica is represented and it is stated in eq. (8).

$$TN_D = \sum_{\forall n_y} \text{ServiceMean dis tan ce}(n_y) \tag{8}$$

Where

$$\text{ServiceMean Dis tan ce}(m_x) =$$

$$\frac{\sum_{\forall cn_z | cn_z \equiv n_y} \left(\sum \forall cn_{z'} \equiv n_y \left| (n_y, n_y)_{pro/con} \, dis_{ac(cn_z), ac(cn_{z'})} \right. \right)}{|cn_z| \times |cn_{z'}|} \tag{9}$$

3.2. ConventionalMoth Flame Optimization Algorithm (MFOA)

The latest stochastic population-based approaches are used called Moth Flame Optimization Algorithm (MFOA) [19] are described in this section. In nature, the most important motivation of MFO came from the navigation approach of moths. Moths are fancy insects that resemble the butterfly family. There are larger than 160,000 a variety of species of this insect in nature. In their lifetime, adult and larvae are the two major objectives. By cocoons, the larva is transformed into a moth. In the night, particular navigation approaches are for the most attractive reality regarding moths. For their navigation, they exploited a method named transverse orientation. For long traveling distances, moths fly by exploiting a fixed angle regarding the moon that is a competent method in a straight line.

Consider the candidate's solutions are moths, and the issues variables are the location of moths in the space. L represents the spiral model whereas moths move around the search space. Every moth updates their location with regarding flame by exploiting the eq. (10).

$$MF_j = L(MF_j, FL_i) \tag{10}$$

In eq. (10), MF_j denotes the j^{th} moth and FL_i is i^{th} flame.

There are other kinds of spiral models, which is used value to the subsequent rules:
a) The primary aim of the spiral must begin from the moth.
b) The last aim of the spiral must be the location of the flame.
c) The variation range of spiral mustn't go beyond the search space.

$$L(MF_j, FL_i) = Dt_j \cdot e^{bt} \cdot \cos(2\pi t) + FL_i \tag{11}$$

In eq. (11), Dt_j indicates the distance of the j^{th} moth for the i^{th} flame, t indicates an arbitrary number in [−1, 1] and b indicates a constant for major the shape of the L. Dt indicates computed by exploiting the eq. (12).

$$Dt_j = \left|FL_i - MF_j\right| \tag{12}$$

In eq. (12), MF_j indicates the j^{th} moth, FL_i states the i^{th} flame and Dt_j states the distance of the j^{th} moth to the i^{th} flame.

One more problem, the moths update their location regarding n diverse positions in the search space that can mortify the optimal shows potential solutions exploitation. Hence, the number of flames adaptively minimizes over the course of iterations by exploiting the eq. (13).

$$Flame_{num} = \text{round}\left(F_N - C * \frac{N-1}{MX}\right) \tag{13}$$

In eq. (13), MX indicates the utmost number of iterations, C indicates the current number of iterations and F_N indicates the maximum number of flames.

3.3. Conventional Simulated Annealing (SA)

The major motivation of Simulated Annealing (SA) came from the imitation modeling of a molecular movement in the materials in annealing [20].

The procedure of heating and cooling a material to recrystallize is named annealing. The particles travel in chaos and whilst the temperature reduces slowly, the particles converge to the least state of energy at high temperature.

For considering the current best solution, SA exploits probability acceptance methods. SA was effectively used to resolve the optimization problems. By exploiting the probability of acceptance of metropolis procedure, it can be simulated the annealing method to find the least energy state that is the best possible solution. The most important parameters of SA are metropolis acceptance method, primary circumstance, and cooling scheduling technique. They are explained as follows:

At first state at the start of the searching procedure of the SA method begins with initializing the initial temperature T_0 that has an important effect on the performance. The superior value T_0 will augment the computational time of the optimization procedure. In contrast, if the T_0 is too minimum, it will reason that SA will not efficiently explore the search space. Hence, it is required to be chosen cautiously to achieve the best solution.

Metropolis acceptance method is the main affecting on attaining a close to best possible solution rapidly. The metropolis rules exploited to point out whether the close by solution with minimum fitness value is established or not like the present solution. This method affects the ability of SA to get away from the local optimum. Consider F_n indicates the fitness value of the neighboring solution and F_c indicates the fitness value of the current solution. While $F(Y_0)$ is improved than $F(Y)$, subsequently SA will employ the acceptance probability method to indicate either to consider the neighbor solution as the current solution or not. The probability of acceptance method is stated as below:

$$P = e^{\left(\frac{-(F(Y)-F(Y_0))}{T_k}\right)} \tag{14}$$

In eq. (14) T_k refers to the temperature value at the time k and P is stated as the probability of acceptance.

Cooling schedule approach

In the physical annealing procedure, as the temperature minimizes, the cooling rate minimizes too to attain a stable ground state. This it's required that the system at the start to be cooled slower steadily and faster as minimized of the temperature. The cooling schedule parameter is stated in eq. (15).

$$T_{k+1} = \beta \times T_k \tag{15}$$

In eq. (15), T_k states the initial temperature value, β states the temperature coefficient and T_{k+1} states the temperature at the time k.

3.4. Adopted Hybrid SA-MFO Approach

In accordance with the MFOA, moths' locations are arbitrarily initialized. Using eq. (11), subsequently, moth swarm updates their location movement. During this, the subsequent moth location is decided. The standard description of MFO mechanically accepts the current moth location as the new moth location.

Nevertheless, SA-MFO exploits the metropolis acceptance method of SA. This method denotes whether to accept the novel moth location or not. In the case of the novel moth, the location is not accepted; one more candidate location is recomputed.

The selection is on the basis of its fitness value. By exploiting this model, a moth solution has the ability to escape from the local optima. In addition, the quality of the solution is enhanced with the greatest convergence rate. On the basis of the temperature, cooling schedule parameters and the fitness value difference of the metropolis method, the optimization procedure will explore the search space efficiently to discover an optimal solution. This procedure will be repeated until a new location is accepted exploiting metropolis acceptance rule, or the termination condition is attained.

In the initial process, SA-MFO begins with setting MFO parameters and arbitrarily initialized moths' locations within the search space. Each location indicates a solution in the search space. At each iteration, every moth location is estimated by exploiting a predefined fitness function $f(z)$. Subsequently, on the basis of this value, the metropolis rule is used to decide whether to allow or not. If the current moth accepted as the new moth, by exploiting the eq. (11) the new moths alter its location. The optimization procedure terminates whilst it attains the maximum number of iterations or whilst the optimal solution is established.

In general, Hybrid SA-MFOA is proposed in this paper. In addition, the flowchart of the proposed method is exhibited in Fig. 2.

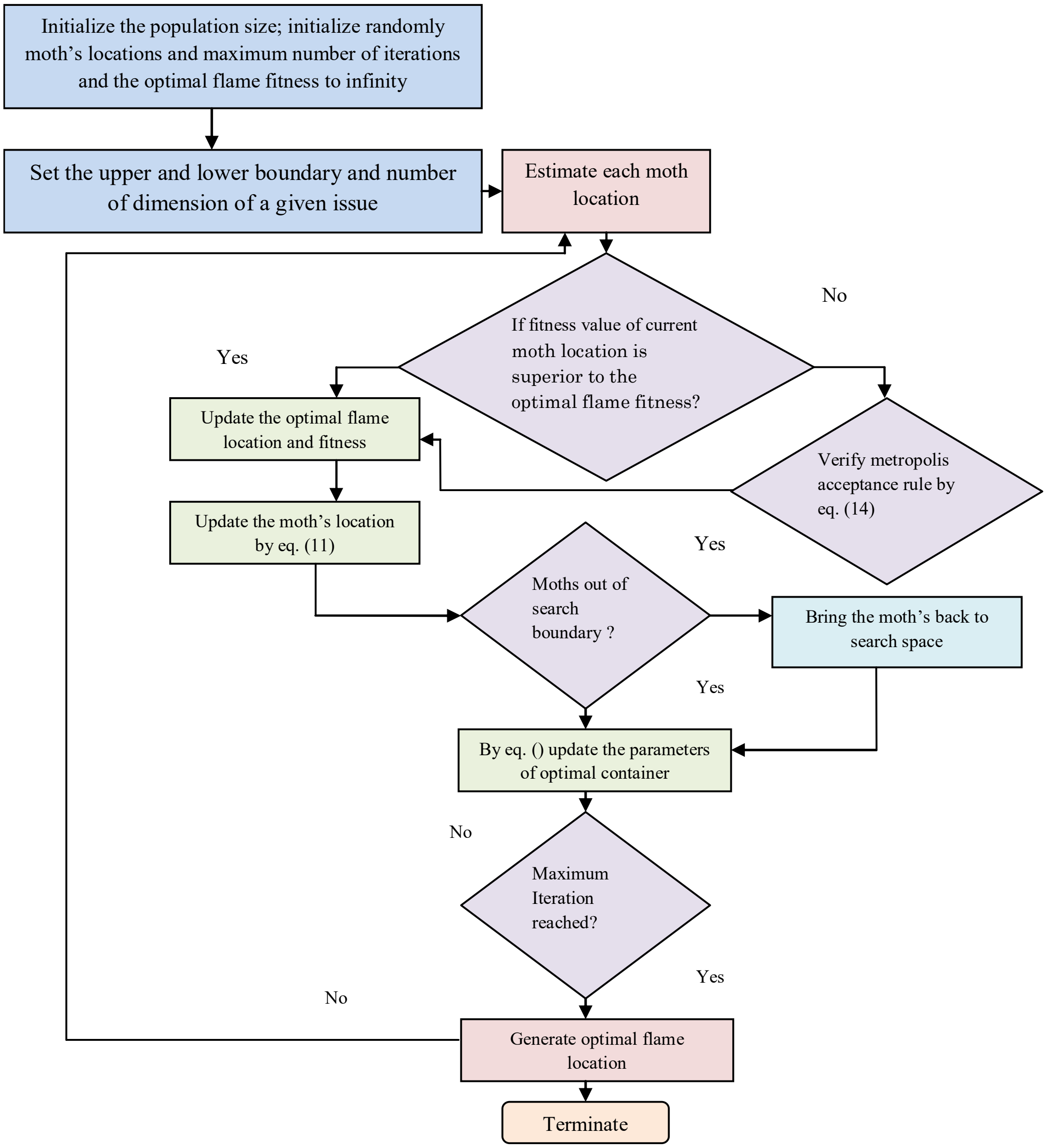

Initialize the population size; initialize randomly moth's locations and maximum number of iterations and the optimal flame fitness to infinity
Set the upper and lower boundary and number of dimension of a given issue
Estimate each moth location
If fitness value of current moth location is superior to the optimal flame fitness?
Yes
No
Update the optimal flame location and fitness
Verify metropolis acceptance rule by eq. (14)
Update the moth's location by eq. (11)
Yes
Moths out of search boundary ?
Bring the moth's back to search space
Yes
By eq. () update the parameters of optimal container
No
Maximum Iteration reached?
Yes
No
Generate optimal flame location
Terminate

Fig. 2. *Flow chart of the proposed Hybrid SA-MFOA model*

4. Results and Discussions

4.1. Experimental Setup

In this section, the experimentation of the proposed container allocation model has experimented in MATLAB 2018a. Moreover, the experimentation was evaluated by taking into consideration the heterogeneous clusters using 250 machines with capacity 100. The performance of the proposed model was evaluated over other traditional techniques such as Genetic Algorithm, Lion Algorithm, and Whale Optimization Algorithm, and the results, which were discussed regarding the modeled cost function.

4.2. Performance Analysis

Here, the performance analysis of the proposed and existing models with respect to the total cost, first, second, third, and fourth objective models is demonstrated for machines with capacity 100. Fig 3 demonstrates the analysis of the proposed technique with the conventional techniques regarding the total cost. For the best case scenario, the proposed method is 12% better than GA, 23% better than LA and 12% better than WOA algorithm. Fig 4 demonstrates the analysis of the proposed technique with the traditional techniques regarding the first objective function. In Fig 5, the analysis of the proposed technique with the traditional techniques regarding the second objective function is shown. For the best case scenario, the proposed method is 22% better than GA, 24% better than LA and 15% better than WOA algorithm. Fig 6 exhibits the analysis of the proposed technique with the traditional techniques regarding the third objective model. In the best case scenario, the proposed method is 15% better than GA, 16% better than LA and 18% better than WOA algorithm. In Fig 7, the analysis of the proposed method with the traditional methods regarding the fourth objective function is shown. For the best case scenario, the proposed method is 32% better than GA, 34% better than LA and 35% better than WOA algorithm. Here, the overall analysis shows the performance of the proposed technique is better than the traditional techniques.

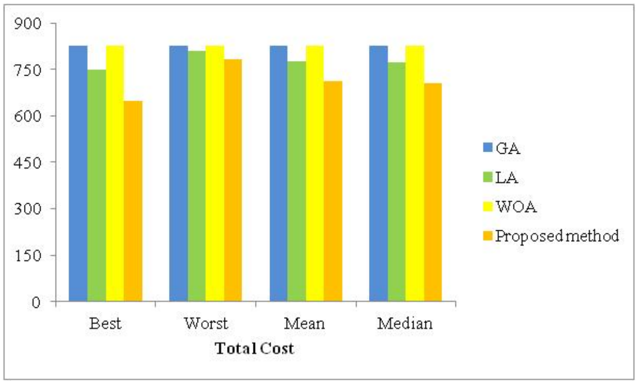


Fig. 3. *Analysis of the proposed and existing models regarding the Total cost*

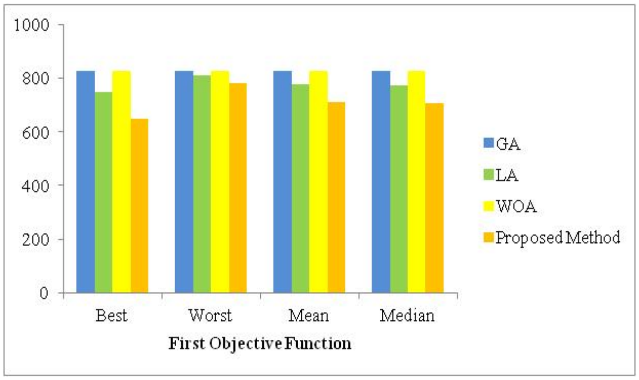


Fig. 4. *Analysis of the proposed and existing models regarding the first objective function*

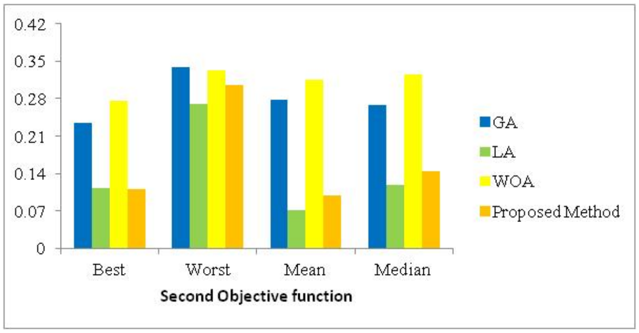


Fig. 5. *Analysis of the proposed and existing models regarding the second objective function*

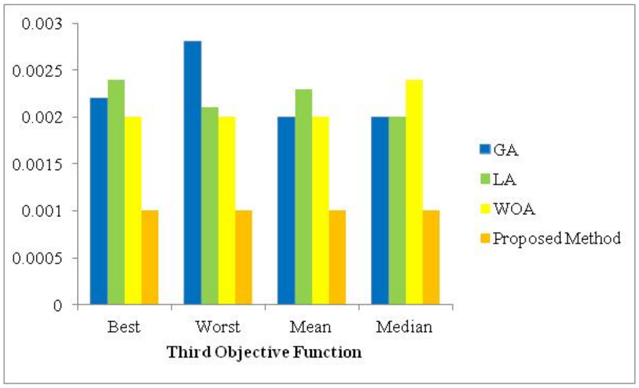


Fig. 6. *Analysis of the proposed and existing models regarding the third objective function*

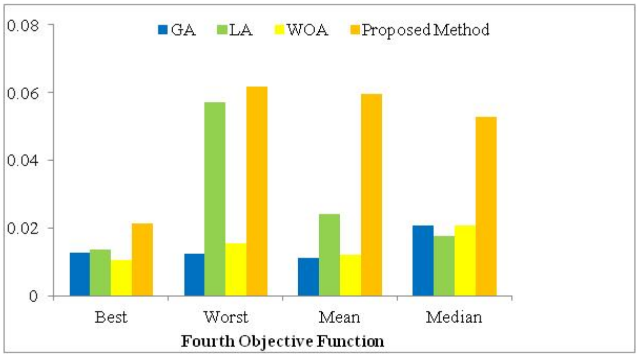


Fig. 7. *Performance analysis of the proposed and existing models regarding the fourth objective function*

5. Conclusion

In this paper, a novel optimized container resource allocation approach has experimented with respect to the new optimization approach. To carry out the optimal container resource allocation procedure, a novel approach was proposed called the Hybrid SA-MFOA algorithm that was the hybridization of SA and MFOA. The proposed model was entirely modeled on considering some objective functions like total network distance, balanced cluster use, threshold distance, and system failure. At last, the performance of the proposed approach was analyzed and evaluated over other methods, and shown the enhancement of the proposed model. From the results, it was analyzed that the performance of the proposed method for experiments has gained a superior solution with reduced cost than LA, GA, and WOA correspondingly.

Compliance with Ethical Standards

Conflicts of interest: Authors declared that they have no conflict of interest.

Human participants: The conducted research follows the ethical standards and the authors ensured that they have not conducted any studies with human participants or animals.

References

[1] Chunlin Li, Hezhi Sun, Hengliang Tang, Youlong Luo, "Adaptive resource allocation based on the billing granularity in edge-cloud architecture", Computer Communications, Volume 145, September 2019, Pages 29-42.

[2] Sadip Midya, Asmita Roy, Koushik Majumder, Santanu Phadikar, "Multi-objective optimization technique for resource allocation and task scheduling in vehicular cloud architecture: A hybrid adaptive nature inspired approach", Journal of Network and Computer Applications, Volume 103, 1 February 2018, Pages 58-84.

[3] K. P. N. Jayasena, Lin Li, Qing Xie, "Multi-modal Multimedia Big Data Analyzing Architecture and Resource Allocation on Cloud Platform", Neurocomputing, Volume 253, 30 August 2017, Pages 135-143.

[4] Y. Liu, F. R. Yu, X. Li, H. Ji and V. C. M. Leung, "Distributed Resource Allocation and Computation Offloading in Fog and Cloud Networks With Non-Orthogonal Multiple Access," IEEE Transactions on Vehicular Technology, vol. 67, no. 12, pp. 12137-12151, Dec. 2018.

[5] G. Peng, H. Wang, J. Dong and H. Zhang, "Knowledge-Based Resource Allocation for Collaborative Simulation Development in a Multi-Tenant Cloud Computing Environment," IEEE Transactions on Services Computing, vol. 11, no. 2, pp. 306-317, 1 March-April 2018.

[6] A. Younis, T. X. Tran and D. Pompili, "Bandwidth and Energy-Aware Resource Allocation for Cloud Radio Access Networks," IEEE Transactions on Wireless Communications, vol. 17, no. 10, pp. 6487-6500, Oct. 2018.

[7] F. Lin, Y. Zhou, G. Pau and M. Collotta, "Optimization-Oriented Resource Allocation Management for Vehicular Fog Computing," IEEE Access, vol. 6, pp. 69294-69303, 2018.

[8] A. Jin, W. Song and W. Zhuang, "Auction-Based Resource Allocation for Sharing Cloudlets in Mobile Cloud Computing," IEEE Transactions on Emerging Topics in Computing, vol. 6, no. 1, pp. 45-57, Jan.-March 2018.

[9] C. Mouradian, D. Naboulsi, S. Yangui, et al., A comprehensive survey on fog computing: State-of-the-art and research challenges, IEEE Commun. Surv. Tutor. 20 (1) (2018) 416–464.

[10] T. Jiang, Z. Wang, Z. Chen, et al., An adaptive strategy of online session migration for streaming media edge cloud, in: 2016 35th Chinese Control Conference (CCC), Chengdu, 2016, pp. 5278–5283.

[11] K. Sasaki, N. Suzuki, S. Makido, et al., Vehicle control system coordinated between cloud and mobile edge computing, in: 2016 55th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), Tsukuba, 2016, pp. 1122–1127.

[12] Y. Zhang, K. Liang, S. Zhang, et al., Applications of edge computing in piot, in: 2017 IEEE Conference on Energy Internet and Energy System Integration (EI2), Beijing, 2017, pp. 1–4.

[13] Chunlin Li, Zhu Liye, Tang Hengliang, Youlong Luo, Mobile user behavior based topology formation and optimization in ad hoc mobile cloud, J. Syst. Softw. 148(2019) 132–147.

[14] M. Lin, J. Xi, W. Bai and J. Wu, "Ant Colony Algorithm for Multi-Objective Optimization of Container-Based Microservice Scheduling in Cloud," IEEE Access, vol. 7, pp. 83088-83100, 2019.

[15] I. Filip, F. Pop, C. Serbanescu and C. Choi, "Microservices Scheduling Model Over Heterogeneous Cloud-Edge Environments As Support for IoT Applications," IEEE Internet of Things Journal, vol. 5, no. 4, pp. 2672-2681, Aug. 2018.

[16] M. Mena, A. Corral, L. Iribarne and J. Criado, "A Progressive Web Application Based on Microservices Combining Geospatial Data and the Internet of Things," in IEEE Access, vol. 7, pp. 104577-104590, 2019.

[17] W. Dai et al., "Semantic Integration of Plug-and-Play Software Components for Industrial Edges Based on Microservices," IEEE Access, vol. 7, pp. 125882-125892, 2019.

[18] K. Huang and Y. Hsieh, "Very fast simulated annealing for pattern detection and seismic applications," 2011 IEEE International Geoscience and Remote Sensing Symposium, Vancouver, BC, 2011, pp. 499-502.

[19] X. Zhao, Y. Fang, Z. Ma and M. Xu, "An Ameliorated Moth-Flame Optimization Algorithm," 2018 37th Chinese Control Conference (CCC), Wuhan, 2018, pp. 2372-2377.

[20] F. A. L. Ferreira and F. A. B. Lemos, "Unbalanced electrical distribution network reconfiguration using simulated anneling," 2010 IEEE/PES Transmission and Distribution Conference and Exposition: Latin America (T&D-LA), Sao Paulo, 2010, pp. 732-737.