BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **20**, No 1
Sofia • 2020 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2020-0003

A Hybrid Technique for Server Consolidation in Cloud Computing Environment

C. Vijaya[1], *P. Srinivasan*[2]

[1]*School of Computer Science and Engineering, VIT University, Vellore 632014, India*
[2]*School of Information Technology, VIT University, Vellore 632014, India*
E-mails: c.vijaya2016@vitstudent.ac.in srinivasan.suriya@vit.ac.in

Abstract**:** *The goal of data centers in the cloud computing environment is to provision the workloads and the computing resources as demanded by the users without the intervention of the providers. To achieve this, virtualization based server consolidation acts as a vital part in virtual machine placement process. Consolidating the Virtual Machines* (*VMs*) *on the Physical Machines* (*PMs*) *cuts down the unused physical servers, decreasing the energy consumption, while keeping the constraints for CPU and memory utilization. This technique also reduces the resource wastage and optimizes the available resources efficiently. Ant Colony Optimization* (*ACO*) *that is a well-known multi objective heuristic algorithm and Grey Wolf Algorithm* (*GWO*) *has been used to consolidate the servers used in the virtual machine placement problem. The proposed Fuzzy HAGA algorithm outperforms the other algorithms MMAS, ACS, FFD and Fuzzy ACS compared against it as the number of processors and memory utilization are lesser than these algorithms.*

Keywords**:** *Virtual Machine* (*VM*) *placement, VM, power consumption, resource wastage, Ant Colony Optimization* (*ACO*)*, Grey Wolf Optimisation* (*GWO*) *Algorithm, HAGA Algorithm.*

1. Introduction

In data centers, a number of heterogeneous workloads would run on servers at different times. The workloads, which are not similar, could be normally classified into two divisions, namely chatty workloads and non-interactive workloads. Chatty workloads can become aggressive at some point and return to rest at some other point. A web video service is an example of chatty workloads because more number of people work at nighttime and few people during the daytime. Non-interactive workload does not require people's interaction to make progress after they are submitted. High performance computing is the best example of non-interactive workload. The requirements of resources of these workloads are dramatically different at different times. To make sure that the workload will always match with the requirement, the workload is allocated in static mode so that the highest demand will also be served. Fig. 1 illustrates server virtualization in data centers. In datacenters, the resource optimization is targeted on CPU, Memory and network interfaces.

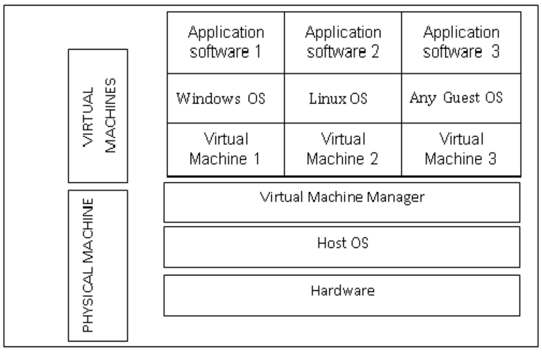

Fig. 1. Server virtualization in the Datacenter

Since, it is usual to see that many of the servers in datacenters are underutilized. A large amount of resources like hardware, space, power and management costs of the servers are getting wasted also causing environmental pollution to the surroundings. By consolidating the server, the utility of those resources should be enhanced by the way of decreasing the quantity of the active servers. There is a variety of consolidation techniques such as centralized and physical consolidation. But consolidation based on server virtualization seems to be the best one in data centers. The resource management is one of the important issues to be taken seriously in order to optimize the resources. The functionalities are performed considering the full server, in which the resource management will be better. Server virtualization enables a part of the Physical Machine (PM) to be allocated instead of the whole. But this kind of allocating part of the physical machine to the requested Virtual Machines (VM) will increase the complexity of the resource management. So, it is considered as a big issue of improving the utilization of resources while guaranteeing the Quality of service to the end users. Here, server consolidation is an approach that improves availability and business continuity. The total cost of ownership is reduced. It also extends the lifetime of servers where there will be no need to purchase the servers often. Server consolidation also reduces the area of the data center, reducing the maintenance costs as well as power consumption thereby cutting down the cooling and wiring resources.

When a group of n virtual machines should be mapped to a group of m physical machines, m^n number of VM placement solutions are possible here. Therefore, it is difficult to find the best solution amongst so many numbers of feasible solutions. The fuzzy HAGA Algorithm efficiently finds out a good solution in such a large space of solutions and thereby solves the problem of Virtual Machine Placement (VMP).

The Paper is organized in such a way that Section 2 discusses about the Literature review, and Section 3 speaks about the general discussions on Evolutionary mutiobjective algorithms. Section 4 explains what is server consolidation problem and the objective function of the server consolidation problem using resource wastage modeling. Section 5 explains about the ACO Algorithm and GWO Algorithms which act as the base of the newly proposed Fuzzy HAGA Optimization Algorithm. Section 6 explains in detail about the proposed Fuzzy HAGA Optimization Algorithm and the results were also discussed in the same.

2. Literature review

Many of the researchers have taken multiple objectives for virtual machine replacement. Chamas, Lopez-Pires and Baran [1] have considered four objectives such as power consumption, economical revenue, resource utilization and reconfigurable time by implementing two phases such as online incremental VMP phase and offline reconfiguration phase. Uddin, and Rehman [2] introduced a new technique to categorize the servers for server consolidation and saved a huge amount of energy. Some of the methods employed for consolidation problem are FFD, Best Fit, Best Fit Decreasing and other methods. Many heuristic techniques also have been suggested by most of the researchers. Ali and Lee [3], proposed a novel heuristic of geography based optimization of VM placement which shows how swarm intelligence is used in solving optimization. Panigrahy et al. [4] in their research paper have proved that several combinations multidimensional vectors can result in scalar size and this technique is used in the VM placement. A group genetic algorithm is proposed by Agrawal, Bose and Sundarrajan [5], where they have solved the VM placement as vector packing problem. Ali et al. [6], in their work have designed an approach to meet the constraints imposed by the users which is energy efficient by reducing the number of physical servers.

Liu et al. [7], used the ACS based approach which is coupled with order exchange and migration technique to bring out a new algorithm for VM placement. It has been an effective approach compared to similar works. S. Dorterler, M. Dorterler and S. Ozdemir [8] have analyzed four multiobjective evolutionary algorithms for optimizing CPU utilization and reducing energy consumption. They have compared the heuristics of the four papers using cloudsim simulator and planet lab dataset.

Sotomayor [9] introduced a lease-based model, FirstCome-FirstServe (FCFS) and back-filling algorithms for scheduling of three kinds of jobs classified into best-effort, immediate and advanced reservation jobs. To improve the performance, the scheduling algorithms always choose free servers (i.e., with minimum load) then allocate a new VM where energy efficiency is not at all considered Kansal and Chana [10]. The decisions which are energy aware are taken by analyzing past resource utilization and energy consumption details. It moves a heavily loaded VM from a physical machine satisfying the minimum criteria for power consumption, to another physical machine which consumes least energy. 49.39% of energy has been saved by using FF-EVMM Algorithm over the baseline algorithms taken.

Zhou et al. [11] improved the reliability of cloud services by virtual machine optimization approach using three algorithms using maximum weight matching in bipartite graphs. Fu and Zhou [12] have introduced VM placement that predicts the accurate estimation of the use of resources. By predicting the affinity between the host and the VMs the placement has been done by placing the VM which has high affinity towards a PM. PAVMP Algorithm has been developed to reduce the VM migrations and SLA violations thereby reducing the energy consumption works better then Power Aware Best fit algorithm. Khosravi, Andrew and Rajkumar Buyya [13] proposed a dynamic VM placement algorithm to approximate the cost based on different constraints, considering four data centers. They have considered access to renewable energy sources apart from off-site grid sources. The result proves to save the energy cost by 10.03 % when compared to its competitive algorithm CRA-DP. Cloud service reliability has been enhanced by placing *i*-th virtual machine on to the optimal host by reducing the consumption of network resources thereby reducing the energy cost. Gao et al. [14] have used dominance based Multi Objective Evolutionary Algorithm to find the optimal server for a VM.

Dilip Kumar and Dr. Tarni Mandal [15] used an hybrid Genetic algorithm and Particle Swarm Optimization (PSO) based Algorithm for bi objective VM placement. B. B. Jacinth et al. has proposed ACO-PSO optimization, which lessen the power consumption and resource wastage through load balancing, also providing fault tolerance. Zhang et al. [16] have suggested a clustering based 2 approximation algorithm to decrease the distance between data centers that minimizes communication latency and increases availability. Barlaskar et al. [17] made a decision to place the VMs based of gilt edged solution of stochastic integer programming.

Tawfeek et al. [18] has cut down the wastage of memory and processor. They have shown a method to apply the ACO Algorithm in order to find out a good solution in a large space. The proposed algorithm is designed where multi-objective VM placement is required. Alboaneen, Tianfield and Glasgow [19] have designed a meta-heuristic optimization algorithm based on glow-worm swarm to solve energy and SLA aware VMP problem. Speitkamp and Bichler [20] have discussed about server consolidation using the linear programming formulations of problems.

Sait, Bala and El-Maleh [21] have given a fitness function for cuckoo search based resource optimization for VM placement. It finds a better placement for incoming and existing VMs using fuzzy rules. Riquelme, Lucken and Baran [22] have narrated 54 performance metrics for multiobjective optimization. Wu et al. [23] considered both the energy consumption in servers and bandwidth, through Genetic Algorithm. Zhao et al. [24] have developed two algorithms for VM placement with VM migration, namely Harmonic Algorithm and DDG Algorithm. DDG Algorithm is applicable when penalty based on SLA is high while job arrival rate is lesser and migration cost too is lesser. Harmonic Algorithm works well than First fit when the arriving rate of the job is high and VM placement without VM migration is more than 1.0.

Shabeera et al. [25] selected the PMs in close proximity and jobs executed on allocated VMs works better. Gagwero and Caviglione [26] considered the reduced effects of churn, mitigated collocation interference, minimize power consumption and enforced security requirements through MPC placement algorithm, which performs better than classic heuristics in datacenters. Jing et al. [27] investigated a similarity between CPU utilization and power consumption to build a nonlinear power model where bi-objective optimization was done by improving the performance of VM and reducing the power consumption of server based on ACO.

Hung et al. [28] have solved a static virtual machine allocation problem which is applicable in universities and researching by taking one lab hours of one working day in a university. Sarvesh Kumar [29] in his paper has found that Meta Heuristic Algorithm which is population based on laws of gravity and motion for solving non-linear problems named as Gravitational Search Algorithm. Fitness calculation is based on number of physical machines utilized. The results were compared against FFD, LL (Least Loaded) and ACO Algorithm. The results showed that it outperforms all other methods.

Boominathan, Aramudan and Saravanaguru [30] in his work has applied fuzzy hybrid bio inspired technique to solve the VM placement problem through server consolidation technique. Fuzzy rules were generated to choose the next VMs for the current server. Cuckoo search is applied to find the new optimal solution. So by combining ACS and cuckoo search, they have developed an algorithm and by combining ACS and firefly colony, they have developed another algorithm. Both of them have proved to give the best results when compared with the similar algorithms like firefly colony, ACS, MMAS, and FFD. S. Mirjalili, S. M. Mirjalili and A. Lewis [31] proposed Grey wolf optimization algorithm followed by the attacking nature of the prey by wolves. Joshi and Bansal [32] in his research has combined grey wolf optimization which is a population based optimization technique with gravitational search to find a new optimization technique which proved to give better result when compared against its base algorithms like gravitational search optimization and grey wolf optimization algorithms.

In the proposed work, we have planned to combine ACS which is a well known optimization technique and Grey Wolf Optimization (GWO) Algorithm for a better server consolidation and it is proven that the proposed fuzzy HAGA algorithm saves the power consumption and also reduces the number of active servers compared against the other algorithms taken for comparison.

3. Evolutionary multiobjective optimization

Evolutionary multi-objective algorithms generally use a population based approach while finding out a Pareto optimal solution. Pareto optimal solutions are the solutions that may be improved not simply in one objective function, but more than one objective functions should be optimized simultaneously. The performance will certainly get affected in any other one of the rest of the functions. Most of the algorithms use dominance concept during the selection process. A multiobjective minimization problem could be stated as

$$\text{Minimize } \overrightarrow{f_{\mathrm{o}}}(\overrightarrow{x_{\mathrm{d}}}) = \left[f_1\left(x_{\mathrm{d}_1} \dots x_{\mathrm{d}_m}\right), \dots, f_n\left(x_{\mathrm{o}_1} \dots x_{\mathrm{o}_n}\right)\right],$$

where

$$\overrightarrow{x_{\mathrm{d}}} = \left(x_{\mathrm{d}_1} \dots x_{\mathrm{d}_m}\right) \in X,$$
$$\overrightarrow{f_{\mathrm{o}}} = \left(x_{\mathrm{o}_1} \dots x_{\mathrm{o}_n}\right) \in Y,$$

m and *n* are decision variables and objectives, respectively, $\overrightarrow{x_{\mathrm{d}}}$ denotes the decision vector, where *X* denotes the variable space and $\overrightarrow{f_{\mathrm{o}}}$ is the objective vector, *Y* denotes the objective area.

The dominating points are those in which the decision vector $\vec{x}_1$ has better objective than any other decision vector. Find all the non-dominated solution set [multi objective]. Start with first decision variable. Compare first variable with all other remaining variables for domination. Mark the dominating solution and all the solution except the marked one are non-dominated solutions

Consider we have some *n* number of physical machines running applications on them. In case assume that, all the applications need VMs to be executed. So mapping a VM to a PM is a multidimensional vector packing problem. Various resource utilizations represent the various dimensions. Let us take for example, a request for a VM containing 20% CPU and 30% memory and another request of VM having 35% and 40%.Then the usage of server shall be calculated as 55% and 70%. We need to impose a boundary say, 90% which is less than 100% utilization in order to avoid the performance degradation of the server otherwise, which may lead to migration of VMs.

4. Server consolidation

In an organization, it is essential to reduce the number of servers that it requires in order to reduce the server sprawl which means that the underutilized servers consume more space, resources and power. So, server consolidation is widely used to cut down the energy consumption.

The main objective of the server consolidation problem is to reduce the number of active server machines needed for placing the virtual machines which are requested by the users. Consider *n* VMs to be placed on *m* Physical Machines (PMs). No VM is larger than any PM in capacity. D_{proc} represents the CPU requirement of each VM and D_{memory} represents the memory demand of a single VM. Let T_{proc} and T_{memory} represent the full capacity of a single physical machine. The proposed algorithm is going to reduce the wastage of resources and wastage of CPU. When the physical machine is utilized fully, then the performance of the physical machine may get degraded. So we have an upper bound of 90% (fuzzy). We define two decision variables allocation matrix and binary variables. If a VM is allocated to a particular server *j*, then allocation matrix is set to 1 or it is set to 0, otherwise. A binary variable is used to denote whether a server is busy or not.

The balance resources on each physical machine may vary based on the virtual machine placement algorithm. For all the resources to be utilized efficiently, the total cost of the wasted resources should be calculated thoroughly.

4.1. Server resource wastage modeling

There are many VM placement solutions which vary in the number of resources remaining on each server. Multidimensional resources should be completely utilized. Therefore the cost of wasted resource is evaluated by the below equation:

(1) $$\text{Waste}_j = \frac{\left|L_j^{\text{proc}} - L_j^{\text{memory}}\right| + \Delta}{\left|U_j^{\text{proc}} - U_j^{\text{memory}}\right|}.$$

Waste$_j$ represents the resource wastage in j-th server. $U_j^{\text{proc}} - U_j^{\text{memory}}$ represents the CPU and memory usage normalized in a physical machine. It is the ratio between used resources to the total resources available. $L_j^{\text{proc}} - L_j^{\text{memory}}$ represents remaining resources in terms of CPU and memory. Δ is a small positive integer to avoid the capacity of the physical machine coming down to zero and it is set to 0.0001.

4.2. Objective function of server consolidation

The objective function is designed to reduce the quantity of physical machines as well as not violating the capacity of the physical server. Consider that we are assigned n number of VMs where ($i \in I$). Here VMs are applications which are need to be assigned to m servers ($j \in J$). VM_{ij} and PM_j are the two binary variables used to represent the placement of VM. The first binary variable VM_{ij} denotes if VM_i is assigned to server$_j$ and the binary variable PM_j represents if server is active. Minimizing the power consumption while minimizing the resource wastage should be done which were taken as our objective for virtual Machine Placement. Therefore, the VMP problem can be mathematically formulated as

(2) $$\text{Minimize } \sum_{j=1}^{m} \text{PM}_j.$$

The limits on constraints are:

(3) $$\sum_{j=1}^{m} \text{all}_{ij} = 1,\ i \in I,$$

(4) $$\sum_{i=1}^{n} D_{\text{proc}}^{i} \text{all}_{ij} \leq T_{\text{proc}}^{j} \text{PM}_j,\ j \in J,$$

(5) $$\sum_{i=1}^{n} D_{\text{memory}}^{i} \text{all}_{ij} \leq T_{\text{memory}}^{j} \text{PM}_j,\ j \in J,$$

(6) $$\text{PM}_{j,}\ \text{all}_{ij} \in \{0, 1\},\ i \in I, \text{ and } j \in J.$$

According to constraint (3), one VM is allocated to only one PM. Constraint (4) and (5) is related to the capacity constraints of the PMs. The binary decision variables states whether a server is active or not as shown in (6).

5. Basics of ACO Algorithm and GWO Algorithm

5.1. Ant colony optimization

Ant colony optimization is an optimization technique learnt from the behavior of the ants in searching their food by finding the nearest path from their habitat to the food source. Ants find their path by choosing random decision taken by the amount of Pheromone (a kind of saliva like substance) secreted by the ants on the way to the food source. The information passed by the other ants also is used to find the optimal solution. The information to assign VM_i to PM_j is

$$\eta_{ij} = \frac{|D^i_{\text{proc}} + D^i_{\text{memory}}|}{|L^i_{\text{proc}} + L^i_{\text{memory}}| + \epsilon}. \tag{7}$$

The Pheromone trail update:

$$\tau_{ij} = \begin{cases} \frac{\sum_{u\in\Omega_{k(j)}} \tau_{ui}}{|\Omega_{k(j)}|} & \text{if } \Omega_{k(j)} - \{i\} \neq 0\,, \\ 1 & \text{otherwise.} \end{cases} \tag{8}$$

The solution is constructed using Pseudo random proportional rule as given in the next equation:

$$I = \begin{cases} \operatorname{argmax}_{u\in\Omega_{k(j)}}\{\propto_{\text{p}} \times \tau_{uj}(-\propto_{\text{p}}) \times \eta_{uj}\} & \text{if } c < c_0, \\ \text{explore } b & \text{otherwise,} \end{cases} \tag{9}$$

where c is a probabilistic parameter, which is distributed uniformly in the range of [0, 1]; c_0 is a variable which has value in the range of 0 and 1. If c is lesser than or equal to c_0, then exploitation process takes place, and if it is greater than c_0, it is called exploration of new solutions. $\propto_p$ is the variable through which the user can control the pheromone trail. Using the roulette wheel selection method we select the random variable b, and using the proportional rule random probability distribution [30]:

$$P^k_{i,j} = \frac{\propto_p \times \tau_{uj} + (1-\propto_p)\times\eta_j}{\sum_{u\in\Omega_{k(j)}}(\propto_p \times \tau_{ij} + (1-\propto_p)\times\eta_j}, \; i\in \Omega_{k(j)}, \tag{10}$$

$$\Omega_{k(j)} = \begin{cases} i \in \{1,\dots,n\}(\sum_{u=1}^{m} \text{all}_{iu} = 0)\wedge\left(\left(\sum_{u=1}^{n}\left(\text{all}_{uj} \times D^{\,i}_{\text{proc}}\right) + D^{\,i}_{\text{proc}}\right) \leq T^{\,j}_{\text{proc}}\right)\wedge \\ \wedge\left(\left(\sum_{u=1}^{n}\left(\text{all}_{uj} \times D^{\,i}_{\text{memory}}\right) + D^{\,i}_{\text{memory}}\right) \leq T^{\,j}_{\text{memory}}\right). \end{cases} \tag{11}$$

The local updating of the Pheromone is done using the following relation:

$$\tau_{ij} = (1-\varphi_g)\tau_{ij}(t-1) + \varphi_1.\tau_0, \tag{12}$$

Here, the pheromone decay coefficient is equal to $\varphi_1 \in \{0, 1\}$, and τ_0 indicates the initial value of the pheromone.

Fitness function of the derived solutions is evaluated using the cost function designed by M. Sadique and other authors according to the fitness of the VM, which will be packed in the PM as given below:

$$\frac{D_i^{\text{proc}} + D_i^{\text{memory}}}{(T_i^{\text{proc}} - \sum_{k=1,k\neq i}^{n} D_k^{\text{proc}}) + (T_i^{\text{memory}} - \sum_{k=1,k\neq i}^{n} D_k^{\text{memory}})}, \tag{13}$$

based on best solution, the global pheromone update is given by

$$\tau_{ij}(t) = (1-\varphi_g)\tau_{ij}\tau_{ij}(t-1) + \varphi\delta\tau_{ij}^{\text{best}}.$$

Here $\varphi_g \in \{0, 1\}$ represents the evaporation rate and

$$\delta\tau_{ij}^{\text{best}} = \begin{cases} f_{sn}(S^{gb}) & \text{if } \text{VM}_i \text{ is placed in server } j, \\ 0 & \text{otherwwise}\,, \end{cases} \tag{14}$$

where f_{sn} indicates the fitness of the solution which is identified by finding out the average fitness of already placed VMs. This average fitness of VMs are computed by the VM fitness equation given in (13).

5.2. Grey Wolf Algorithm [31]

Inspired by the behavior of grey wolves attacking on the prey in groups, the algorithm has been designed to find the optimal solution. Assume the first fittest solution are

$\propto$ wolves and next consecutive fittest are β and δ and the remaining solutions are ω wolves. The order of the fittest in the hunting process is shown in the Fig. (2).

The important phases of Grey wolf hunting process are as given in [34]:

1. "Tracking, chasing and approaching the prey.
2. Pursuing, encircling and harassing till it stops its movement.
3. Finally, making an attack on the prey."

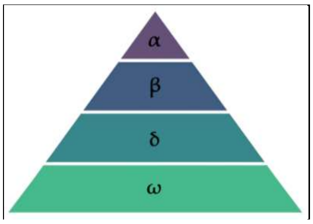

Fig. 2. Hierarchy of Grey wolves in the process of hunting

The mathematical notation of encircling is given as

(15) $$D=\left|C_V X_p^v(t) - X_G^v(t)\right|,$$

(16) $$X_G^v(t+1) = X_p^v(\mathrm{t}) - A_V D,$$

t is the iteration; A_V and C_V indicates the coefficient vectors; X_p^v gives the position vector of the prey and $X_G^v(t)$ indicates the position vector of the Grey wolf.

The calculation of the values of vectors A_V and C_V are:

(17) $$A_V = 2br_{v_1} - b,$$

(18) $$C_V = 2r_{v_2},$$

where the values of b are linearly decreased from 2 to 0 during the process of iteration and r_{v_1}, r_{v_2} are any random vectors between the values [0, 1].

5.2.1. Hunting

The best search agents, Alpha, Beta, Gamma and all the other Omega agents are updated using the following equations:

(19) $$D_\alpha = \left|C_{v1} . X_{G\alpha}^v - X_G^v(t)\right|,$$

(20) $$D_\beta = \left|C_{v2} . X_{G\beta}^v - X_G^v(t)\right|,$$

(21) $$D_\delta = \left|C_{v3} . X_{G\delta}^v - X_G^v(t)\right|,$$

(22) $$X_{G1}^v = X_{G\alpha}^v - A_{v1} . (D_\alpha),$$

(23) $$X_{G2}^v = X_{G\beta}^v - A_{v1} . (D_\beta),$$

(24) $$X_{G3}^v = X_{G\delta}^v - A_{v1} . (D_\delta),$$

(25) $$X_G^v(t+1) = \frac{X_{G1}^v + X_{G2}^v + X_{G3}^v}{3}.$$

5.2.2. Attacking the prey

The value of b is decreased to show that the agents goes near the prey. The variations in the value of A also depends on b.When random values lies in between [–1, 1], the next position of the search agent can be located in any place from its current position and the prey's position when $|A_v|$ <1, it attacks the prey, it means exploitation and if it is greater than 1 means exploration.

5.2.3. Searching for prey

Wolves generally go apart for searching the prey and they come towards the prey to attack it. This is called as divergence and convergence respectively. A_v is applied with random numbers which are in the range between 1 and –1 so as to force the search agent to get away from the prey. When $|A_v|$ >1, it diverges from the prey to get a better prey.

5.2.4. Dynamic update of parameter b

As we see that the parameter b directly affects A_v, and (1) and (2) gets affected because of b. When the coefficient A_v's values are in the range [–1, 1], the exploitation process is been simulated. The next position of the wolves can be anywhere between the present position and the prey. Therefore, b plays a vital role in exploitation as well as exploration.

The ranges for the parameter b are b [0, 2], b [0, 3], and b [0, 0.5]. A fuzzy Inference System rules are taken from [34] to increase and decrease b; C_v is also taken as important as b in decision making of the movement of the wolves. This C_v parameter catalyzes exploration process. It does not linearly decrease but decreases randomly.

GWO saves the best solution so far over several iterations. The parameters A_v and C_v help the solutions generated to find out where is the prey located. Exploration process and exploitation process are done using the values of b and A_v. There is no direct relation between the search agents and fitness function. The fitness value of the i-th agent is denoted by $\text{fit}(i)$. The formula for calculating the fitness is given in (25). A set of fuzzy rules were used for the variables b and C_v. The rules are taken from [34] and used to increase or decrease the values of b. GWO Algorithm is used for exploring the solutions obtained by the ACO.

GWO Algorithm [35]

Table 1. GWO Algorithm

```
Initialize the grey wolf population X(i) = (1, 2, ..., n)
Initialize b, A_v and C_v
Calculate the fitness of each search agent α, β, δ
Loop:
While t< maximum number of iterations
   For each search agent
       update the position of the current search agent by (19)-(25).
   Endfor
  Update X_α, X_β and X_δ
Increment t
End while
Return X_α and X_β
Add the non dominated sollutions to the archive
repeat the loop until number of iterations
```

6. Proposed hybrid optimization

6.1. Fuzzy HAGA optimization

Now, we should decide about the next VM_i to be placed in the currently chosen server as given in [29].

"If β_{ij} is low and η_{ij} is low then the efficacy e_{ij} of choosing VM_i is very very low.
If β_{ij} is medium and η_{ij} is low then the efficacy e_{ij} of choosing VM_i is very low.
If β_{ij} is high and η_{ij} is low then the efficacy e_{ij} of choosing VM_i is low.
If β_{ij} is low and η_{ij} is medium then the efficacy e_{ij} of choosing VM_i is low.
If β_{ij} is medium and η_{ij} is medium then the efficacy e_{ij} of choosing VM_i is medium.
If β_{ij} is high and η_{ij} is medium then the efficacy e_{ij} of choosing VM_i is high.
If β_{ij} is low and η_{ij} is high then the efficacy e_{ij} of choosing VM_i is high.
If β_{ij} is medium and η_{ij} is high then the efficacy e_{ij} of choosing VM_i is very high.
If β_{ij} is high and η_{ij} is high then the efficacy e_{ij} of choosing VM_i is very high."

This Fuzzy HAGA technique uses the minimal and max-min operations in implication and composition respectively. Subsequently, the maximum efficiency e_{ij}^{k} is obtained for every virtual machine i. The rule given below has been used which is stated in (12) to decide which VM_i should be assigned for an individual server j,

$$I=\begin{cases} \text{Fuzzy strategy}, q \le q_0, \\ \text{Fuzzy probable strategy}, q > q_0. \end{cases}$$

The fuzzy probable strategy is derived from (25), as given in the GWO Algorithm. In the output, we will get the number of virtual machine to be placed in that corresponding server.

To implement the exploitation process, a fuzzy technique is used as in [29].

Fuzzy strategy:

$$\lfloor e_{\mathrm{u}^* j} \rfloor = \sup_{u \in \Omega_{k(j)}} \{ e_{\mathrm{u}_j} \}, \tag{26}$$

where $i = \mathbf{u}^*$.

Fuzzy probabale strategy:

$$X_{ij}^{k} = \frac{e_{ij}^{k}}{\sum_{u \in \omega_k}(j)\, e_{\mathrm{u}_j}^{k}}. \tag{27}$$

We have used GWO Algorithm for exploitation process, on the solutions generated by the ACO for exploration process. The procedure for updating the positions of the wolf is same as given in [34]. GWO Algorithm is used for the exploration process while ACO is used for exploitation process of the solutions. The processor and memory instances are created using algorithm from [14] (Table 2).

Table 2. Algorithm to create the processor and memory instances in random

for $i = 0$ to $(n - 1)$ do
$\boldsymbol{D}_{\mathbf{proc}(i)} = \mathbf{rand}\ (2\overline{\boldsymbol{D}_{\mathbf{proc}}})$
$\boldsymbol{D}_{\mathbf{memory}(i)} = \mathbf{rand}\ (2\overline{\boldsymbol{D}_{\mathbf{memory}}})$
S= Numbers are generated in random using function rand(1.0)
If $(S < P \wedge \boldsymbol{D}_{\mathbf{proc}(i)} \ge \boldsymbol{D}_{\mathbf{proc}}) \vee (S \ge P \wedge \boldsymbol{D}_{\mathbf{proc}(i)} < \boldsymbol{D}_{\mathbf{proc}})$ then
$\boldsymbol{D}_{\mathbf{memory}(i)} = \boldsymbol{D}_{\mathbf{memory}(i)} + \overline{\boldsymbol{D}_{\mathbf{memory}}}$
Endif
Endfor

6.2. Fuzzy HAGA Algorithm

Table 3. Fuzzy HAGA Algorithm

1. Initialize the required quantity of Physical Machines (PMs), and the requested quantity of Virtual Machines (VMs). 2. The capacity constraint is set for the Physical Machines. 3. The requirement demands of VMs are initialized. 4. The maximum number for the iteration is fixed. 5. Initialize the Pheromone matrix $\boldsymbol{\tau_j}$ and the total number of ants 6. Use procedure given in GWO algorithm to generate server consolidation problem instances as in Table. 1 7. **Loop1:** **Repeat for k=0** **Loop2:** Take a server which is not used so far from a set of Physical servers. **Loop3:** For Number of VMs =1 to n Determine the desirable heuristic data from (7) Determine the probabilistic movement from (10) End for Pick a VM from the list of VMs for replacing using Fuzzy state transition rule applying (23) and (24). If there are any remaining VM that could be fit in the server, Go to **Loop3** Go to **Loop2** Upgrade the pheromones to the best solution using the Upgrade rule for local solution given in (12). **Do until total number of ants (TNA) – 1** 8. The Objective function value is set using the (2). 9. Apply HAGA Algorithm to obtain new optimal solutions. 10. Modify the values of Pheromones applying updating rule globally as stated in (14). 11. Goto **Loop1.** 12. Display the global best solution along with the fitness value

6.3. Results and discussion

Using (14), the VM demand instances are generated in Java platform using ACO Algorithm [14]. The mapping is done in such a way that one VM is assigned to a single PM as the worst case complexity. 200 VMs are taken for performing our experiment. The no. of VMs are scalable as it is given in [14]. The initialization of the variables are done as follows:

$\boldsymbol{q_0}$ = 0.8. NA=10, M=100, $\boldsymbol{\alpha = 0.45}$, $\boldsymbol{\rho_l}$=$\boldsymbol{\rho_g}$=0.35, $\boldsymbol{\tau_{pg}}$=$\boldsymbol{\tau_{mj}}$=90%, η =0.0001.

20 runs per instance has been performed and the results are taken for an average of 20 runs. LB is the Lower Bound and defined as:

$$\left\{\left\lceil\left(\textstyle\sum_{i=1}^{n} D_{\text{proc}}^{i}\right)/T_{\text{proc}}^{j}\right\rceil, \left\lceil\left(\textstyle\sum_{i=1}^{n} D_{\text{memory}}^{i}\right)/T_{\text{memory}}^{j}\right\rceil\right\}. \quad (28)$$

When the number of non-dominated solutions is more than one, a non-dominated solution is chosen in random. The demand set of CPU and the memory utilization were the problem instances generated. The reference values of CPU and memory were set to 25% and 45% for experiment. The threshold of utilization of both CPU and memory is set to 90%. The demand for CPU and memory are taken as references and it is noticed that the lower bound of servers taken for replacement has come down in our experiment.

Table 4. VM requirements of Server consolidation (reference values = 25% and 45%)

Algorithm	$\overline{D_{\text{proc}}}$=$\overline{D_{\text{memory}}}$=25%			$\overline{D_{\text{proc}}}$=$\overline{D_{\text{memory}}}$=45%		
	Probability Value = –0.754			Probability Value = –0.755		
	Count of PMs, m	m/LB	Time, s	Count of PMs, m	m/LB	Time, s
Fuzzy HAGA	94	1.04	7.13	191	1.19	8.53
Fuzzy ACS	95	1.05	7.16	191	1.20	8.56
FFD	125	1.38	8.34	218	1.36	24.61
ACS	97	1.07	5.31	194	1.21	6.47
MMAS	101	1.12	5.26	195	1.21	6.53

The average power consumption and resource wastage of virtual machine requirements having reference value as 25% and probability value as –0.754 and –0.755. It is observed that the count of Physical machines is decreased in our algorithm by one when compared to that of Fuzzy ACS and also the lower bound of the count of the servers has also come down in our proposed algorithm.

Table. 5. VM requirements of Server consolidation (reference values = 25% and 45%)

Algorithm	$\overline{D_{\text{proc}}}$=$\overline{D_{\text{memory}}}$=25%			$\overline{D_{\text{proc}}}$=$\overline{D_{\text{memory}}}$=45%		
	Probability Value = –0.348			Probability Value = –0.374		
	Count of PMs, m	m/LB	Time, s	Count of PMs, m	m/LB	Time, s
Fuzzy HAGA	93	1.03	7.14	186	1.11	8.51
Fuzzy ACS	94	1.04	7.16	189	1.12	8.53
FFD	121	1.34	8.33	207	1.21	24.69
ACS	96	1.05	5.32	191	1.15	6.24
MMAS	98	1.06	5.21	192	1.15	6.47

The average power consumption and resource wastage of virtual machine requirements having reference value as 25% and probability value as –0.754 and –0.755. It is observed that the count of Physical machines is decreased in our algorithm by one when compared to that of Fuzzy ACS and also the lower bound of the count of the servers has also come down in our proposed algorithm.

Table 6. VM requirements of Server consolidation (reference values = 25% and 45%)

Algorithm	$\overline{D_{\text{proc}}}$=$\overline{D_{\text{memory}}}$=25%			$\overline{D_{\text{proc}}}$=$\overline{D_{\text{memory}}}$=45%		
	Probability Value = –0.072			Probability Value = –0.052		
	Count of PMs, m	m/LB	Time, s	Count of PMs, m	m/LB	Time, s
Fuzzy HAGA	92	1.02	7.08	178	1.12	8.44
Fuzzy ACS	93	1.03	7.12	180	1.14	8.48
FFD	112	1.24	8.21	195	1.21	24.61
ACS	94	1.04	5.29	184	1.15	6.27
MMAS	96	1.06	5.21	185	1.15	6.49

The average power consumption and resource wastage of VM requirements having reference value as 25% and probability value as –0.072 and –0.052. It is observed that the count of PMs is decreased in our algorithm by one when compared to that of Fuzzy ACS and also the lower bound of the count of the servers has also come down in our proposed algorithm.

Table. 7. VM requirements of Server consolidation (reference values = 25% and 45%)

Algorithm	$\overline{D}_{proc}=\overline{D}_{memory}$=25%			$\overline{D}_{proc}=\overline{D}_{memory}$=45%		
	Probability Value = 0.371			Probability Value = 0.398		
	Count of PMs, m	m/LB	Time, s	Count of PMs, m	m/LB	Time, s
Fuzzy AG	92	1.02	7.08	178	1.10	8.40
Fuzzy ACS	93	1.03	7.12	180	1.12	8.43
FFD	112	1.24	8.21	195	1.21	24.54
ACS	94	1.04	5.29	184	1.15	6.24
MMAS	96	1.06	5.21	185	1.15	6.47

The average power consumption and resource wastage of virtual machine requirements having reference value as 25% and probability value as0.371and 0.398. It is observed that the count of Physical machines is decreased in our algorithm by one when compared to that of Fuzzy ACS and also the lower bound of the count of the servers which are used for replacement has also come down in our proposed algorithm.

Table. 8. VM requirements of Server Consolidation (reference values = 25% and 45%)

Algorithm	$\overline{D}_{proc}=\overline{D}_{memory}$25%			$\overline{D}_{proc}=\overline{D}_{memory}$45%		
	Probability Value = 0.755			Probability Value = 0.751		
	Count of PMs, m	m/LB	Time, s	Count of PMs, m	m/LB	Time, s
Fuzzy HAGA	90	1.00	7.03	171	1.06	8.36
Fuzzy ACS	91	1.01	7.09	172	1.08	8.41
FFD	105	1.16	8.19	190	1.18	24.23
ACS	93	1.03	5.28	176	1.10	6.23
MMAS	95	1.05	5.18	181	1.13	6.42

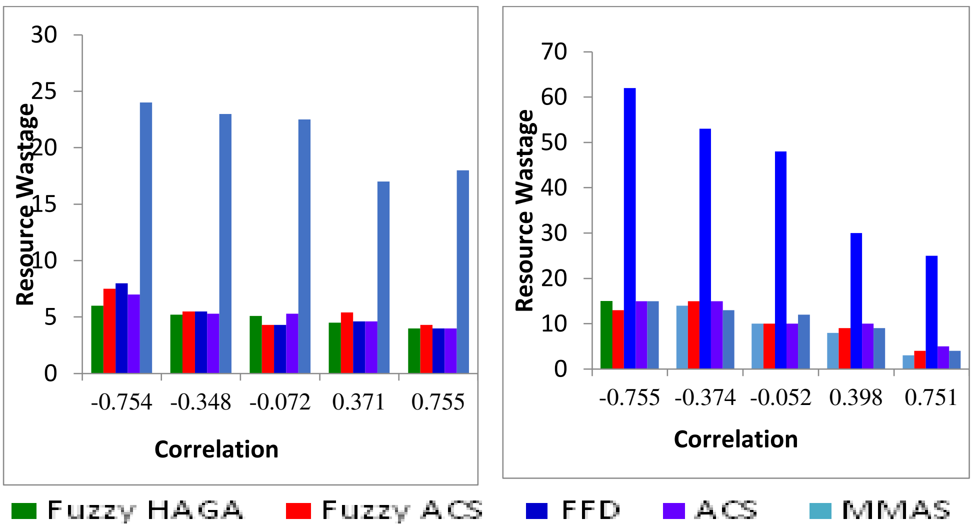

Fig. 4. Number of servers initialized when $D_{proc} = D_{memory}$ = 25% and 45%

The average power consumption and resource wastage of virtual machine requirements having reference value as 25% and probability value as = 0.775 and 0.751. It is observed that the count of PMs is decreased in our algorithm by one when compared to that of Fuzzy ACS and also the lower bound of the count of the servers has also come down in our proposed algorithm.

From the above results, we can see in the graph that our Fuzzy HAGA algorithm performs better server consolidation when compared with other approaches in terms of the number of PMs. The number of PMs were reduced by implementing our proposed algorithm when compared to other algorithms like Fuzzy ACS and ACS. The lower bound on it also has been decreased when compared to other given algorithms. It is also noted that the time taken for consolidating the physical machines are also decreased.

7. Conclusion

With the growing advancement in cloud computing environment, a research problem that needs an attention is the Virtual Machine Placement Problem (VMPP), which needs server consolidation as its strategy. It targets the optimal placement of VM to a suitable PM to obtain an optimal solution that minimizes the total power consumption and resource wastage. To achieve this, server consolidation helps to pack maximum number of VMs that could be packed in one server so as to reduce the power consumption and resource wastage. The proposed Fuzzy HAGA Algorithm has been tested against some of the algorithms given above. In our experiment, GWO Algorithm has been used to optimize the solutions got through using ACO Algorithm. The results prove that our Fuzzy HAGA Algorithm performs better server consolidation than the Fuzzy ACS, FFD, ACS and MMAS. Server consolidation problem is a problem with high complexity and Fuzzy HAGA Algorithm has been designed efficiently to handle this problem. In future, a better optimal solution will be explored using some other hybrid optimization techniques to reduce the number of PMs and decrease the lower bound value of the physical machines.

References

1. Chamas, N., F. Lopez-Pires, B. Baran. Two-Phase Virtual Machine Placement Algorithms for Cloud Computing: An Experimental Evaluation under Uncertainty. – IEEE Conference, 2017.
2. Uddin, M., A. A. Rehman. Server Consolidation: An Approach to Make Data Centers Energy Efficient and Green. – International Journal of Scientific and Engineering Research, Vol. **1**, 2010, Issue 1.
3. Ali, H. M., D. C. Lee. A Biogeography-Based Optimization Algorithm for Energy Efficient Virtual Machine Placement. – IEEE Symposium on Swarm Intelligence, 2014.
4. Panigrahy, R., K. Talwar, L. Uyeda, U. Wieder. Heuristics for Vector Bin Packing, 2011.
   Research. microsoft.com

5. A g r a w a l, S., S. B o s e, S. S u n d a r r a j a n. Grouping Genetic Algorithm for Solving the Server Consolidation Problem with Conflicts. Genetics and Evolutionary Computation. – In: Proc. of 1st ACM/SIGEVO Summit, ACM, 2009.
6. A l i, R., Y. S h e n, X. H u a n g, J. Z h a n g, A. A l i. VMR: Virtual Machine Replacement Algorithm for QoS and Energy-Awareness in Cloud Data Centers. – In: IEEE International Conference on Computational Science and Engineering, 2017.
7. L i u, X., Z. Z h a n, J. D. D e n g, Y. L i, T. G u, J. Z h a n g. An Energy Efficient Ant Colony System for Virtual Machine Placement in Cloud Computing. – IEEE Transactions on Evolutionary Algortihm, Vol. **22**, 2018, Issue 1.
8. D o r t e r l e r, S., M. D o r t e r l e r, S. O z d e m i r. Multi-Objective Virtual Machine Placement Optimization for Cloud Computing. – In: IEEE International Symposium on Networks, Computers and Communication, 2017.
9. S o t o m a y o r, B. Provisioning Computational Resources Using Virtual Machines and Leases. PhD Thesis Submitted to the University of Chicago, USA, 2010.
10. K a n s a l, N. J., I. C h a n a. Energy-Aware Virtual Machine Migration for Cloud Computing – A Firefly Optimization Approach. – Journal of Grid Computing, SpringerLink, Vol. **14**, 2016, Issue 2, pp. 327-345.
11. Z h o u, A., S. W a n g, B. C h e n g, Z. Z h e n g, F. Y a n g, R. N. C h a n g, M. R. L y u, R a j k u m a r B u y y a. Cloud Service Reliability Enhancement via Virtual Machine Placement Optimization. – IEEE Transactions on Services Computing, Vol. **10**, 2017, No 6.
12. F u, X., C. Z h o u. Predicted Affinity Based Virtual Machine Placement in Cloud Computing Environments. – IEEE Transaction on Cloud Computing, Vol. **13**, 2014, No 9.
13. K h o s r a v i, A., L. L. H. A n d r e w, R a j k u m a r B u y y a. Dynamic VM Placement Method for Minimizing Energy and Carbon Cost in Geographically Distributed Cloud Data Centers. – IEEE Transactions on Sustainable Computing, Vol. **2**, 2017, No 2.
14. G a o, Y., H. G u a n, Z. Q i, Y. H o u, L. L i u. A Multi-Objective Ant Colony System Algorithm for Virtual Machine Placement in Cloud Computing. – Journal of Computer and System Sciences, Vol. **79**, 2013, No 8, pp. 1230-1242.
15. K u m a r, D., D r. T a r n i M a n d a l. Bi-Objective Virtual Machine Placement Using Hybrid of Genetic Algorithm and Particle Swarm Optimization in Cloud Data Center. – International Journal of Applied Engineering Research, Vol. **12**, 2017, No 22, pp. 12044-12051. ISSN 0973-4562.
16. Z h a n g, J., X. W a n g, H. H u a n g, S. C h e n. Clustering Based Virtual Machines Placement in Distributed Cloud Computing. – Future Generation Computer Systems, Elsevier, Vol. **66**, 2017, pp.1-10.
17. B a r l a s k a r, E., N. A j i t h S i n g h, Y u m n u m, Y. J. S i n g h. Energy Optimization Methods for Virtual Machine Placement in Cloud Data Center, ADBU. – Journal of Engineering and Technology, 2014.
18. T a w f e e k, M. A., A. B. E l-S i s i, A. E. K e s h k, F. A. T o r k e y. Virtual Machine Placement Based on Ant Colony Optimization for Minimizing Resource Wastage. – In: International Conference on Advanced Machine Learning Technologies and Applications (AMLTA'14), 2014, p.153.
19. A l b o a n e e n, D. A., H. T i a n f i e l d, Y. Z. G l a s g o w. Glowworm Swarm Optimization Algorithm for Virtual Machine Placement in Cloud Computing. – In: Proc. of IEEE International Conference on Cloud and Big Data Computing, 2016, pp. 808-814.
20. S p e i t k a m p, B., M. B i c h l e r. A Mathematical Programming Approach for Server Consolidation Problems in Virtualized Data Centers. – IEEE Transactions on Services Computing, 2010, pp. 266-278.
21. S a i t, S. M., A. B a l a, A. H. E l-M a l e h. Cuckoo Search Based Resource Optimization of Datacenters. – Applied Intelligence, 2015, pp. 1-18.
22. R i q u e l m e, N., C. V. L u c k e n, B. B a r a n. Performance Metrics in Multi-Objective Optimization. – In: IEEE Latin American Computing Conference, 2015.
23. W u, G., M. T a n g, Y. C. T i a n, W. L i. Energy Efficient Virtual Machine Placement in Data Centers by Genetic Algorithm. – In: International Conference on Neural Information Processing (ICONIP), Neural Information Processing, 2012, pp. 315-323.

24. Z h a o, L., L. L u, Z. J i n, C. Y u. Online Virtual Machine Placement for Increasing Cloud Provider's Revenue. – IEEE Transactions on Services Computing, Vol. **10**, 2017, No 2.
25. S h a b e e r a, T. P., S. D. M a d h u k u m a r, S. M. S a l a m, K. M u r a l i k r i s h n a n. Optimizing VM Allocation and Data Placement for Data-Intensive Applications in Cloud Using ACO Metaheuristic Algorithm. – International Journal of Engineering Science and Technology, Vol. **20**, 2017, Issue 2, pp. 616-628.
26. G a g w e r o, M. G., L. C a v i g l i o n e. Model Predictive Control for Energy Efficient, Quality-Aware Virtual Machine Placement. – IEEE Transactions on Automation Science and Engineering, 2018, Issue 1, pp. 1-13.
27. J i n g, H. Z., W. F. L i u, Q. W a n g, W. Z h a n g, Q. Z h e n g. Power-Aware and Performance – Guaranteed Virtual Machine Placement in the Cloud. – IEEE Transactions on Parallel and Distributed Systems, Vol. **29**, 2018, No 6.
28. H u n g, N. Q., P. D. N i e n, N. H. N a m, N. H. T u o n g, N. T h o a i. A Genetic Algorithm for Power-Aware VirtuaMachine Allocation in Private Cloud. – Lecture Notes in Computer Science, Vol. **7804**, 2013.
29. S a r v e s h, K u m a r. Discrete Gravitational Search Algorithm for Virtual Machine Placement in Cloud Computing. – International Journal of Pure and Applied Mathematics, Vol. **117**, 2017, No 19, pp. 337-342.
30. B o o m i n a t h a n, P., M. A r a m u d a n, Ra. K. S a r a v a n a g u r u. Fuzzy Bio-Inspired Hybrid Techniques for Server Consolidation and Virtual Machine Placement in Cloud Environment. – Cybernetics and Information Technologies, Vol. **17**, 2017, No 4, pp. 52-68.
31. M i r j a l i l i, S., S. M. M i r j a l i l i, A. L e w i s. Grey Wolf Optimizer. – Journal in Advances in Engineering Software, Vol. **69**, 2014, pp. 46-61, ScienceDirect.
32. J o s h i, S., J. C. B a n s a l. Grey Wolf Gravitational Search Algorithm. – International Workshop on Computational Intelligence (IWCI), IEEE, 2016.
33. M a r y u m a, K., S. K. C h a n g, D. T. T a n g. A General Packing Algorithm for Mutidimensional Resource Requirements. – International Journal of Computer and Information Sciences, Vol. **6**, 1977, Issue 2, pp. 131-149.
34. R o d r i g u e z, L., O. C a s t i l l o, J. S o r i a. Grey Wolf Optimizer with Dynamic Adaptation Parameters Using Fuzzy Logic. – In: IEEE Congress on Evolutionary Computation(CEC), 2016.
35. M i r j a l i l i, S., S. S a r e m i, S. M. M i r j a l i l i, L. S. C o e l h o. Multiobjective Grey Wolf Optimizer: A Novel Algorithm for Multi-Criterion Optimization. – Elsevier Journal of Expert Systems with Applications, 2015.

Received: 31.10.2018; Second Version: 15.12.2019; Accepted: 27.12.2019