JACET

J. ADV COMP ENG TECHNOL, 7(1) Winter 2021

Energy-aware and Reliable Service Placement of IoT applications on Fog Computing Platforms by Utilizing Whale Optimization Algorithm

Yaser Ramzanpoor[1], Mirsaeid Hosseini Shirvani[2], Mehdi Golsorkhtabaramiri[3]
1- Computer Engineering Department, Babol Branch, Islamic Azad University, Babol, IRAN.
2- Computer Engineering Department, Sari Branch, Islamic Azad University, Sari, IRAN. (mirsaeid_hosseini@iausari.ac.ir)
3- Computer Engineering Department, Babol Branch, Islamic Azad University, Babol, IRAN.

Received (2021-03-12) Accepted (2021-07-27)

Abstract: *Fog computing is known as a new computing technology where it covers cloud computing shortcomings in term of delay. This is a potential for running IoT applications containing multiple services taking benefit of closeness to fog nodes near to devices where the data are sensed. This article formulates service placement issue into an optimization problem with total power consumption minimization inclination. It considers resource utilization and traffic transmission between different services as two prominent factors of power consumption, once they are placed on different fog nodes. On the other hand, placing all of the services on the single fog node owing to power reduction reduces system reliability because of one point of failure phenomenon. In the proposed optimization model, reliability limitations are considered as constraints of stated problem. To solve this combinatorial problem, an energy-aware reliable service placement algorithm based on whale optimization algorithm (ER-SPA-WOA) is proposed. The suggested algorithm was validated in different circumstances. The results reported from simulations prove the dominance of proposed algorithm in comparison with counterpart state-of-the-arts.*

Keywords: *Fog computing; Service Placement Problem (SPP); Whale Optimization Algorithm (WOA); Internet of Things (IoT).*

How to cite this article:
Yaser Ramzanpoor, Mirsaeid Hosseini Shirvani, Mehdi Golsorkhtabaramiri. Energy-aware and Reliable Service Placement of IoT applications on Fog Computing Platforms by Utilizing Whale Optimization Algorithm. J. ADV COMP ENG TECHNOL, 7(1) Winter 2021 : 67-80

1. INTRODUCTION

FOG computing paradigm was recently emerged to process new IoT applications at the edge networks near to IoT devices where the data is being sensed [1]. In this way, it covers cloud datacenter's intrinsic delay. Service oriented architecture (SOA) leads reduction of software and application development costs owing to reusability attribute [2-4]. Each IoT application may contain couple of components or services. Service placement of IoT applications at the edge network also yields reduction of network traffic load and guarantees to deliver on-time services to its users. For the sake of fog computing heterogeneity in terms of the number of processing nodes; memory, processing, and storage capacities; and also dynamic nature of workload, the service placement of IoT application in aforesaid environment is a great challenge. Each IoT application has its own requirements such as computing resources, degree of delay sensitivity, and privacy concerns. In this regards, each proposal of service placement scheme must be aware of both application requirements and fog infrastructure capacities in terms of software facilities, hardware, and delay between each pair of fog nodes [5-6]. To have efficient utilization of fog resources commensurate

with requested quality of service (QoS) associated to IoT applications, designing a smart scheduler is necessary to be aware of requirement both sides, namely users and providers [7]. In provider's perspective the power management is the most important business objective whereas the desired QoS is the most important objective to be met. In addition, the users also need the degree of reliability in this vulnerable environment. Service placement relevant to requested application on the minimum number of fog nodes brings different plus points such as increase the resource utilization, lower power consumption, and shorten the amount of total delay between communicative services. Placement of services associated to customer application only on one node of fog infrastructure leads to optimal use of physical and network resources. Also, it minimizes energy consumption, but it suffers from reliability problem. As shown in Fig. 1 (a), when a fog node containing all services of an application crashes, the application cannot work correctly in which it affects the quality of its service [8-9]. For this reason, a solution must be adopted for the reliable placement of the services of an application. For example, as shown in Fig. 1 (b), a fault tolerance threshold is set for the application. Depending on the amount of fault tolerance, the minimum number of fog nodes must be determined to ensure the service quality while reducing energy consumption to accommodate the services of this application and also the effect of fog node failure on the application must be acceptable.

Therefore, presenting an efficient and at the same time reliable service placement in the fog environment is necessary. This problem is computationally NP-Hard especially in larger search space it needs huge amount of processing time to take a trade-off between conflicting objective functions. Several works were proposed in literature to solve service placement problem with regards to different objectives such as service delay, resource utilization, and network latency [10-14], but less paid on service reliability at the same time. For instance, Yousefpour et al. proposed an IoT-fog-cloud framework for service placement problem to minimize delay as an objective function [15]. In this regards, Mahmud et al. presented a latency-aware application module management for fog computing environment [13]. Literature review points out that there is a clear lack in presenting energy aware reliable service placement scheme for IoT application in fog platforms. To address this issue, the current paper presents an energy-aware reliable service deployment scheme on fog platforms.

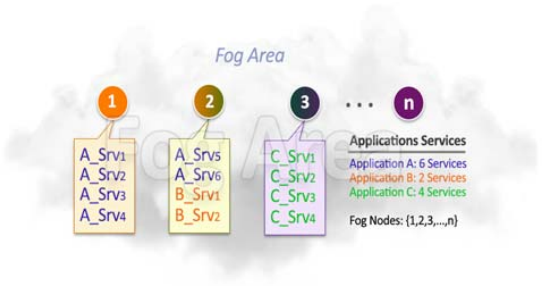


Fig. 1 (a). Application Services Placed on a Single Fog Node

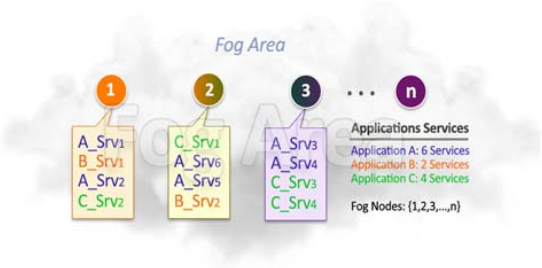


Fig. 1 (b). Application Services Placed on a Distributed Fog Nodes

Fig. 1. Placement of IoT Application Services

The main contribution of the current paper are as follow:

1- It presents a reliability model in fog platform. For this purpose, a threshold for the minimum number of fog nodes is determined for the placement of application services, to compromise the reliability of application and energy consumption. Services are then placed on the distributed nodes of the fog to deal with a single point of failure.

2- It formulates the service placement problem as an optimization problem

3- It presents a customized whale optimization algorithm to solve the problem with the aim of reducing energy consumption and enhancing the reliability of applications.

The paper reminder is structured as follows: section 2 discuses related works. System framework is presented in section 3. Section 4 is dedicated to problem formulation. The proposed methodology is placed in section 5. In section 6, simulation and evaluation are conducted. Section 7 sums up the paper.

2. RELATED WORKS

Fog computing is the newest computing era which has its own challenges similar to each new technology. To be familiar with fog computing challenges, this section pays on literature review in regards to the main scope of current paper.

A network-aware service placement algorithm was presented to improve resource utilization in fog environment [10]. It sorts fog nodes based on their capacities and application components (services) based on their requirements. Then, the service distribution is done provided the requirement is met [10]. In addition, a general and extensible reference model was proposed which presents a description of QoS-aware deployment for IoT devices in fog environment [11].

A typical IoT application consists of various services running together with active interdependencies; traditionally running on the cloud hosted in global data centers. Authors in [11], have presented a module (service) mapping algorithm for efficient utilization of resources in the network infrastructure by efficiently deploying Application services in Fog-Cloud Infrastructure for IoT based applications. This algorithm detects fog nodes based on their capacity and application services requirement. If requirement is met, the mapping of services over fog nodes is done. This policy leads a way to maximizing resource utilization for distributed IoT applications. Also, this algorithm proves that interaction between cloud and fog yields better performance in term of end-to-end delay in comparison with only cloud-based approaches.

Authors in [12] have proposed to employ topology and orchestration specification for cloud applications as a new standard for cloud service management to systematically specify the services and configurations of IoT applications. By using this standard, application models can be reused, and deployment processes can be automated in heterogeneous IoT system environments. In this model, placement of application services is automatically done by applying conceptual description of components topology and related application placement.

To fully leverage the capabilities of the fog nodes, large-scale applications that are decomposed into interdependent application modules (services) can be deployed in an orderly way over the nodes based on their latency sensitivity. A latency-aware application service placement policy for the fog environment that meets the diverse latency in service delivery and the data volume to be performed for different applications [13]. This policy guarantees the application QoS in favorable deadline. In addition to, this procedure optimize underlying fog resource utilization. To do so, the algorithm prioritize the sensitivity of services against elapsed time; then, based on the degree of delay sensitivity it dispatches the service to be placed in the nearest fog nodes. For the sake of optimization in utilizing the number of active fog nodes, the forward and re-allocation application modules strategies are used. In this managerial algorithm, delay to service access, service delivery time, and internal communication delay have been considered.

A framework has been presented for the dynamic generation of optimized placement topologies for IoT cloud applications that are tailored to the currently available physical infrastructure [14]. Based on a declarative, constraint-based model of the desired application placement, this approach enables flexible provisioning of application services on edge devices deployed in the field [29]. In production process of deployment topology, some parameters such as time needed for deployment, time and bandwidth request for application running, and exploitation of edge devices are evaluated.

A lightweight QoS-aware dynamic fog service provisioning framework proposed in [15]. In this proposal, dynamic service placement in fog environment is done. In this regards, the previously deployed modules are set free to meet favorable QoS and low latency along with minimizing overall cost.

Similar to each computing system, the cost and efficiency are two prominent metrics for system evaluation. For this, several works have

been published for service placement problem in Fog-IoT environment with regards to cost and efficiency parameters [16-18]. A geographically dispersed sensor application has been run in fog environment which proposed by Canali and Lancellotti [16].

The QoS-aware approach has been proposed to sole service placement problem in fog-cloud computing systems [17]. This algorithm considers the deadline requirement of each IoT application so that the most sensitive applications on delay are deployed on the devices as proximity as possible to serve the users. In addition, to decline the network bandwidth wastage and the cost of execution in the cloud, the number of assigned services relevant to applications to the fog environment is maximized.

Three different meta-heuristic approaches were compared in service placement of fog environment [18]. In this comparison the fog infrastructure and fog applications have been under study by considering some objective functions such as to optimize the network latency, the service distribution rate and the resources utilization.

Table 1 summarizes comparison of related works associated to IoT application service placement on fog and a cloud infrastructure.

Table 1.Summary of the Literature Study

work	Service Placement Policy				
	Fault Tolerant	Latency Aware	Traffic Aware	Resource Aware	Energy Efficient
[10]	×	✓	×	✓	×
[11]	×	✓	×	✓	✓
[12]	×	×	×	×	×
[13]	×	✓	×	✓	✓
[14]	×	✓	×	✓	✓
[15]	×	✓	×	✓	✓
[16]	×	✓	×	✓	×
[17]	×	✓	×	✓	✓
[18]	×	✓	×	✓	✓
Proposed Paper	✓	✓	✓	✓	✓

Review of related works show that there is a gap for presenting reliable service placement for IoT application in fog environment which is energy-aware at the same time.

3. SYSTEM FRAMEWORK AND MODELS

This section presents proposal system framework and its components specifications. In addition, some models are presented to ease problem formulation. Fig. 2 demonstrates a suggested system frame work which has differ components. As the Fig. 2 shows, at the topmost level there is an organizer component in which its duty is to extract all sub full mesh so called-clique. Then, an appropriate clique, sub full mesh, which can fulfill services requirement in terms of processing, memory, bandwidth, and delay between sub services is selected. To have efficient deployment, designing of service deployment planner (SDP) which should be aware aware of both application requirement and underlying fog platform is necessary.

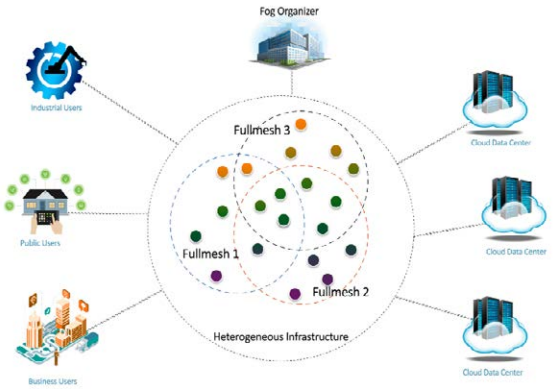


Fig. 2. Proposed System Framework

The components associated to SDP module are enlisted as below:

Application Service Manager (ASM): This is the main component for making decision how to deploy services on fog nodes. In an application with several dependent services, the deployment scheme must be aware of application requirement and underlying fog platforms capacities.

Service Requirement Information (SRI): This component registers service requirement in terms of processing, memory, delay, and bandwidth requirements delivered from IoT applications.

Services Communication Information (SCI): This component extracts services communication pattern from user's IoT application requests.

Then, it delivers it to ASM component.

Resource Detector of Full Submesh (RDFS): This component extracts sub full mesh from whole network of fog nodes which makes a connection graph. Then, it detects all resources and their capabilities from detected full sub mesh and shares this information to ASM component.

Sub Mesh Network Manager (SMNM): This component keeps underlying network topology periodically since the environment is changeable and dynamic.

Fig.3 demonstrates elements of SDP component.

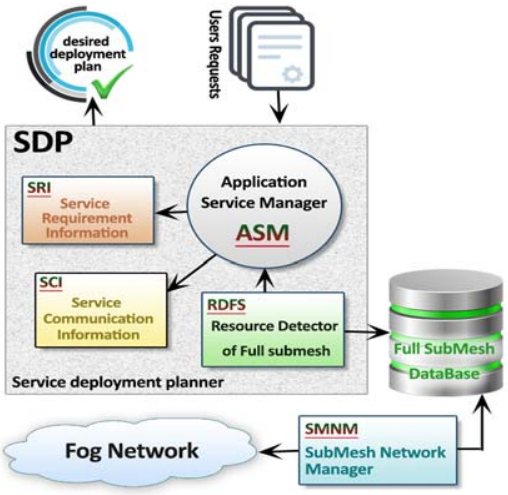


Fig. 3. Proposed SDP model

3-1 Fog System Model

The fog system (F) under study comprises n heterogeneous fog nodes which are heterogeneous in terms of processing strength and power consumption. Each node which belongs to sub full mesh can host services associated to IoT applications. The fog nodes can communicate via wireless protocols. In this line each fog node $fn \in F$ is specified by a vector {*id,mid,H,S,sensorlist*} where *id* is fog node identifier, *mid* is mesh network *id*; and *H*, *S*, and *sensor lists* are hardware, software, and available sensor list relevant to that fog node respectively. As the services associated to an application are distribute on fog nodes, the communication between fog nodes is specified by {*L,B*} where *L* and *B* indicate to delay and bandwidth respectively. The Fig. 4 elaborates an example full mesh network.

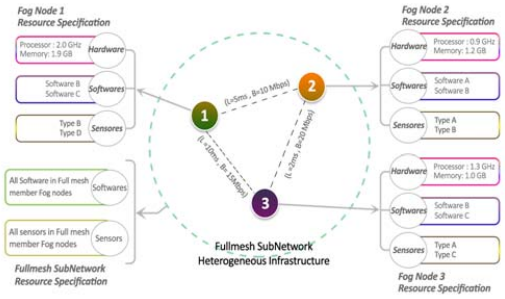


Fig. 4. Proposed Full Mesh Network.

3-2 Communication Network Model

The communication between fog nodes is modeled via graph data structure. This graph mode is $G=(F,D)$, where $F=\{fn_1,fn_2,\ldots,fn_n\}$, is the set of fog nodes and D is distance matrix. In two dimensional matrix D, each element d_{ij} is 1 if fog node fn_i can directly access fog node fn_j otherwise this element is set to zero.

3-3 Application Model

Similar to other internet-based applications, IoT application design adopts SOA architecture to be flexible enough commensurate with changeable users requests [19]. To this end, the proposed application model has multi-service structure [20]. The services have mutually dependency to collaboratively fulfill users' requests. For instance, consider a healthcare system which is designed for aging people. It has three services *SR1, SR2,* and *SR3*. Service one (*SR1*) is the status manager that monitors aging person's status and urgent information to the nearest hospital or medical center. Service two (*SR2*) is controller center which is used for interpreting the gathered data and manually the system control. Finally, the service three (*SR3*) is machine learning service that is used for registering data history of each person, the estimation of near future wellbeing of each aging person that is not delay-sensitive the reason why it can be deployment on fog nodes. Fig. 5 illustrates each software and hardware requirement for each service.

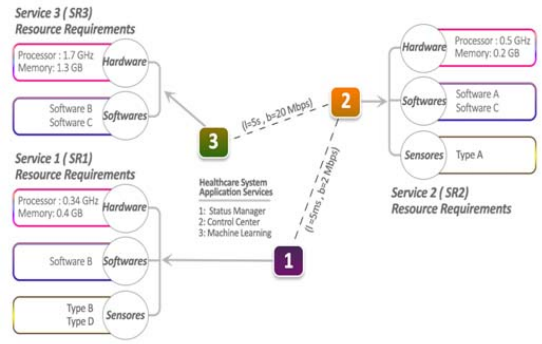


Fig.5. Specification of Services Associated to an Application.

In addition to, the communication link and tolerable delay were drawn in Fig. 5. A request application (*App*) is modeled via a graph $G=(SR,T)$ where $SR=\{SR_1, SR_2, \ldots, SR_m\}$ associated to requested application and T is traffic matrix pattern between each pair of services so that t_{ij} indicates the amount of traffic transferring from service SR_i to service SR_j.

3-4 Reliability Model

Placement of an application's services on the minimum number of nodes in a fog leads to efficient utilization of computing resources and maximum achievement of objectives such as reducing energy consumption. But one of the challenges is the occurrence of a single failure point phenomenon for applications. Therefore, in order to simultaneously meet the objectives of fog stakeholders in achieving optimal goals and also to reduce the degree of vulnerability of applications in the centralized placement of services in the fog infrastructure, a threshold for the number of fog nodes for the placement of application services is considered. For this purpose, the minimum number of host fog nodes to accommodate services is considered equal to the member nodes of the selected full mesh subnetwork, where fog nodes reach to each other with only one hop connection.

3-5 Power Model

To present power model, the effective elements in regards to power consumption must be determined. In the proposed system framework two important factors are fog node resource utilization and traffic pattern between each node which have direct impact on total power consumption. The average normalized resource utilization of each fog node fn_i is formulated via Eq. (1) where W_1 and W_2 are respectively the importance coefficient of processor and memory resource in the under study system and $0 \le W_1 \le 1$, $0 \le W_2 \le 1$, $W_1+W_2=1$. The terms $r_{SR_j}^{CPU}$ and $r_{SR_j}^{RAM}$ are respectively the requested amount of *CPU* and *RAM* for service *SRj*. Moreover, the terms $R_{fn_i}^{CPU}$ and $R_{fn_i}^{RAM}$ are the maximum *CPU* and *RAM* capacity of fog node fn_i respectively.

$$U_{fn_i}^{Res} = \frac{w_1 . \sum_j^{fn_i} \frac{r_{SR_j}^{CPU}}{R_{fn_i}^{CPU}} + w_2 . \sum_j^{fn_i} \frac{r_{SR_j}^{RAM}}{R_{fn_i}^{RAM}}}{2} \quad (1)$$

So, the power consumption (PW) of each fog node fn_i is calculated via Eq. (2).

$$\mathrm{PW}(U_{fn_i}^{Res}) = y_{fn} \times (P_{max} - P_{min}) \times U_{fn_i}^{Res} + P_{min} \quad (2)$$

In Eq. (2), variables P_{max} and P_{min} are respectively the fog node power consumption in the full load and idle modes. In this line, decesion binary variable y_{fn} indicates whether the fog node is active or not. Power consumption owing to data transmission between each pair of fog nodes are measured by Eq. (3) where P_{tr} indicates the amount of power consumed for each unit of data transmission. Note that if two services are placed in the same fog node, the transmission power consumption is ignored.

$$\mathrm{PW(Tr)} = \sum_{fn_i \neq fn_j} t_{SR_i,SR_j} \times P_{tr} \quad (3)$$

The power consumption of each fog node is measured by summation of Eqs. (2-3) in Eq. (4).

$$\mathrm{PW}(fn_i) = \mathrm{PW}(U_{fn_i}^{Res}) + \mathrm{PW}(\mathrm{Tr}) \quad (4)$$

4. PROBLEM FORMULATION

This section formulates service placement problem into an optimization problem with minimization of total power consumption (TPC) in the fog system which Eq. (5) indicates. Since this optimization problem must be reliable, some constraints are imposed to proposed optimization problem. The constraints are formulated in Eqs. (6-12).

$$\boldsymbol{min\ TPC} = Min \sum_{i=\{1,\dots n\}} \mathrm{PW}(fn_i).\, y_{fn,i} \quad (5)$$

Subject to:

$$\sum_{SR_i \in fn_m} \sum_{SR_j \in fn_n} b_{ij} \times l_{ij} < B_{mn} \times L_{mn} \quad (6)$$

$$\sum_{\mathrm{SR} \in App} x_{SR,fn}.\, hw_{SR} \le HW_{fn}, \forall fn \in F \quad (7)$$

$$\sum_{SR \in App} x_{SR,fn}.\, sw_{SR} \le SW_{fmesh}, \forall fmesh \in F \quad (8)$$

$$\sum_{SR \in App} x_{SR,fn}.\, s_{SR} \le S_{fmesh}, \forall fmesh \in F \quad (9)$$

$$x_{SR,fn} \le y_{fn},\ \forall SR \in App,\ fn \in F \quad (10)$$

$$\sum_{fn \in F} x_{SR,fn} = 1\ , \forall SR \in App \quad (11)$$

$$x_{SR,fn} \in \{0,1\},\ y_{fn} \in \{0,1\}, y_{fn,i} \in \{0,1\} \quad (12)$$

Constraint in Eq. (6) determines the distribution and placement of services associated to each application must be in such a way that the bandwidth and delay limitation of the system cannot ruin the application. The terms b_{ij} and l_{ij} are favorite bandwidth and latency between services SR_i and SR_j. Also, parameters B_{mn} and L_{mn} are bandwidth and latency between fog nodes fn_m and fn_n respectively. Eq. (7) indicates that each service must be placed over the fog node which provides the requested hardware. In Eq. (7), parameter HW_{fn} is relevant to fog node capacity in term of hardware and hw_{SR} is requetsed resources relevant to services. Also, the software requested by the application can be provided by underlying full mesh that Eq. (8) shows. In Eq.(8), the term SW_{fmesh} is software capacity of full mesh subnetwork and sw_{SR} is the requested software by application services. In this regards, the considered sub full mesh must provide all the sensors that the application needs; this is well depicted in Eq. (9). In Eq. (9), the term S_{fmesh} is sensor capacity of full mesh subnetwork and s_{SR} is the requested sensor by application services. Eq.(10) means that only an active fog node can adopt a service. For this reason, decision variable y_{fn} is set to one when fog node fn is an active node to adopt a service. Eq. (11) determines each service is only placed on one fog node and final Eq. (12) shows binary decision variables which have been used in aforementioned equations.

5. PROPOSED METHODOLOGY

To solve the formulated optimization problem, an enhanced whale optimization (EWOA) is proposed. This section has two parts. At first the inspiration of WOA is presented and second part elaborates the proposed algorithm.

5-1 Inspiration

Evolutionary computation was inspired from natural phenomenon especially from animal behaviors. In this regards, ant colony optimization (ACO) [21], bee colony optimization (BCO) [22], bat optimization algorithm (BOA) [23], grey wolf optimization (GWO) [24] are the most famous among others; each of which has its merits and demerits. One important thing to mention in evolutionary computation is how to reach of equilibrium between local and global optimization in search space. One of the successful swarm-based algorithm which is inspired its functionality from nature and animal behavior is the whale optimization algorithm (WOA) that was intricately designed to yield balance between exploitation (by utilizing bubble-net attack behavior) and exploration (by utilizing search for pray behavior) in search field [25]. Therefore, its final performance is better than other evolutionary algorithms in which the final simulation outcome endorse it; this is the reason that the paper cornerstone is based on customized WOA.

5-2 Proposed Energy-aware Reliable Service Placement Algorithm based on WOA (ER-SPA-WOA)

This sub section presents proposed ER-SPA-WOA for solving service placement problem in fog platform which optimizes total power consumption subject to some constraints. One importance element of each evolutionary computation is how to encode real world problems in which it has deep effect of overall performance [26-28]. The encoding phase determines how the real world problem (phonotype) is translated to the evolutionary world (genotype). Fig. 6 exemplifies how the real world application containing different services is encoded in whale optimization language. For instance, there are seven requested services unified in an application delivered by a user that must be run on a near fog system with three available fog nodes.

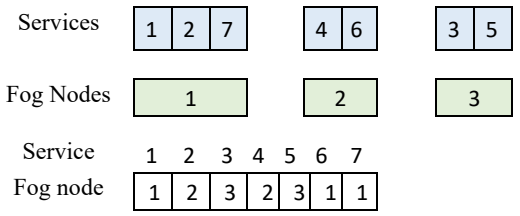


Fig. 6. A Sample of Problem Encoding

The proposed algorithm is depicted in Fig.7. Similar to other meta-heuristic population-based algorithm, the proposed algorithm starts by producing initial population which is placed in the first line of algorithm.

After population is created each whale competency is measured by fitness function in line 2 which adopts Eq.(5). In Eq.(5), the decision variable $y_{fn,i}$ determines whether the fog node fn_i is active or not. At line 3 and at the point population is updates in line 20 the best so far solution is kept in W*. The main loop is iterated MaxIteration times between lines 4-21. In fact, the parameter associated to the maximum number of iterations is the termination criteria. The loop in lines 5-18 is repeated for each whale. In line 6, vectors a, A, C are updated. Note that a is a vector which is gradually decreased from 2 to 0; also, $\vec{r}$ is a random real value in [0..1] interval.

$$\vec{A}=2\vec{a}.\vec{r}-\vec{a} \tag{13}$$

$$\vec{C}=2\,\vec{r} \tag{14}$$

Procedure ER-SPA-WOA
Input:
n number of fog nodes;
m number of services;
PopSize as whale population size;
MaxIteration as maximum number of iteration;
Output:
An optimal service placement

1. Initialize the whale population Pop including W_i (i=1,2,…,*PopSize*)
2. Calculate the fitness fucntion according to Eq. (5).
3. let W^* to be the best whale recognized so far.
4. While iteration is not reach to *MaxIteration* Do
5. for i=1 To *PopSize* Do
6. Update vectors a, A, and C based on Eqs.(13-14) ; and random variable *l*
7. Draw new real random number 0≤P≤1;
8. if (P<0.5) then
9. if (|A| < 1) then
10. Update the whlale W_i position based on Eq. (16)
11. else if (|A| ≥ 1) then
12. Select a random whale W_j from population
13. Update the whale W_i position based on Eq.(18) incorporating W_j
14. end-if
15. else (* P ≥0.5 *)
16. Update the whale W_i position based on Eq.(20)
17. end-if
18. Call Clamping (W_i) to correct the whale position if it returns infeasible solution
19. end-for
20. find current best so far from population and put it to W^*
21. end-while
22. return W^*

Fig. 7. Proposed ER-SPA-WOA Algorithm

In addition the random real value l is randomly drawn from [-1..1] interval. The art of WOA is the fluctuation from exploration to exploitation vice versa during to the whole algorithm life time. To this end, a random variable P is drawn to determine whether to explore or exploit the search space. Update of line 10 in proposed algorithm is dedicated for exploitation.

$$\vec{D} = \left|\vec{C}.\overrightarrow{W*}(t) - \overrightarrow{W_l}(t)\right| \tag{15}$$

$$\overrightarrow{W_l}(t+1) = \overrightarrow{W*}(t) - \vec{A}.\vec{D} \tag{16}$$

In this regards, for exploration the update is done via Eq. (18) which line 13 is dedicated for.

In this case, the update is unbiasedly done based random whale position.

$$\vec{D} = \left|\vec{C}.\overrightarrow{W_J}(t) - \overrightarrow{W_i}(t)\right| \quad (17)$$

$$\overrightarrow{W_i}(t+1) = \overrightarrow{W_J}(t) - \vec{A}.\vec{D} \quad (18)$$

To simulate the special circular movement of whales which is called spiral update position, the algorithm updates selected whale position by Eq. (20).

$$\overrightarrow{D'} = \left|\overrightarrow{W*}(t) - \overrightarrow{W_i}(t)\right| \quad (19)$$

$$\overrightarrow{W_i}(t+1) = \overrightarrow{D'}.e^{bl}.\mathrm{Cos}(2\pi) + \overrightarrow{W*}(t) \quad (20)$$

Once the update is done for all whales, if each whale is infeasible solution the Clamping function is called to correct encoding. Line 19 draws Clamping (.) function. It can be arbitrary implemented. Finally, the best so far whale which is representative of an optimal solution is returned.

6. SIMULATION AND EVALUATION

In this section, the result of simulations are investigated in five different scenarios which has different datasets. In the first and second scenarios the number of services are increased whereas in the third, fourth, and fifth, both number of services and underlying fog nodes are gradually increased. The fifth scenario is dedicated for scalability testing. Table 2 elaborates scenarios in details. All scenarios are executed in fair conditions on a dual core Intel Corei3 380M platform with 2.53 GHZ clock rate, four logical processors, 8 GB as main memory and MATLAB simulation programming environment.

Table 2. Different Scenarios of Simulation

Scenarios #:	1	2	3	4	5
Fog nodes #:	10	10	20	20	50
services #:	10	20	30	40	50

Table. 3. Setting Parameters of Different Algorithms in Simulation

	Specific parameters				Pop Size	Max Iter
EWOA					100	100
GWO						
ACO	tau rho	0.1 0.1	alpha beta	1 0.02		
BCO	a	1				
BOA	r0 alpha1 gamma	1 0.97 0.1	Freq_min Freq_max	0 2		

tau Initial Pheromone
alpha Pheromone Exponential Weight
beta Heuristic Exponential Weight
rho Evaporation Rate
a Acceleration Coefficient
r0 The initial pulse rate
alpha1 loudness decay factor
gamma pulse increase factor
Freq_min Frequency minimum
Freq_max Frequency maximum

To assess comparative algorithms, status figures associated to algorithms' iterations and elapsed time are used in regards to objective function. To construct underlying fog network, the fog nodes' coordinate is made by random normal distribution fashion. The fog nodes distribution plan is then illustrated. Also, the fog nodes which host services associated to IoT applications are depicted by red color.

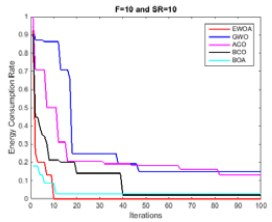
Since there is not any standard benchmark in this domain, the datasets are made based on processing power, fog nodes' storage capacity commensurate with existing fog nodes such as personal digital assistant (PDA), smart phones, tablets, and etc. to be utilized in simulation scenarios.

For simulations and comparisons, parameter settings of algorithms EWOA, GWO, ACO, BCO, and BOA are brought in Table 3.

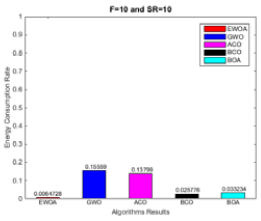
6-1 First Scenario: 10 Fog Nodes and 10 Application Services

In Fig. 8-a, the objective function value, TPC, based on Eq. (5) relevant to comparative algorithms EWOA, GWO, ACO, BCO, and BOA are depicted in a network having 10 fog

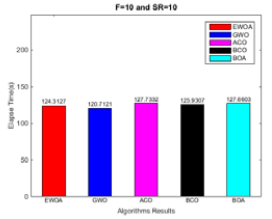
nodes and 10 requested services. According to drawn figure, once the iteration increases, algorithm EWOA outperforms other counterpart algorithms and reaches to stable point. In this regards, Fig. 8-b illustrates better performance in term of minimum value of TPC related to proposed algorithm against other state-of-the-arts. Regarding to Fig. 8-c, the elapsed time of proposed algorithm in the first scenario is less than others which means that it works faster than others. The selected sub full mesh utilizes fog nodes 3, 6, 7, and 10 for service distribution. The position of utilized fog nodes for this requested services are depicted in Fig. 8-d.



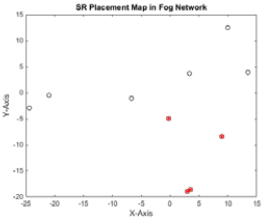
a. Performance evaluation based on TPC



b. The minimum value of TPC after end of simulation



c. Elapsed time of algorithms

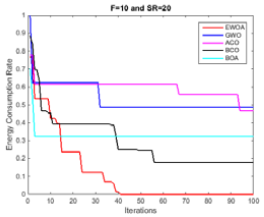


d. Distribution of services and deployment plan

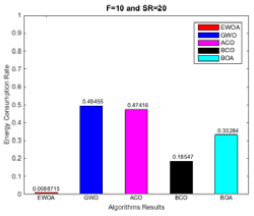
Fig.8. Execution results of comparative algorithms in scenario with 10 fog nodes and 10 requested services

6-2 Second Scenario: 10 Fog Nodes and 20 Application Services

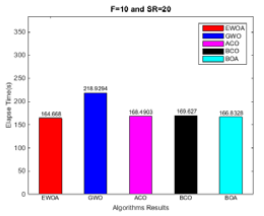
In Fig. 9-a, the objective function value, TPC, based on Eq. (5) relevant to comparative algorithms EWOA, GWO, ACO, BCO, and BOA are depicted in a network having 10 fog nodes and 20 requested services. According to drawn figure, once the iteration increases, algorithm EWOA outperforms other counterpart algorithms and reaches to stable point. In this regards, Fig. 9-b illustrates better performance in term of minimum value of TPC related to proposed algorithm against other state-of-the-arts. Regarding to Fig. 9-c, the elapsed time of proposed algorithm in the first scenario is less than others which means that it works faster than others. The selected sub full mesh utilizes fog nodes 1, 2, 3, and 10 for service distribution. The position of utilized fog nodes for this requested services are depicted in Fig. 9-d.



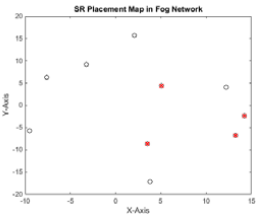
a. Performance evaluation based on TPC



b. The minimum value of TPC after end of simulation



c. Elapsed time of algorithms

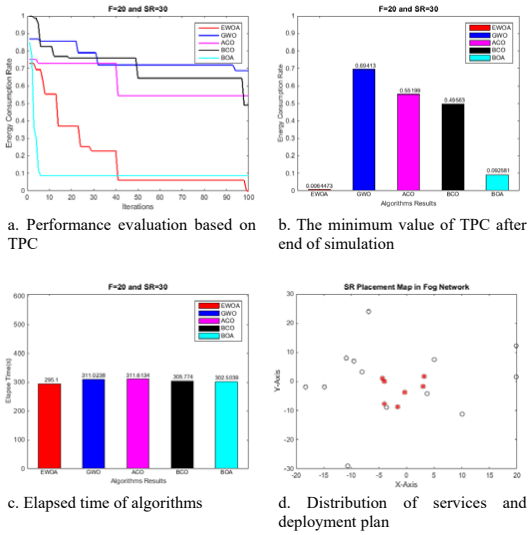


d. Distribution of services and deployment plan

Fig.9. Execution results of comparative algorithms in scenario with 10 fog nodes and 20 requested services

6-3 Third Scenario: 20 Fog Nodes and 30 Application Services

In Fig. 10-a, the objective function value, TPC, based on Eq. (5) relevant to comparative algorithms EWOA, GWO, ACO, BCO, and BOA are depicted in a network having 20 fog nodes and 30 requested services. According to drawn figure, once the iteration increases, algorithm EWOA outperforms other counterpart algorithms and reaches to stable point. In this regards, Fig. 10-b illustrates better performance in term of minimum value of TPC related to proposed algorithm against other state-of-the-arts. Regarding to Fig. 10-c, the elapsed time of proposed algorithm in the first scenario is less than others which means that it works faster than others. The selected sub full mesh utilizes fog nodes 2, 3, 4, 6, 8, 10, and 13 for service distribution. The position of utilized fog nodes for this requested services are depicted in Fig. 10-d.



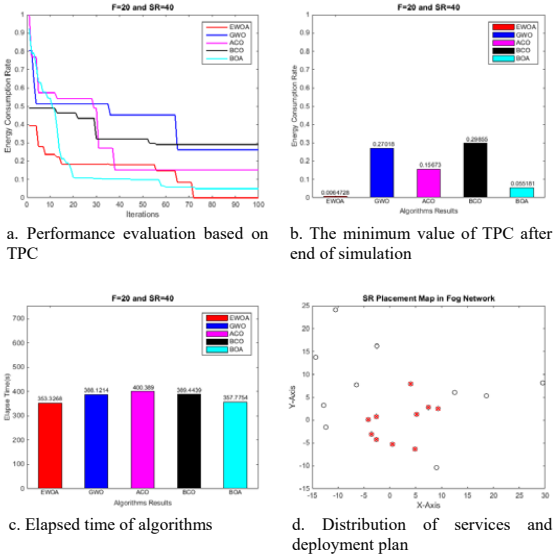
a. Performance evaluation based on TPC

b. The minimum value of TPC after end of simulation

c. Elapsed time of algorithms

d. Distribution of services and deployment plan

Fig. 10. Execution results of comparative algorithms in scenario with 20 fog nodes and 30 requested services



a. Performance evaluation based on TPC

b. The minimum value of TPC after end of simulation

c. Elapsed time of algorithms

d. Distribution of services and deployment plan

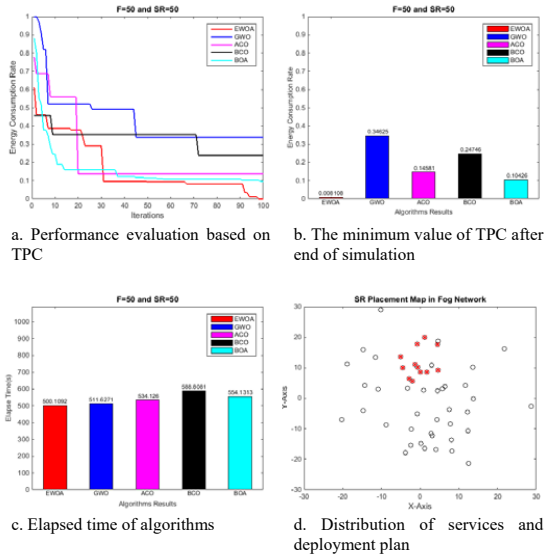
Fig. 11. Execution results of comparative algorithms in scenario with 20 fog nodes and 40 requested services

6-4 Fourth Scenario: 20 Fog Nodes and 40 Application Services

In Fig. 11-a, the objective function value, TPC, based on Eq. (5) relevant to comparative algorithms EWOA, GWO, ACO, BCO, and BOA are depicted in a network having 20 fog nodes and 40 requested services. According to drawn figure, once the iteration increases, algorithm EWOA outperforms other counterpart algorithms and reaches to stable point. In this regards, Fig. 11-b illustrates better performance in term of minimum value of TPC related to proposed algorithm against other state-of-the-arts. Regarding to Fig. 11-c, the elapsed time of proposed algorithm in the first scenario is less than others which means that it works faster than others. The selected sub full mesh utilizes fog nodes 4, 5, 7, 8, 9, 14, 16, 17, and 20 for service distribution. The position of utilized fog nodes for this requested services are depicted in Fig. 11-d.

6-5 Fifth Scenario: 50 Fog Nodes and 50 Application Services

In Fig. 12-a, the objective function value, TPC, based on Eq. (5) relevant to comparative algorithms EWOA, GWO, ACO, BCO, and BOA are depicted in a network having 50 fog nodes and 50 requested services. According to drawn figure, once the iteration increases, algorithm EWOA outperforms other counterpart algorithms and reaches to stable point. In this regards, Fig. 12-b illustrates better performance in term of minimum value of TPC related to proposed algorithm against other state-of-the-arts. Regarding to Fig. 12-c, the elapsed time of proposed algorithm in the first scenario is less than others which means that it works faster than others. The selected sub full mesh utilizes fog nodes 8, 11, 14, 15, 24, 26, 29, 30, 33, 35, 41 and 43 for service distribution. The position of utilized fog nodes for this requested services are depicted in Fig. 12-d.



a. Performance evaluation based on TPC

b. The minimum value of TPC after end of simulation

c. Elapsed time of algorithms

d. Distribution of services and deployment plan

Fig. 12. Execution results of comparative algorithms in scenario with 50 fog nodes and 50 requested services

For the sake of data analysis statistically, the proposed EWOA outperforms 48%, 36%, 30% and 15% improvement against GWO, ACO, BCO and BOA in term of average minimum value of power consumption.

6-6 Time Complexity

The time complexity of the proposed algorithm can be simply obtained. The fitness value, which Eq. (5) gives, takes $O(m+n)$ where n and m are number of fog nodes and number requested services respectively. In addition, the main loop of algorithm between lines (4-21) is repeated *MaxIteration* times. Inner for-loop which was placed between lines (5-18) is iterated *PopSize* times where *PopSize* indicates the number of whales in the population.

Overall, the time complexity of proposed algorithm belongs to $O(MaxIteration*PopSize*(m+n))$.

7. CONCLUSION AND FUTURE WORK

This paper formulated the service placement problem associated to IoT applications over fog nodes to an optimization problem with minimization of total power consumption of engaged fog nodes. The power consumption model was presented where is affected by underlying resource utilization and data transfer between each pair of services provided they are deployed on different fog nodes. To avoid one point of failure, the requested services delivered by users are distributed on dispersed fog nodes to fulfil reliability requirement. To figure out this problem, an energy-aware reliable service placement algorithm based on whale optimization algorithm (ER-SPA-WOA) was extended. It has been tested in different scenarios in fair conditions. The simulation results proved the dominance of proposed ER-SPA-WOA against other comparative state-of-the-arts. For future work, we envisage to model mobile computing in fog-cloud environment by taking QoS and monetary cost into account.

REFERENCES

1. Azimi Sh, Pahl C, Hosseini Shirvani M. Particle Swarm Optimization for Performance Management in Multi-cluster IoT Edge Architectures. International cloud computing conference CLOSER. 2020; 328-337. http://dx.doi.org/10.5220/0009391203280337.

2. Karimi M. B, Isazadeh A, Rahmani A. M. QoS-aware service composition in cloud computing using data mining techniques and genetic algorithm. The Journal of Supercomputing. 2017; 73(4):1387–1415. https://doi.org/10.1007/s11227-016-1814-8.

3. Hosseini Shirvani M. Bi-objective web service composition problem in multi-cloud environment: a bi-objective time-varying particle swarm optimisation algorithm. J Exp Theor Artif Intell. 2020; 33(2):179-202. https://doi.org/10.1080/0952813X.2020.1725652.

4. Hosseini Shirvani M, Babazadeh Gorji A. Optimisation of automatic web services composition using genetic algorithm. Int J Cloud Comput. 2020; 9(4):397–411. https://dx.doi.org/10.1504/IJCC.2020.112313

5. Ramzanpoor Y, Hosseini Shirvani M. Multi-objective QoS-aware Optimization for Deployment of IoT Applications on Cloud and Fog Computing Infrastructure. Cluster Computing. 2021; Under Review.

6. Ramzanpoor, Y., Hosseini Shirvani, M. & Golsorkhtabaramiri, M. Multi-objective fault-tolerant optimization algorithm for deployment of IoT applications on fog computing infrastructure. Complex Intell. Syst. (2021). https://doi.org/10.1007/s40747-021-00368-z.

7. Farzai S, Hosseini Shirvani M, Rabbani M, Multi-Objective Communication-Aware Optimization for Virtual Machine Placement in Cloud Datacenters. Sustainable Computing: Informatics and Systems. 2020; 28. https://doi.org/10.1016/j.suscom.2020.100374.

8. Foukalas F. Cognitive IoT platform for fog computing industrial applications. Computers and Electrical Engineering. 2020; 87: 1-13. https://doi.org/10.1016/j.compeleceng.2020.106770

9. OpenFog. An OpenFog Architecture Overview. https://www.iiconsortium.org/pdf/OpenFog_ Reference_ Architecture_2_09_17.pdf. Accessed February, 2017.

10. Brogi A, Forti A. QoS-aware Deployment of IoT Applications Through the Fog. IEEE Internet of Things Journal. 2017; 4:1185-1192. https://doi.org/10.1109/JIOT.2017.2701408.

11. Taneja M, Davy A. Resource-aware Placement of IoT Application Modules in Fog-Cloud Computing Paradigm. in Proc. of the IFIP/IEEE Symposium on Integrated Network and Service Management. IM '15. IEEE. 2017; 1222–1228. https://doi.org/10.23919/INM.2017.7987464.

12. Li F, Vogler M, Claeßens M, Dustdar S. Towards automated iot application deployment by a cloud-based approach. in 6th International Conference on Service-Oriented Computing and Applications. IEEE. 2013; 61–68. https://doi.org/10.1109/SOCA.2013.12.

13. Mahmud R, Ramamohanarao K, Buyya R. Latency-aware application module Management for fog Computing Environments. ACM Transactions on Internet Technology. 2018; 1–21. https://doi.org/10.1145/3186592.

14. Vögler M, Schleicher J. M, Inzinger C, Dustdar S. DIANE - Dynamic IoT Application Deployment. IEEE International Conference on Mobile Services. 2015; 298-305. https://doi.org/10.1109/MobServ.2015.49.

15. Yousefpour A, Patil A, Ishigaki G, Kim I, Wang X, Cankaya H. C, Zhang Q, Xie W, Jue J. P. Fogplan: A lightweight qos-aware dynamic fog service provisioning framework. IEEE Internet of Things Journal. 2019; 6(3): 5080 – 5096. https://doi.org/10.1109/JIOT.2019.2896311.

16. Canali C, Lancellotti R. Gasp: Genetic algorithms for service placement in fog computing systems. Algorithms. 2019; 12(10): 201. https://doi.org/10.3390/a12100201.

17. Azizi S, Khosroabadi F, Shojafar M. A priority-based service placement policy for fog-cloud computing systems. Computational Methods for Differential Equations. 2019; 7(4):521–534.

18. Guerrero C, Lera I, Juiz C. Evaluation and efficiency comparison of evolutionary algorithms for service placement optimization in fog architectures. Future Generation Computer Systems. 2019; 97: 131–144. https://doi.org/10.1016/j.future.2019.02.056.

19. Hosseini Shirvani M. Web service composition in multi-cloud environment: a bi-objective genetic optimization algorithm. In 2018 innovations in intelligent systems and applications (INISTA). IEEE. 2018; pp 1–6. https://doi.org/10.1109/INISTA.2018.8466267.

20. Arcangeli J. P, Boujbel R, Leriche S. Automatic deployment of distributed software systems: Definitions and state of the art. The Journal of Systems and Software. January 2015; 3:198-218. https://doi.org/10.1016/j.jss.2015.01.040.

21. Dorigo M. Optimization, Learning and Natural Algorithms. PhD thesis, Politecnico di Milano. Italy. 1992.

22. Teodorović D. Bee Colony Optimization (BCO). In: Lim C.P., Jain L.C., Dehuri S. (eds) Innovations in Swarm Intelligence. Studies in Computational Intelligence. Springer. 2009; 248, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-04225-6_3.

23. Yang X. S. A New Metaheuristic Bat-Inspired Algorithm, in: Nature Inspired Cooperative Strategies for Optimization. Studies in Computational Intelligence. 2010; 284: 65–74. https://doi.org/10.1007/978-3-642-12538-6_6.

24. Mirjalili S, Mirjalili S. M, Lewis A. Grey wolf optimizer. Advances in Engineering Software. 2014; 69: 46-61. https://doi.org/10.1016/j.advengsoft.2013.12.007.

25. Mirjalili S, Lewis A. The whale optimization algorithm. Advances in Engineering Software. 2016; 95: 51-67. https://doi.org/10.1016/j.advengsoft.2016.01.008.

26. Saeedi, P, Hosseini Shirvani M. An improved thermodynamic simulated annealing-based approach for resource-skewness-aware and power-efficient virtual

machine consolidation in cloud datacenters. Soft Comput. 2021. https://doi.org/10.1007/s00500-020-05523-1.

27. Hosseini Shirvani M. A hybrid meta-heuristic algorithm for scientific workflow scheduling in heterogeneous distributed computing systems. Engineering Applications of Artificial Intelligence. 2020; 90:1–20. https://doi.org/10.1016/j.engappai.2020.103501.

28. Javadian Kootanaee, A, Poor Aghajan A, Hosseini Shirvani M. A hybrid model based on machine learning and genetic algorithm for detecting fraud in financial statements. Journal of Optimization in Industrial Engineering. 2021; 14(2):183-201. doi: 10.22094/joie.2020.1877455.1685.

29. Azimi, S., Pahl, C., Hosseijni Shirvani, M.: Performance Management in Clustered Edge Architectures Using Particle swarm optimization. In: Cloud Computing and Services Science. 2021; 1399: 233–257.