Received: March 21, 2017

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

An Improved Fruit Fly Optimization Algorithm for QoS Aware Cloud Service Composition

Bharath Bhushan Savarala[1*] **Pradeep Reddy Chella[1]**

[1]*VIT University, Vellore, India*
* Corresponding author's Email: bharath.bhushan4@gmail.com

Abstract: An improved fruit fly optimization algorithm based on discrete immune optimization is proposed for quality of service (QoS) aware cloud service composition. The selection and composition of cloud services based on QoS criteria is formulated as NP hard optimization problem. We determined pareto optimal service set which is non-dominated solution set as input to the improved fruit fly optimization algorithm. A mathematical model is derived to enhance local search capabilities and also improves the fitness value of composite service sequence. The fruit fly optimization (FOA) performs the evolutionary search process and enhances the convergence speed with good fitness value. The experimental results show that the improved FOA outperforms the genetic algorithm (GA), particle swarm optimization (PSO) and hybrid particle swarm optimization (HPSO) in terms of fitness value, execution time and error rate.

Keywords: Service composition, Cloud computing, Fruit fly optimization, Immune optimization, Quality of service.

1. Introduction

Cloud computing is a leading platform for providing elastic web services, which helps for the seamless composition of enterprise applications to create new value added services [1]. Today, users and enterprises are increasingly using the cloud to access software resources in the form of web services [2]. As web services are self-contained, loosely coupled processes deployed over standard middleware platforms that can be described, published, discovered and invoked over a network.

The different service providers will publish a large number of similar services with different quality of service in various clouds [3]. QoS parameters can be used to select the best service among similar services and will make a QoS aware service composition [4], which is an MCDM (multi criteria decision making) problem. The traditional service composition methods compose services from single cloud, which won't always meet user requirements. There is a need to compose the services from multiple clouds [5,6] by taking advantage of different QoS levels.

In this paper, we consider QoS parameters such as availability, reliability, price, response time, throughput and reputation [7]. Many researchers have identified that QoS parameter value should be real number, language value and interval. The main objective is to find the best service composition [8] by taking QoS parameters into consideration with minimal number of clouds. As the services that are communicating from multiple clouds are expensive and time consuming.

The various service providers will publish a large number of similar services which have different levels of quality of service. As QoS aware service composition is a hot research topic in the field of cloud computing and networks. The improper selection and composition of services can affect the service level agreements and which leads to user dissatisfaction. Service composition is a three step process: (1) Handling user composite request (2) Service request (3) Composing selected services. In the first step, the user submits his preferences in terms of weights. In the second step, the proposed approach will select the best service from several possible similar services and those

composed results will return to the user in the final step.

In past few years, the researchers proposed various techniques to solve the QoS aware service composition problem, but user dissatisfaction or violation of service level agreements (SLA) is major problem. Linear integer programming (LIP) model is adopted by few scholars to solve service matching, ranking and selection problem by maximizing the objective value. As the number of candidate services increases the linear integer programming model suffers from increase in computational cost. As our problem is multi-constraint multi objective problem, the effective and efficient particle swarm optimization algorithm is applied to solve it. PSO has been applied in various ways to solve the service selection problem, but it attains sub-optimal solution because of its premature convergence nature [9]. So, to improve the accuracy of PSO the fruit fly optimization algorithm is incorporated.

The artificial, evolutionary and classic algorithms have been used extensively to solve service composition problem [10,11] But the traditional approaches follow the global optimization approach to solve service composition problem by considering all possible candidate service combinations. This approach attains optimal combination with decent fitness value and execution time by meeting global constraints, but still it suffers from poor performance.

In this paper, we proposed an improved FOA based on immune optimization. First, we generated initial population which are initial optimal weights for the resultant services from pareto applied on initial candidate services. Then it finds an optimal composed service based on user preferences attains good fitness value, execution time and error rate. In second phase, we applied vision and smell operators of FOA to achieve better optimal weights. Finally, the experimental results of proposed discrete immune fruit fly optimization algorithm (DIFOA) are more effective than the simple GA, PSO and HPSO.

The contribution of this research as follows:

1. It reduces number of candidate services
2. A significant increase in fitness value
3. It reduces the root mean square error with decent execution time
4. It will ensure optimal service selection with minimal SLA violations

The rest of the paper is organized as follows: related work about the QoS aware service composition in section.2. In section.3 the problem is formally defined. Section.4 focuses on our proposed DIFOA method: simple immune optimization and improved FOA in detail, including the fitness evaluation, initial weights and the applied vision, smell operators of FOA. Experimental results and discussion are presented in section.5. Finally, we concluded our work in section.6.

2. Related work

Service selection from a minimal number of clouds has been an important issue for service composition. This is because any improper selection of the service can affect the overall QoS of a composite service and leads to user dissatisfaction and number of clouds leads to expensive and time consuming. Researchers have adopted different approaches to compose the best services from a minimal number of clouds.

In [12] Zeng et al., proposed a QoS aware middleware to solve global optimization problem in web service composition, which is based on linear integer programming. However, if number of candidate services increases the model suffers from increase in computation cost. The authors proposed a model to reduce the computational cost by finding best web services based on mixed integer programming and local selection method. But it fails to handle the different services with different QoS values [13]. A multi-objective optimization framework is proposed which is based on genetic algorithm to solve QoS aware service composition problem. The disadvantage is slower than integer programming and not suitable for large scale applications.

The authors proposed an algorithm [14], which evaluates all possible combinations by using depth first search algorithm and its preferences is improved by using a pruning technique. In [15] an artificial intelligence based service composition algorithm is proposed to maximize the fitness value and to minimize the cost of composition. The algorithm fails to achieve optimal solution within acceptable time. The authors applied pruning and branch and bound methods to their proposed linear integer programming approach in order to maximize the fitness value with good aggregate QoS value.

In [16] QoS aware service composition architecture is proposed based on tabu search and simulated annealing. The simulated annealing optimizes the service execution plan by which execution time is minimized. A single objective particle swarm optimization algorithm to avoid local optima. A pareto optima is applied to prune dominated service instance and then a muti objective PSO algorithm is applied on resultant candidate services by which an improved fitness

value, execution time is achieved. A two phase mixed integer programming model is implemented in [17]. In first phase, k-means is applied to reduce number of candidate services and then mixed integer programming is used to select the suitable service from reduced candidate set.

Another two phase method is proposed by [6], where the composite services are represented as state transition matrix. In second phase, to find optimal services the business process execution language for web services (BPEL4WS) is applied. A three level hierarchical model is presented in [18], in which QoS weights are generated by analytical hierarchy process (AHP). In first layer the optimal service selection is done based on user preferences. The next two layers are represented as criterion layer and QoS parameter layer. In [19] they proposed three algorithms to reduce search space and generate dominated skyline services. The first algorithm finds a best solution from all execution plans with improved execution time and consumed space. The second algorithm evaluates all service execution plans by adopting tree structure with pipeline capabilities. The third algorithm overcomes the expensive computational cost of second algorithm by preparing bottom up algorithm.

In [20] the author proposed an efficient genetic algorithm to select individuals for the crossover and mutation operators by replacing tournament selection algorithm with clonal selection algorithm. A skyline operator and integer array coding is applied on particle swarm optimization to achieve service composition with minimal execution time within reduced search space. As the number of services increases, the model suffers from increase in computational cost. In [21,22] they proposed an imperialist competitive gravitational attraction algorithm to minimize the response time and computation cost. However, they adopted exploration capability of an imperialist competitive algorithm to address similar services with different QoS level.

Therefore, most of the meta-heuristic approaches suffer from sub optimal solutions because of its premature convergence behaviour and large search space. So, we proposed an improved particle swarm optimization by hybridizing the smell and vision operators of fruit fly optimization algorithm. The objective is to achieve better fitness value, execution time and minimal error rate.

3. FOA, Immune optimization algorithm and Pareto optimal solution

3.1 Fruit fly optimization

The FOA is a meta heuristic global optimization algorithm, which was inspired by the foraging behaviour of the fruit fly. The fruit fly finds the best food source locations by using unique vision and smell based search operators. The basic FOA flowchart is shown in Fig.1. At first an initial population is randomly generated and evaluates smell concentration value for each fruit fly. Then fly towards the best location by using vision based search process. The process is repeated until the stopping criteria meets.

3.2 Immune optimization algorithm

In this section, we concentrate on some aspects of the IS and is shown in Fig.2. IS is a complex of cells, molecules, and organs that represent an identification mechanism capable of Perceiving and combating dysfunction from our own cells and the action of exogenous infectious microorganisms [23].

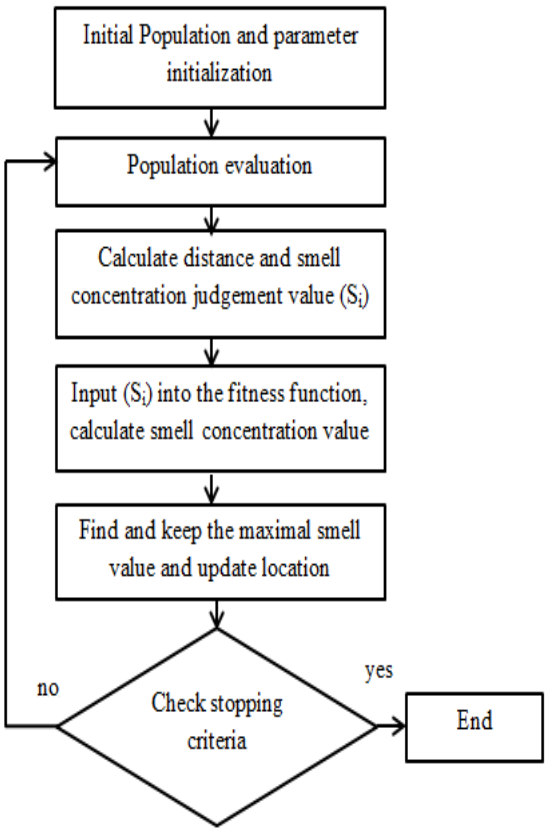

Figure.1 Flowchart of simple fruit fly optimization algorithm

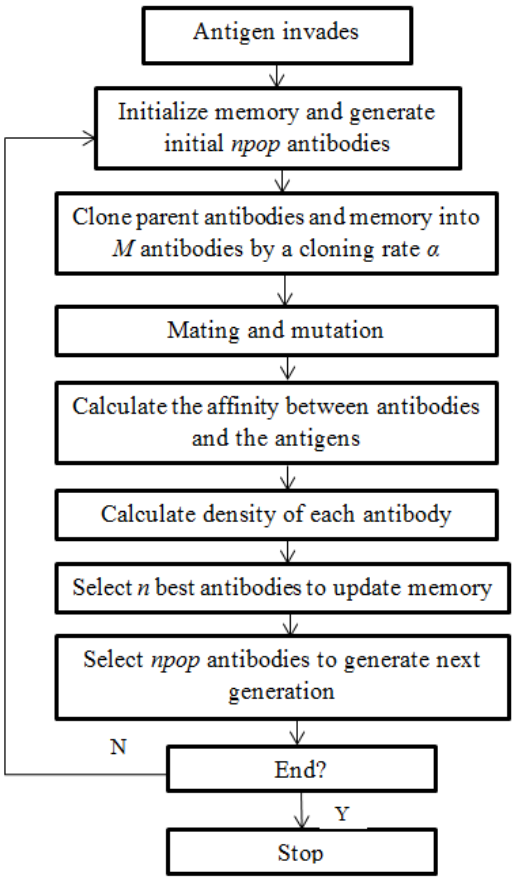

Figure.2 Flowchart of simple immune optimization algorithm

The interaction among the IS and several other systems and organs allows the regulation of the body, guaranteeing the stable functioning of the body [24]. The innate IS and the adaptive IS are two inter-related systems, which can identify a foreign molecule [i.e., antigen (Ag)] from a bacterium or other invaders. In the innate IS, specialized Ag presenting cells (APCs), such as macrophages, roam the body, ingest, and digest the Ags that they find and then fragment these Ags into gene segments which are later attached to major histocompatibility complex (MHC) molecules and manifested on the surface of MHC as peptide-MHC combinations. However, T-cells in IS have receptor molecules that enable each of them to recognize a different peptide-MHC combination. T-cells, activated by that recognition, divide and secrete lymphocytes or chemical signals.

The adaptive IS uses somatically generated Ag receptors that are clonally distributed on the two types of lymphocytes: 1) B-cells and 2) T-cells. The B-cells, which also have receptor molecules of a single specificity on their surface, respond to those signals. Unlike the receptors of T-cells, however, those of B-cells can recognize parts of Ags freely in solution, without MHC molecules. Each B-cell produces a specific antibody (Ab), which can recognize and bind to an Ag. Affinity measures the ability of an Ab to bind an Ag, the higher the affinity is, the more easily the Ag is neutralized. In order to neutralize diverse Ags, the activated B-cells divide and differentiate into a large number of identical B-cells. This process is called affinity maturation. Maturated B-cells turn into plasma cells that secrete Ab proteins to fight off the Ags, which are soluble forms of their receptors. Some T-cells and B-cells become memory cells that persist in the circulation and boost the IS readiness to eliminate the same Ag if it presents itself in the future.

3.3 Pareto optimal solution

To reduce the search space of candidate services, we applied a pareto ranking approach which utilizes the concept of pareto dominance in allocating selection probability to a solution set. The size of pareto optimal set is varied with increase in the number of objectives. The QoS values of candidate services are ranked according to a dominance rule. Then based upon their fitness value the ranks are assigned and treated as a better solution. If a solution is not dominated by any other solution in search space then it is pareto optimal and all non-dominated solutions in a set is referred as pareto optimal set. The corresponding value of an pareto optimal in a search space is referred as pareto front. The adopted pareto ranking approach is shown below

Pareto optimal algorithm for dominated candidate service selection

```
current-rank:=1
p:=(QoS values of candidate services)
n:=p
while p!=0 { /* process entire QoS values of candidate services */
for i:=1 to n { /* find services in current rank
if vi is non-dominated {
rank (vi):=current-rank
}
}
for i:=1 to n { /* remove ranked services from candidate services */
if rank (vi):=current-rank {
remove vi from candidate services
p:=p-1
}
}
current-rank:=current-rank+1
```

n:=p
}

4. The improved fruit fly optimization algorithm for QoS aware cloud service composition

In this paper, we proposed DIFOA for service composition and their components are explained in detail in the following sub sections

4.1. Normalization

The values of various QoS parameters have different dimensions. So, the QoS parameters need to be quantified to have a uniform distribution. The QoS parameters are classified into positive and negative criteria based on their impact on classification function. The positive and negative parameters are normalized by Eq. (1) and Eq. (2) as below

$$N_{ij}^{+} = \frac{q_{ij} - q_j^{min}}{q_j^{max} - q_j^{min}} \quad if\ q_j^{max} - q_j^{min} \neq 0$$

$$1 \qquad if\ q_j^{max} - q_j^{min} = 0 \qquad (1)$$

$$N_{ij}^{-} = \frac{q_j^{min} - q_{ij}}{q_j^{max} - q_j^{min}} \quad if\ q_j^{max} - q_j^{min} \neq 0$$

$$1 \qquad if\ q_j^{max} - q_j^{min} = 0 \qquad (2)$$

Where N_{ij} is the normalized value of j^{th} parameter of the i^{th} service. Where N_{ij} is the maximum value of the j[th] column of the QoS matrix, $q_j^{max} = max(q_{i,j}), 1 \leq i \leq n$ and N_{ij} is the minimum value of the j[th] column of the QoS matrix, if $q_j^{min} = min(q_{i,j}), 1 \leq i . q_{min}$ denotes minimum QoS value and q_{max} represents the maximum QoS value. After normalization, all the QoS value are lies between [0, 1] interval.

4.2. Coding strategy

We adopted fixed length integer coding strategy, where a task number is given to each abstract service and a candidate number is given to every concrete service in each abstract service. Let us assume there are five abstract services in the given composite service, so the length of the position is five and ten concrete services for each abstract service, and then the composed service is (4, 1, 3, 2, 1). The composed scheme shows that the task 1 selects the number 4 service and task 2 selects the number 1 service and so on, which is represented in the below Fig.3.

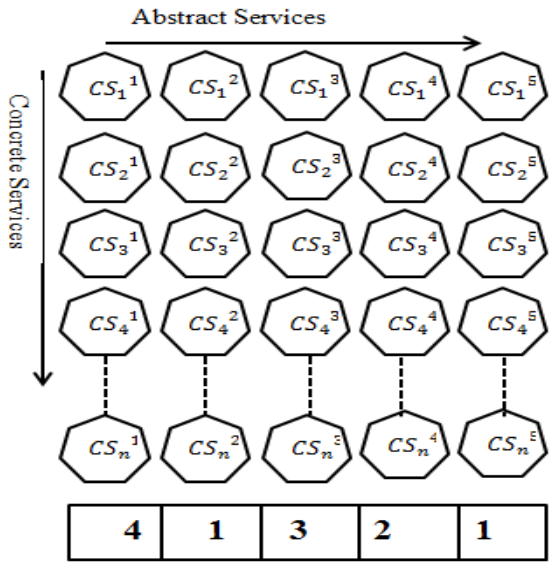

Figure.3 Array integer coding solution

4.3. Fitness function

Fitness function is used to evaluate the performance of each service composition. The main objective is to select one atomic service from candidate services of each abstract service. So, the aggregate QoS value of composite service can be minimized. To solve QoS aware service composition problem, a weighted sum of each parameter of a composite service is adopted as below, where w_i denotes weight of parameter q_i.

$$f(x) = \sum_{i=1}^{6} w_i q_i \qquad (3)$$

4.4. Immune optimization phase

In immune optimization, each antibody position represents a possible solution of optimization problem. In DIFOA, we first generate 100 random weights for each and every service from the resultant pareto set. Then evaluate the performance of each antibody from its current position $x_t{}^i$ and by the fitness function in Eq. (3). The important

operations involved in immune optimization are described below:

Suppose $X_i = (X_{i1}, X_{i2}, X_{i3}.....X_{in})$ is the position vector of a particle i. X_{ij} is an integer ranging from 0 to N_j. where N_j is the number of candidate services in the task j. $F(X_i)$ is the fitness of particle X_i. $V_i = (V_{i1}, V_{i2}, V_{i3}.....V_{in})$ represents the velocity of particle X_i. V_{ij} has the value of 0 or 1.

The probabilities of a particle to sustain its velocity, to the local optimal particle P_i and to the global optimal particle P_g during the velocity update process. The velocity update formula is expressed as follows:

$$V_{i+1} = P_1 V_i \oplus P_2(P_i \ominus X_i) \oplus P_3(P_g \ominus X_i) \quad (4)$$

Antigen and Antibody: In immunology, an antigen is any substance that causes the immune system to produce antibodies against it. In this paper, antigen stands for the optimization problem and antibody refers to the decision vector. The term of "antibody" and "particle" are used indistinctively which come from the areas of IS and PSO.

Crowd Distance: The crowd distance between two antibodies is defined as the square root of the difference vector of the QoS values of two antibodies. The square root of five dimensional difference vector for QoS are calculated as the crowd distance of two antibodies.

Affinity: Affinity is used to evaluate the cloud service composition on fitness value and crowd distance to the optimal solution is calculated by Eq. (5).

$$Affinity_i = {fitness_i}/{\rho_i + 0.1} \quad (5)$$

Where ρ_i stands for the distance between antibodies i and the global optimal antibody. Antibody with higher fitness and shorter distance has higher affinity.

Clone Proliferation: Clonal selection theory states that a clonal expansion of the original lymphocyte occurs when the original lymphocyte is activated by binding to the antigen. The affinity maturation is triggered when the average affinity increases with B cell clonal expansion. The clonal with high affinity is cloned more times, the cloning copied are decided by below Eq. (6).

$$num_i = [\frac{affinity_i}{\sum_{i=1}^{m} affinity_i} \times m_j] \quad (6)$$

Hyper mutation: Affinity maturation is caused by a somatic hyper mutation and a selection mechanism. Somatic hyper mutation results in the diversity of antibodies by introducing random changes to the genes that encode for them. The selection mechanism guarantees that only those clones with higher affinities for the encountered antigen will survive.

4.5. Fruit fly optimization phase

In FOA phase, we have defined a position equation to which we added velocity vector from immune optimization algorithm as shown in Eq. (7).

$$X_t^{i+1,p} = X_i \otimes V_{i+1} \quad (7)$$

The $x_t^{i+1,p}$ represents the location of the t^{th} particle of the p^{th} swarm in the $(i+1)^{th}$ iteration process. V_{i+1} is the velocity update formula. Then the fitness function is executed after calculating smell concentration by using the relation between distance and smell operators as shown in below Eq. (8) and Eq. (9).

$$Dist_t^{i+1,p} = {1}/{x_t^{i+1,j}} \quad (8)$$

$$smell_t^{i+1,p} = {1}/{Dist_t^{i+1,p}} \quad (9)$$

$Dist_t^{i+1,p}$ denotes the distance of the t^{th} particle of the p^{th} swarm in the i^{th} iteration process and $smell_t^{i+1,p}$ represents the smell concentration value of the t^{th} particle of the p^{th} swarm in the $(i+1)^{th}$ iteration process.

Finally, we obtained an optimal composite service with better fitness value and minimal error rate over simple GA, PSO and HPSO based QoS aware service composition.

4.6. The improved fruit fly optimization algorithm based on discrete immune algorithm (DIFOA)

In this subsection, the improved fruit fly optimization algorithm for QoS aware cloud service composition is constructed based on clonal selection and immune algorithm. The algorithm is implemented as follows:

The following is the detailed steps of the proposed DIFOA

Step 1: Initialization Phase

1. Set the parameters such as initial population A(0), initial velocity (v), Iterations (Mgen).

2. Generate the initial population and obtain the candidate services from pareto optimal solution.
3. The fitness value is calculated for each solution in the initial population by using Eq. (3)

Step 2: Terminate the DIFOA algorithm and return the best solution, if the stopping criterion is satisfied; otherwise proceed from step.3 to step.6.

Step 3: Immune Optimization Phase

The new population is generated by using the current population and the velocity and position is obtained by following steps:

1. Maturation: Calculate the affinities of all the antibodies by using Eq. (5) to obtain A (k).
2. Proliferation: The clonal proliferation is applied on A (k) by using Eq. (6) to obtain updated population B (k)
3. Hyper mutation: On B (k) the hyper mutation is applied to obtain c (k).
4. The position of a particle is updated by calculating the velocity using Eq. (4).

Step 4: FOA Phase

1. The motion equation is proposed to update position of a particle with the velocity generated from Immune optimization phase by using Eq. (7).
2. Apply the distance and smell operator on each particle by using the Eq. (8) and Eq. (9).

Step 5: Return back to step.2 to check the stopping criteria.

5. Experimental results

The proposed DIFOA algorithm has undergone a series of experiments to test the effectiveness and its performance. In our experiments, we considered four abstract and 27 related concrete services. For each concrete cloud service, we have considered six QoS parameters in which four are positive parameters (availability, reliability, throughput and reputation) and two are negative parameters (price, response time) and their values are taken from QWS dataset. The proposed DIFOA, HPSO and simple PSO are coded on MATLAB R2014a using HP Pavilion dv6 laptop with 2.10 Ghz Intel core processor and 2GB RAM. The experiments are conducted to test the feasibility, efficiency, stability and then the results were compared with simple PSO GA and HPSO.

In order to evaluate the global searching capability of the proposed HPSO algorithm, three scenarios are considered. In first scenario, the number of abstract services is four and number of concrete services varies from 30 to 120 with an increment of 30. In second scenario, the fitness value is compared by keeping the same number of iterations (1000) over different population size varies from 100 to 1000 with an increment of 100. In third scenario, by keeping the same number of population (100) over different iteration size varies from 100 to 1000 with an increment of 100.

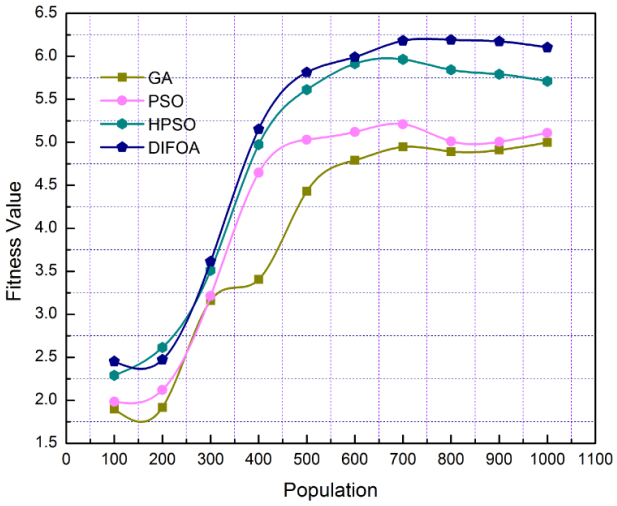

Figure.4 Optimality comparison over different population size and the same number of iterations

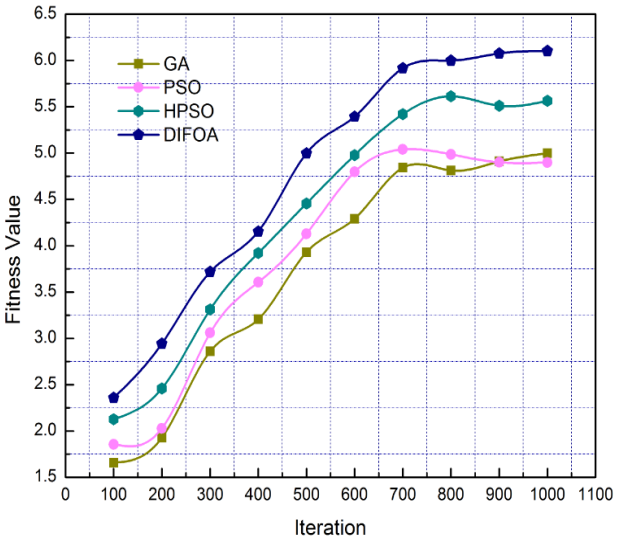

Figure.5 Optimality comparison over same population size and the different number of iterations

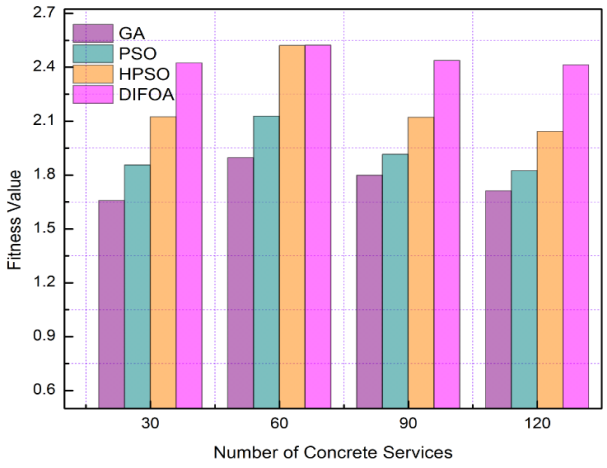

Figure.6 Optimality comparison over different service scale

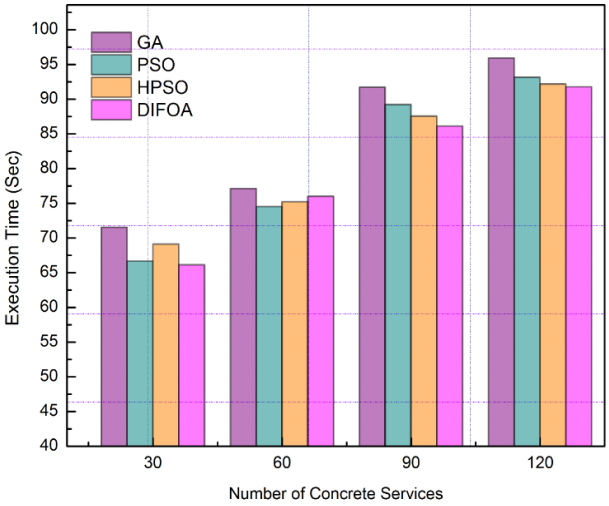

Figure.7 Execution time comparison over different service scale

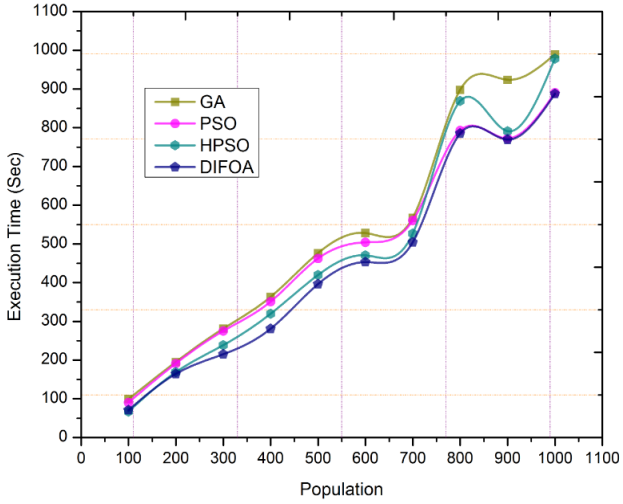

Figure.8 Execution time comparison over different population size and the same number of iterations

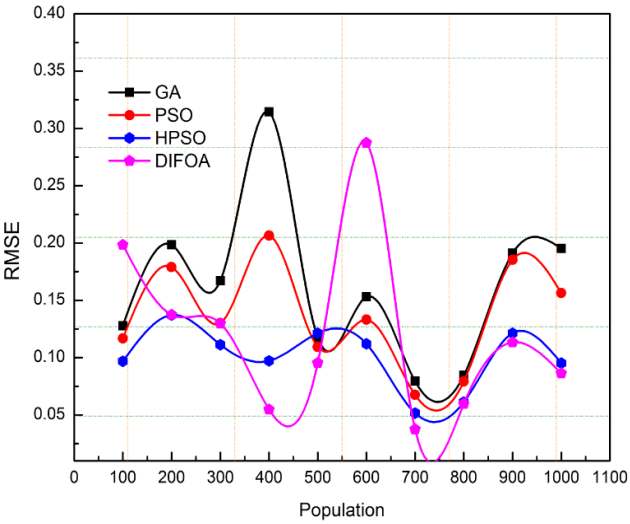

Figure.9 Analysis of RMSE over different population size and the same number of iterations

From Fig.4-6 it is clear that the proposed DIFOA algorithm outperforms the simple GA [20], PSO [9] and HPSO [13] by achieving better fitness value in all three scenarios. In third scenario, the DIFOA achieves extremely better fitness value when the iteration size is greater than 500. The DIFOA obtained an average fitness value of 6.5, which is better than GA, PSO and HPSO where their average fitness values are (4.7, 4.9, and 5.3), respectively.

In order to evaluate the efficiency of the proposed DIFOA algorithm, the two scenarios are considered. In Fig.7, the proposed DIFOA achieves optimal execution time compared with the simple PSO, GA and HPSO. However, when the service scale is more than 60, the DIFOA achieves slightly better execution time than other algorithms in which the abstract services are same. The proposed motion equation helps in avoiding premature convergence, where the proposed method explores globally to attain better solution.

The DIFOA achieves execution time a slightly better when compared with other bench marks. The HPSO results are very close to the proposed DIFOA method. The proposed method is even performs good with increase in service scale and population.

In Fig.8, the execution time is compared by keeping the same number of iterations (1000) over different population size varies from 100 to 1000 with increase of 100. The DIFOA achieves better execution time when the population size is less than 800. However, it falls back a little when the population size is 800 and 900.

In order to know the stability of the proposed HPSO algorithm, we have used error function which

is the root mean square error (RMSE) is shown in Eq. (10) as follows

$$RMSE = \sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2 / n} \qquad (10)$$

Where $\bar{x}$ represents the user preferences of *'n'* QoS parameters, x_i represents the optimal QoS weights of i^{th} candidate service.

The customer satisfaction is measured by calculating root mean square error. The user preferences are compared with selected candidate services to infer the error values by which the user satisfaction is computed. The objective of service selection and composition is to minimize the error value.

From Fig.9, it is clear that the DIFOA algorithm outperforms the GA, PSO and HPSO in terms of achieving minimal error rate by varying the population size. Even HPSO performs achieves minimal error rate in most of the sceneries.

6. Conclusion

An improved optimization algorithm for QoS aware service composition is proposed using discrete immune optimization algorithm and fruit fly (FOA). The pareto optimal solution is determined to enhance the convergence speed of the proposed DIFOA method. The immune optimization performs global search process by which there is significant decrease in computation time. The FOA employs local search strategy to enhance the convergence speed with good fitness value. The proposed DIFOA is tested with 20 user preferences, the experimental results shows that the DIFOA performs better compared with simple GA, PSO and HPSO in terms of fitness value, execution time and error value. The DIFOA achieves fitness value of nearly 6.4 when the number of population and iteration value is 1000. The DIFOA obtains an error rate below 0.10 for most of the cases where other benchmarks are failed to minimize error rate.

Future work will follow three directions. First of all, since the workflow and concrete services were fairly similar in terms of range, the effect of larger variations in the request-workflow pairs will be investigated. Secondly, a multi-optimization PSO will be investigated and compared to the current approaches. Thirdly, different selection matching functions will be experimented with.

References

[1] Y. Qiang, L. Chen, and L. Bin, "Ant colony optimization applied to web service compositions in cloud computing", *Computers & Electrical Engineering*, Vol. 41, pp.18-27, 2015.

[2] J. Amin, E. Sundararajan, and Z. Othman, "Cloud computing service composition: A systematic literature review", *Expert Systems with Applications*, Vol. 41, No. 8, pp.3809-3824, 2014.

[3] M. Rajeswari, G. Sambasivam, N. Balaji, MS. Saleem Basha, T. Vengattaraman, and P. Dhavachelvan, "Appraisal and analysis on various web service composition approaches based on QoS factors", *Journal of King Saud University-Computer and Information Sciences*, Vol. 26, No. 1, pp.143-152, 2014.

[4] Z. Wei, J. Wen, M. Gao, and J. Liu, "A QoS preference-based algorithm for service composition in service-oriented network", *Optik-International Journal for Light and Electron Optics*, Vol.124, No. 20, pp.4439-4444, 2013.

[5] K. Heba, A. Al-Anazi, C. Campbell, and A. Al Faries, "A combinatorial optimization algorithm for multiple cloud service composition", *Computers & Electrical Engineering*, Vol.42, pp.107-113, 2015.

[6] Z. Guobing, Y. Chen, Y. Yang, R. Huang, and Y. Xu, "AI planning and combinatorial optimization for web service composition in cloud computing", In: *Proc. of international conference on cloud computing and virtualization*, pp. 1-8, 2010.

[7] L. Yuan-sheng, Y. Kun, T. Qiang, J. Zhang and B. Xiong, "A multi-criteria network-aware service composition algorithm in wireless environments", *Computer Communications*, Vol.35, pp.1882–1892, 2012.

[8] H. Serge, L. Mokdad, and S. Youcef, "Selection of the Best composite Web Service Based on Quality of Service", In: *Proc. of ISSS/BPSC*, Vol.10, pp.255-266, 2010.

[9] J. Liao, Y. Liu, X. Zhu, T. Xu and J. Wang, "Niching particle swarm optimization algorithm for service composition", In: *Proc. of Global Telecommu-nications Conference (GLOBECOM)*, pp. 1-6, 2011.

[10] S.B. Bhushan and P. Reddy, "A four level linear discriminant analysis based service selection in the cloud environment", *International Journal of Technology*, Vol.5, pp. 859-870, 2016.

[11] S. Liu, G. Xiong, H. Zhao, X. Dong and J. Yao, "Service composition execution optimization based on state transition matrix for cloud computing", In: *Proc. of Intelligent Control and Automation (WCICA)*, pp. 4126-4131, 2012.

[12] L. Zeng, B. Benatallah, A.H. Ngu, M. Dumas, J. Kalagnanam and H. Chang, "Qos-aware middleware for web services

composition", *IEEE Transactions on software engineering*, Vol.30, No.5, pp.311-327, 2004.
[13] M. Alrifai and T. Risse, "Combining global optimization with local selection for efficient QoS-aware service composition", In: *Proc. of inter-national conference on World wide web*, pp.881-890, 2009.
[14] I. Estévez-Ayres, P. Basanta-Val, M. García-Valls, J.A. Fisteus and L. Almeida, "QoS-aware real-time composition algorithms for service-based applications", *IEEE Transactions on Industrial Informatics*, Vol.5, No.3, pp. 278-288, 2009.
[15] S.C. Oh, D. Lee and S.R. Kumara, "Effective web service composition in diverse and large-scale service networks", *IEEE Transactions on Services Computing*, Vol.1, No.1, pp. 15-32, 2008.
[16] J.M. Ko, C.O. Kim and I.H. Kwon, "Quality-of-service oriented web service composition algorithm and planning architecture", *Journal of Systems and Software*, Vol.81, No.11, pp. 2079-2090, 2008.
[17] Y. Zhu, W. Li, J. Luo and X. Zheng, "A novel two-phase approach for QoS-aware service composition based on history records", In: *Proc. of Service-Oriented Computing and Applications (SOCA)*, pp.1–8, 2012.
[18] S.B. Bhushan and C.H. Reddy, "A Network QoS Aware Service Ranking Using Hybrid AHP-PROMETHEE Method in Multi Cloud Domain", *International Journal of Engineering Research in Africa*, Vol.24, pp. 153-164, 2016.
[19] Y. Qi and A. Bouguettaya, "Efficient service skyline computation for composite service selection", *IEEE Transactions on Knowledge and Data Engineering*, Vol.25, pp.776–789, 2013.
[20] S.A. Ludwig, "Clonal selection based genetic algorithm for workflow service selection", In: *Proc. of Evolutionary Computation (CEC)*, pp. 1–7, 2012.
[21] Y. Liu, M. Li and Q. Wang, "A novel user-preference-driven service selection strategy in cloud computing", *International Journal of Advancements in Computing Technology*, Vol.4,pp. 414–421, 2012.
[22] S.B. Bhushan and P. Reddy, "BB-LBA: biogeo graphy-based load balancing algorithm in multi cloud domain," *International Journal of Internet Protocol Technology*, Vol. 9, No.2-3, pp. 100-106, 2016.
[23] K. Nossal, J.V. Gustav, "Life, death and the immune system", *Scientific American*, Vol.269, No. 3, pp.53-53, 1993.
[24] A. Simoes and E. Costa, "An immune system-based genetic algorithm to deal with dynamic environments: Diversity and memory", *Artificial Neural Nets and Genetic Algorithms*, Vol.14, pp.168-174, 2003.