International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-5, June 2019

QoS-Driven Optimal Multi-Cloud Service Composition using Discrete and Fuzzy Integrated Cuckoo Search Algorithm

A V L N Sujith, a Rama Mohan Reddy, K Madhavi

Abstract*: *Recent generations of service computing evidence that quality of Service (QoS) driven optimal composition of multi-cloud services (QoS-CSC) is considered as a vital problem in the context of manufacturing complex cloud services based on the client requirements. Due to the proliferation of the seamless cloud services, optimal service composition of the multi-cloud services is considered as an NP-Hard problem. The QoS-CSC problems are solved by using continuous optimization algorithms to make them discrete a suitable encoding scheme is adopted with continuous optimization. In this paper, we propose a fuzzy integrated numerical encoding scheme in which the solution is represented as a fuzzy matrix and the composition of cloud services is obtained using that fuzzy matrix and further the global QoS attributes are computed. Discrete and Fuzzy integrated Cuckoo Search algorithm (DFCS) is developed by using our proposed fuzzy solution encoding schema. Further, during the process of performance evaluation, an empirical comparison various existing metaheuristic algorithms and proposed DFCS algorithm is illustrated using a set of real-world cloud services to identify exceptional service composition that optimizes local along with the global QoS attributes.

Index Terms*: *Service-oriented Computing (SoC), Quality of Service (QoS); Discrete optimization; Cloud Service Composition; Cuckoo search; Bat algorithm; Fuzzy encoding.

I. INTRODUCTION

Current generation researchers extensively recognized Cloud computing (CC) as the most influential virtualization based information technologies because of its outstanding benefits in terms of service provisioning. In general, CC enables the user with 3 categories of services that include software-as-a-service (SaaS) that enables the user to work with his desired software application over the internet based on the pay-as-u-go principle, infrastructure-as-a-service (IaaS) provides the high-end infrastructure through server virtualization that enables the user to work on virtualized infrastructure with enhanced memory and storage based on his requirements, and platform-as-a-service (PaaS) allows the user to develop his application in a virtualized development environment securely maintained at the data center using thin client devices.

Manuscript published on 30 June 2019.
* Correspondence Author (s)
A V L N Sujith, Research Scholar in CSE, JNTUA University, Anantapuramu, India.
Dr. A Rama Mohan Reddy, Professor in CSE, Sri Venkateswara University College of Engineering, Tirupati, India.
Dr. K Madhavi, Associate Professor in CSE, JNTUA University College of Engineering , Anantapuramu, India.

Utilization of these Cloud services (CS's) benefits the organizations the in terms of the economy such that, instead of investing on the computational infrastructure along with its maintenance, they could make use of the virtualized services provisioned by various cloud service providers based on the principles of utility computing. They gained further popularity with the emergence of CC that provides an easy-to-use and set up an environment for hosting applications virtually as service to the client based on the pay-as-you-go phenomenon [1]. However, these CS's as individual entities often cannot fulfill the client requirements in the context of accomplishing a complex task. To accomplish complex tasks on satisfying user requirements, multiple cloud services (CS) may need/require to be composed [2, 3]. The choice of selecting and utilization of the service depends on the quality of service (QoS) parameters. For composite CS's, there are disparate CS providers with similar functionalities, which have different QoS parameters. The QoS parameters determine the values of functional as well as non-functional attributes such as cost indicted to utilize the service, response time taken by the service, throughput, availability, and reliability of the desired service. These QoS parameters apply to both service provider and service composition with global QoS constraints. Cloud service composition (CSC) involves selecting a service from the pool of available cloud services for each subtask and combining them together to build a desired composite CS. The coherent expansion of CS makes service selection and CSC with diverse QoS parameters under multiple user constraints that lead to a multi-objective optimization problem which is considered to be in the class of NP-hard problem [3]. The primary objective of QoS-CSC is to select the appropriate CSC which optimizes the QoS parameters of the composite CS that fulfill the user's requirements [4]. Because of the discrete nature of the CSC issue, an enhanced numerical encoding schema is utilized to work with continuous optimization algorithms that help in handling QoS-CSC. Most of the existing models while solving QoS-CSC utilize nature-inspired continuous optimization techniques [6, 7, 8, 9, 5] that encodes the schema as a vector of integer numbers having a measurement equivalent to the number of tasks in which every section of the vector includes records related to the corresponding candidate CS that implements the task. The inherent drawbacks of encoding solution as a vector of integers involve performing dissimilar operations for updating the vector to reflect the new and better solution [9, 10].

We consider the influence of neighborhood of solutions for updating the subtask, whereas vector encoding scheme considers only the best solutions. Hence, the probability of being confined to local optima gets increased [11]. The influence on the other subtask solutions is calculated using fuzzy set. With this concept, more than one subtask solutions can affect others to some extent. Apart from this, the fuzzy encoding scheme considers shifting and mutation that enhances the convergence speed to overcome the trap within the local minima. This article is intended to provide a solution for QoS-CSC using a discrete and fuzzy integrated encoding mechanism that enhances Cuckoo Search (CS) [12] in terms of optimization. The proposed fuzzy encoding strategy is transformed as the discrete optimization problem from a continuous optimization problem and rapidly converges with the numerical encoding strategy [13] for continuous optimization methods. It is observed from the literature that, the discrete and fuzzy encoded numerical schema has not been integrated together with Cuckoo Search (CS) while solving QoS-CSC.

The salient contributions of this research study are: (i) A formal model describing QoS-aware CSC (ii) A proposal of discrete enhanced and fuzzy integrated cuckoo search (DFCS) (iii) A comparison of the vector of integers encoding scheme of PSO, CS, BA, and other algorithms with fuzzy encoding scheme of PSO, CS, and BA. The results illustrate the improved convergence and efficiency of the proposed fuzzy numerical encoding scheme. The rest of the article is organized as follows: preliminary research studies are discussed in Section 2. Section 3 discusses the modeling of CSC using rudimentary fuzzy theory. Section 4 discusses the proposed discrete and fuzzy integrated cuckoo search algorithm. Performance evaluation of the proposed algorithm is illustrated and demonstrated in Section 5 which is followed by remarks and conclusion in Section 6.

II. PRILIMINARIES

This segment of the article details the concepts of the standard Cuckoo search algorithm, together with the basics of the fuzzy encoding scheme. Further, this section illustrates the multi-objective metaheuristic techniques such as NSGA-II as well as E^3-MOGA. It is observed that these continuous optimization algorithms obtain near-optimal solutions for QoS-CSC using an encoding scheme that makes these algorithms discrete.

A. Fuzzy Theory

Fuzzy systems (FS) employ possibility theory in the process of dealing with uncertainty in their reasoning process [14]. A fuzzy set is an essential part of every FS. Based on the basic principles of mathematics, a set is considered to be an infinite, finite and countably infinite collection of elements. Every element may or may not be considered as a member of the set. But in the context of FS, each element could be considered as a partial member of the set. In every fuzzy set A, multiple pairs of finite and infinite elements are represented as $(y, \mu A(y))$, in which y represent the set and μA indicates the function related to the degree of membership which represents the quality and compatibility of the set. FS involves the process of fuzzification and defuzzification [15] in which it has the capability of transforming the crisp values to fuzzy and vice-versa. During the process of fuzzification, based on the degree of the membership function the crisp variables are assigned to the fuzzy set. Each fuzzy set is associated with a membership function which can be constructed employing any approach: exact, meta-heuristics well as a heuristic, that includes Gaussian, Sigmoid and triangular or any other function. The values of the membership function must be confined in between [0, 1] where 0 means that it is not a member and 1 means that it is fully a member of the fuzzy set and the membership function should be unique. At the time of defuzzification, the output of the fuzzy systems is allocated to the crisp values using defuzzification approaches [16].

B. Cuckoo Search (CS)

In general Cuckoo Search adopts the concept of the off-spring parasitism of few cuckoo species such as Ani and Guira cuckoos. Initially, the eggs of the female cuckoos are laid within the nests of the different birds. These have an extraordinary ability to pair their eggs based on some physical properties that include size color and texture of the other eggs in the nest. If the host bird identifies the eggs of the cuckoos they are either thrown out or the nest may be abandoned. Cuckoos reproductive strategy is the base for algorithm development [17].

The idealized system for CS algorithm is as follows:

1. For every iteration, one egg is laid by each cuckoo and these are randomly dumped in the selected nest.
2. The appropriate nest with the superlative quality of the eggs are forwarded for the next generation
3. In this context, there will be a fixed quantity of host nest and the probability of identifying the cuckoos egg by the host bird will be $P_a \in [0, 1]$ [14]. While producing the new solution for n^{th} cuckoo says $Y_n^{(t+1)}$, an L´evy flight is computed as follows:

$$Y_n^{(t+1)} = Y_n^{(t)} + \alpha \oplus levy(\lambda) \qquad (1)$$

In the above equation $\oplus$ denotes the entrywise multiplication in which $\alpha > 0$. The Levy distribution mechanism consists of Levy flight which is considered to be essential in random walk step size. The Levy distribution notated as the distribution of M identical and independent random variables that are distributed and their Fourier transformation is notated as shown below:

$$F_M(l) = \exp[-M|l|^{\beta}] \qquad (2)$$

The inverse provides us with the actual form of distribution that does not have the analytical forms despite few cases. Thereby we can express the inverse as an asymptotic series as well as its leading order approximation to determine the heavy-tailed flight length results in based on the power law distribution.

C. Non-dominated Sorting Genetic algorithm II (NSGA-II)

NSGA-II is the considered most popular genetic algorithm and one of the important multi-objective optimization technique. The method works based on the fast non-dominated sorting, fast crowded distance estimation approach as well as the simple crowded comparison operator [15].

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

The working procedure of NSGA-II is illustrated below:

Initial Population: Initial population is generated based on the range of the content and problems if any.

Non-dominated Sorting: Sort the generated population in the first phase using the Pareto front. Compute the fitness value of each individual and assign the ranking. Along with rank, the crowd distance for every individual is analyzed to evaluate the distance measure with its neighbors.

Selection: Crowd-comparison operator plays a vital role to choose the individual using binary tournament.

Genetic Operators: A coded Genetic algorithm is generated with the simulation of binary crossover along with the polynomial mutation.

Recombination and Selection: In recombination, and offspring and current generation populations are merged further to select the individual of the next generation population. Further, each front of the new generation is filled until the amount of the population exceeds the current generation population size.

D. E3- MOGA

E^3- MOGA is popular multi-objective genetic algorithm by [16] that provides equal quality solution sets, assessing the trade-off among different solutions. In this technique, the evaluation of the fitness value is computed depending on the density and domination ranking phenomenon. The individual density is utilized to analyze the count of individuals providing similar objective values as the other individuals whereas the domination ranking indicates how an individual outperforms the other. Furthermore, the domination rank persuades the fitness value whereas the density dissuades the fitness value. E^3- MOGA maintains an elite population that includes the individuals with higher fitness values and these are evolved from genes across the generations based on the genetic operations like crossover and mutation.

E. GDE3

Generalized Differential Evolution (GDE3) is an extended version of Differential Evolution (DE) algorithm intended to enable global optimization based on the uninformed number of constraints along with their objective functions [17]. Specific cases in which there is only single objective without any constraints then GDE3 functions the same as DE. GDE3 enables the multi-objective problems with appropriate distributed solution. Furthermore, the GDE3 modifies DE with L constraints along with N objective functions with the crowding distance that approximates the vector crowdedness within its non-dominated set [15]. It handles multiple numbers of N objectives and L constraints that includes L = 0 which is considered as an unconstrained problem and N = 0 which I considered as a Constraint Satisfaction Problem. GDE3 the process of the selection is made by the following rules [17]:

1. In the case of the infeasibility of both vectors in the context of the constraint violation space, the trailing vector is elected if and only if it dominates the old vector, else the old vector is elected.
2. The feasible vector is selected among feasible and invisible vectors
3. The context in which both the vectors are feasible, the old vector and the trailing vector are compared to analyze the domination levels based on the objective function space such that if the old vector slightly dominates the trailing vector then trail vector opts else the old vector is elected. If none of the vectors are dominated by each other the both of them are forwarded for next-generation processing.

III. MODELING QoS-DRIVEN SERVICE COMPOSITION IN CLOUD

Let a service composition in the cloud (CSC) is considered to be a fusion of m tasks CSC = $T_1, \ldots, T_m$, let the concrete services (CS) in a set $A_i = S_1, \ldots, S_{Xi}$ is the collection of the Xi CS that can address T_i, and let $a = A1, \ldots, an$. The candidate CS's within a subset Ai consists of identical functionalities however they differ in their QoS attributes. The binding of CS S_{ij} for performing T_i can be illustrated as a matrix Y having dimension $n \times m$.

$Y_{ij} = 1$ if S_{ij} is selected for task T_i ; Otherwise $Y_{ij} = 0$ (3)

Where $1 \leq j \leq m$ and $1 \leq i \leq n$.

Every task T_i could be implemented by only a single service S_{ij} which was picked from subset $\{S_{i1}, S_{i2}, \ldots, S_{im}\}$. Each service S_{ij} has 4 QoS parameters denote by V: Reliability (R), Throughput (T), Availability (A), as well as Response Time (RT).

QoS parameters can be categorized as positive parameters (V^+) and negative parameters (V^-). Positive parameters indicate higher values with a maximum rate of user utility. Negative parameters indicate higher values with a minimum rate of user utility [18, 19]. For example, reliability, availability, and throughput are positive parameters whereas response time and cost are the negative parameters. The QoS properties of CS are described by its QoS parameters. The list of QoS parameters (V^+, V^-) and their meaning is described in Table 1.

The normalized equation for QoS parameters is as follows:

$$\eta_{norm} = \begin{cases} \dfrac{v - v_{min}}{v_{max} - v_{min}} & \text{if } V_{max} \neq V_{min} \text{ for positive QoS Parameters } (V^+) \\ \dfrac{v_{max} - v}{v_{max} - v_{min}} & \text{if } V_{max} \neq V_{min} \text{ for negative QoS Parameters } (V^-) \\ 1 & \text{if } V_{max} = V_{min} \text{ for both} \end{cases} \quad (4)$$

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

Where V_{max} denotes the maximum QoS values for all the composition paths similarly V_{min} represents the minimum QoS values for every single composition path. If V_{min} and V_{max} are equivalent, then the normalized value is considered to be 1.

Table 1: Description of Various QoS Parameters

S.No	Parameter Name	Description
1	Reputation	Evaluation of certainty of the cloud service in terms of user experience
2	Reliability	The rate of probability in which the request is responded correctly within the maximum expected time
3	Availability	The rate of probability in which the cloud service is available to the client upon his during the time period of the request
4	Throughput	The number of invocations by a cloud service in a certain time period
5	Cost	The price that is charged against the utility of the cloud service
6	Response Time	The rate of time period taken by cloud service to accomplish the client request

The aggregate functions (Agg_q) are utilized to compute the global QoS parameters of the cloud service composition. The normalizing function η_{norm} determines the quantity of the QoS parameters that are most needed for the client [20, 21]. Thus, the primary purpose of multi-objective optimization is to identify a set of compromise solutions X that minimizes the global utility/ fitness function ($G(X)$), given as follows:

$$MinG(X) = \begin{cases} g_1(x), g_2(x), g_3 ..... , g_n(x) \\ x \in \omega \end{cases} \quad (5)$$

Therein, $g(n):\omega \leftarrow \mathbb{R}^n$ is vector that values function, n > 2 objectives, $\omega \epsilon \mathbb{R}^n$ is considered to be a feasible solution set. Global QoS parameters are computed by employing the aggregate functions of the QoS parameters towards the building blocks that design the composition structure [5]. For instance, if the process of CS is fulfilled in sequence, the global response time is determined by sum of all individual response times of cuckoo search. If CS's are accomplished in parallel, then the global response time is determined by the highest response time of the individual concrete service.

However the current studies concentrate on the sequential composition because of the easiness of converting the different modes into the sequential mode. The aggregation functions of various QoS parameters for different modes of composition are represented as follows:

$$Agg_{Pc} = \sum_{i=1}^{n} Pc(S_{ij}) \quad ; \quad Agg_{Tp} = \min_{i=1}^{n} Tp(S_{ij}) \quad ;$$

$$Agg_{Ab} = \pi_{i=1}^{n} Ab(S_{ij}) \quad ; \quad Agg_{Rb} = \pi_{i=1}^{n} Rb(S_{ij}) \quad ;$$

$$Agg_{Rt} = \sum_{i=1}^{n} Rt(S_{ij})$$

In the above context S_{ij} denotes the selected service from task T_i. The aggregation function of every quality of service parameter is illustrated as shown in Table 2, PR_i denotes the probability of branch I as well as k is considered as number of loops.

Table 2: QoS parameters along with their aggregation functions

QoS attributes	*Sequence*	*Parallel*	*Loop*	*Conditional*
Throughput(Tp)	$Agg_{Tp} = \min_{i=1}^{n} Tp(S_{ij})$	$Agg_{Tp} = \min_{i=1}^{n} Tp(S_{ij})$	$Agg_{Tp} = K * \pi_{i=1}^{n} Tp(S_{ij})$	$Agg_{Tp} = PR_i * \pi_{i=1}^{n} Tp(S_{ij})$
Price(Pc)	$Agg_{Pc} = \sum_{i=1}^{n} Pc(S_{ij})$	$Agg_{Pc} = \sum_{i=1}^{n} Pc(S_{ij})$	$Agg_{Pc} = K * \sum_{i=1}^{n} Pc(S_{ij})$	$Agg_{Pc} = PR_i * \sum_{i=1}^{n} Pc(S_{ij})$
Reliability(Rb)	$Agg_{Rb} = \pi_{i=1}^{n} Rb(S_{ij})$	$Agg_{Rb} = \max_{i=1}^{n} Rb(S_{ij})$	$Agg_{Rb} = K * \pi_{i=1}^{n} Rb(S_{ij})$	$Agg_{Rb} = PR_i * \pi_{i=1}^{n} Rb(S_{ij})$
Availability(Ab)	$Agg_{Ab} = \pi_{i=1}^{n} Ab(S_{ij})$	$Agg_{Ab} = \pi_{i=1}^{n} Ab(S_{ij})$	$Agg_{Ab} = K * \pi_{i=1}^{n} Ab(S_{ij})$	$Agg_{Ab} = PR_i * \pi_{i=1}^{n} Ab(S_{ij})$
Response time(Rt)	$Agg_{Rt} = \sum_{i=1}^{n} Rt(S_{ij})$	$Agg_{Rt} = \min_{i=1}^{n} Rt(S_{ij})$	$Agg_{Rt} = K * \sum_{i=1}^{n} Rt(S_{ij})$	$Agg_{Rt} = PR_i * \sum_{i=1}^{n} Rt(S_{ij})$

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

The main objective of the single objective optimization is to develop a scalarization based approach which can yield the convex optimization of the objective functions, i.e., the relation between two potential solutions. On the contrary, algorithms based on multi-objective optimization determine multiple solutions that approximate Pareto Front (PF) set, maintaining good solution diversity. Hence the solutions for multi-objective optimization are determined by PF set. Let the objectives be $min\ (g_1(x), g_2(x), g_3(x) \ldots\ldots\ g_n(x))$, $C = [1, 2, 3, \ldots\ldots, n]$ and $g_c(x) \in G(X)$. Further the PF set is described as follows:

The decision vector $M \in \mathbb{R}$ will dominate other vector $w \in \mathbb{R}$ (m<w) if and only if:

$\forall_c \in C : g_c(m) \leq g_c(w) \cap \exists_c \in C : g_c(m) < g_c(w)$

A solution $x \in \mathbb{R}$ is considered to be a Pareto optimal solution if :

$\forall x \in \propto, \forall_c \in C, g_c(x^*) = g_c(x)\ \exists_c \in C, g_c(x^*) < g_k(x).$

Where $\propto$ represents the feasible Pareto Optimal Solution Set (POS) that comprises of various Pareto optimal solutions and PF include solutions of POS along with the set of objective function values that contributes towards the solution.

$PF = g_1(x), g_2(x), g_3(x) \ldots\ldots\ g_n(x)\ /x \in POS.$

IV. PROPOSED METHOD

A. Discrete and Enhanced fuzzy integrated Cuckoo Search algorithm

Let Cloud service composition $CSC = \{T_1, T_2, \ldots\ T_K\}$, $K \in 1,2,3,\ldots\ldots,m$ be the collection of subtasks T_K in which each subtask is resolved by the available services $T_K = \{SC_{k1}, SC_{k2}, SC_{k3}, \ldots\ SC_{kn}\}$ then the fuzzy relation in between the tasks and available services are denoted as follows:

$$SC' = \begin{pmatrix} SC_{11} & SC_{12} & \ldots & SC_{1m} \\ SC_{21} & SC_{22} & \ldots & SC_{2m} \\ . & . & \ldots & . \\ . & . & \ldots & . \\ SC_{n1} & SC_{n2} & \ldots & SC_{nm} \end{pmatrix} \quad (6)$$

In the above equation indicates the degree of membership between k^{th} element within the task t_K and l^{th} element in SC_{kl} within the available services T_K to the relation SC'.

$$SC_{kl} = \delta R\ (T_K, SC_l),\ k \in 1, 2, 3, \ldots n \text{ and } l \in 1, 2, 3, \ldots m \quad (7)$$

Where δR is considered as a membership function, SC' indicates the degree of membership in which the task T_K is resolved by the service SC_l in a feasible solution. In this context, each service SC_l is considered as a fuzzy set, in which subtask t_k is assumed as a partial member by means of a quantity of membership value V_{kl} where $1 \leq k \leq m$ and $1 \leq l \leq n$. Therefore for every subtask t_o the location of each nest can be represented as:

$$Z' = \begin{pmatrix} V_{11} & V_{21} & \ldots & V_{m1} \\ V_{12} & V_{22} & \ldots & V_{m2} \\ . & . & \ldots & . \\ . & . & \ldots & . \\ V_{1n} & V_{2n} & \ldots & V_{mn} \end{pmatrix} \quad (7)$$

Consequently, the element within the matrix Z' has to satisfy the following conditions that include:

$V_{kl} \epsilon\ [0,1],\ k \epsilon\ 1,2,3,\ldots.n,\ l \epsilon 1,2,3,\ldots.n \quad (8)$

and $$\sum_{l=1}^{n} V_{kl} = 1, k \in 1,2,3,\ldots n, l \in 1,2,3,\ldots m \quad (9)$$

The solution of each nest is depicted as a fuzzy matrix by using a fuzzy encoding scheme. The position matrix value of V_{kl} may violate the constraints in equation 8 as well as equation 9 subsequent to the updated location (or some iteration). To satisfy the attained constraints, the following computations are performed:

$$\text{If } V_{kl} < 0, V_{kl} = 0 \quad (10)$$

In the context of obtaining the crisp values from the specified defuzzification process the max value of $[V_{k1}, V_{k2}, V_{k3\ldots\ldots}, V_k]$ for every subtask t_k is obtained and assign the subtask T_k to SC_{kl} if V_{k1} is maximum. Further, the requisite matrix Y_{ij} represented in equation 3 can be acquired on setting V_{ij} to the minimum value let us say '1' and all others with the value '0'. Based on equation 5 the global fitness/utility function is evaluated and further this process is applied to the Cuckoo search algorithm.

$$Z'_{normalize} = \begin{pmatrix} \frac{V_{11}}{\sum_{l=1}^{n} V_{1j}} & \frac{V_{21}}{\sum_{l=1}^{n} V_{2j}} & \ldots & \frac{V_{m1}}{\sum_{l=1}^{n} V_{mj}} \\ \frac{V_{12}}{\sum_{l=1}^{n} V_{1j}} & \frac{V_{22}}{\sum_{l=1}^{n} V_{2j}} & \ldots & \frac{V_{m2}}{\sum_{l=1}^{n} V_{mj}} \\ . & . & \ldots & . \\ . & . & \ldots & . \\ \frac{V_{1n}}{\sum_{l=1}^{n} V_{1j}} & \frac{V_{2n}}{\sum_{l=1}^{n} V_{2j}} & \ldots & \frac{V_{mn}}{\sum_{l=1}^{n} V_{1j}} \end{pmatrix} \quad (11)$$

In Enhanced Fuzzy Cuckoo Search algorithm (DFCS), every nest (solution) utilized in Cuckoo search algorithm is described as a fuzzy matrix (Z') as depicted in equation 7. In the first phase, a population of n host nests is initialized randomly using equation 7 and normalized ($Z'_{normalize}$) by using equation 11. In every iteration, the best solution is obtained among the initial solutions.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

The new set of solutions (nests) is derived in every iteration using the random walk and Levy distribution by reserving the current best solution. Further few solutions are substituted and initialized with new nests/solutions. The binding matrix (Y_{ij}) as in equation 3 is computed by defuzzification, and the global utility/ fitness function (G(X)) is evaluated using equation 5. Finally, based on the ranking of nests/ solutions, the current best solution is discovered. DFCS is described in Algorithm 1.

Algorithm : Enhanced Discrete and Fuzzy Integrated Cuckoo Search Algorithm

Input: No. of cloud Services, Host nest size, PR_i in [0,1].

Output: Optimal best service composition.

for each task ($T_{K,}$ $\forall_k \epsilon$ *0,1,2,3…..,n*) **do**

for each Service $SC_{l,}$ $\forall_k \epsilon$ *0,1,2,3…..,n* do

Initial population of host nest $Z^{'}$ is randomly generated using equation 7;

Normalize $Z^{'}_{normalize}$ using equation 11;

while task t < Maximum-generations or Stop criterion **do**

A new random solution is randomly generated by Cuckoo using Levy weight in equation 1;

Normalize the generated solution based on equation 11;

Compute the binding matrix ($Z^{'}$) using equation 7;

Evaluate the fitness function ($G(X_k)$) of the service using equation 5;

Randomly identify the nest (say, b) among n nests;

If $G_{X(K)} > G_{x(b)}$ then

replace the solution *b* with current best solution *k*

End

The portion of worst nests/solutions is discarded;

Further new nests/solutions are generated/build and normalized using equation 7;

Maintain the best quality nests/solutions;

Optimize and rank the solutions based on the current best solution

End

End

End

V. PERFORMANCE EVALUATION

In this paper, we proposed DFCS algorithms to handle QoS-aware CSC. The paper applies the concepts of a fuzzy encoding scheme for QoS-aware CSC on a dataset of real-world services [21]. The main objective of this section is to evaluate the performance of fuzzy integrated proposed mechanism for CSC and compute the recital of the vectors in numerical encoding scheme of Cuckoo Search (CS), Bat Algorithm (BA), Particle swarm optimization (PSO), Generalized Differential Evolution (GDE3),

*Non-dominated Sorting Genetic Algorithm II (*NSGA-II) and *E*3 – *MOGA* against our proposed fuzzy encoding scheme Discrete and fuzzy integrated Cuckoo Search algorithm (DFCS).

A. Experimental setup

All algorithms are executed and evaluated in Matlab R2013a. The QWS Dataset v2 [21] is utilized, and the algorithms are assessed on a computer system that configured with a processor of Intel Core i5 with 2.60GHz and 12 GB of memory. The parameters tuned are as follows: the population size was 50 for all algorithms, *pa* of the DFCS and CS was 0.25, and α was set to 1. The *fmin* and *fmax* were set to 0 and 1. For DFPSO and PSO with ω = 0.72984, φ1 = 1.4962, and φ2 = 1.4962 were used. For NSGA-II, crossover probability and mutation probability values set to 0.9 and 0.1. The distribution indexes for both crossover and mutation operators were 20. Simulated polynomial mutation and binary crossover were applied (based on [40, 48, 49]). For GDE3 algorithm, we set *CR* = 0.80 and *F* = 0.30 (obtained from [22, 23]).

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

For *E*3 − *MOGA*, mutation probability is 0.03 and crossover probability is 0.5. All these values are adopted empirically after several experiments. In order to check the distribution of the QWS data set, we performed a Shapiro-Wilk test. The results disclosed normal distribution for the given data set. Table 3 exhibits the list of algorithms that are compared against our proposed mechanism. The count of iterations is set to 1000 for all experiments.

Table 3. Algorithms evaluated with the proposed approach

Algorithm Name	Abbreviations	References
Bat Algorithm	BA	[24]
Fuzzy particle swarm optimization	FPSO	[10]
Cuckoo Search	CS	[12]
Generalized Differential evolution	GDE	[25]
Evolutionary Multi-objective genetic algorithm	E^3-MOGA	[26]
Non-Dominated Sorting Genetic Algorithm-II	NSGA-II	[27]

As visual observations of the Pareto Front (PF) are not sufficient to decide the effectiveness of the approaches and to compare the approaches exactly, an objective evaluation is required. We have assessed the performance of our proposed algorithm using Spreading-based comparisons.

5.2 Performance results

We have considered the following QoS parameters in our experiments: price, throughput, availability, reliability, and response time. All the parameters are taken into consideration equal importance, and hence the weights were set to 0.20. Since all the methods are non-deterministic, each mechanism is performed with 25 independent runs. In Figure 1, we have illustrated the balanced solution obtained by our proposed algorithms and other compared algorithms (PSO, BA, CS, MOGA, GDE3, and NSGA-II) with the objective functions $f1$ and $f2$ (where $f1$ represents the positive QoS parameter (availability) and $f2$ represents the negative QoS parameter (response time). Figures 1 and 2 represent Availability and response time of the 2-objective optimization problem and Availability, response time and throughput of the 3-objective optimization problem respectively. The problem scenario comprises of 25 abstract services in which every abstract service is inculcated with 100 candidate services. As observed in Figure 1, each algorithm produced slightly different and not identical solutions due to the stochastic nature of the approaches and their characteristics. From Figure 1, we observed that most of the solutions obtained by our proposed approaches are in PF. A good solution should have a balance between positive and negative QoS parameters, i.e., the solutions with high availability, high throughput, high reliability, low price, and low response time. For instance, PSO, BA, and CS provide numerous optimal solutions with high availability and high throughput. Similarly, MOGA, GDE3, and NSGA-II produce optimal solutions with high availability and high throughput. But some of the solutions are dominated by NSGA-II. DFCS, DFPSO, and DFBA produce several good solutions with high availability, high throughput, low response time, and low price.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)

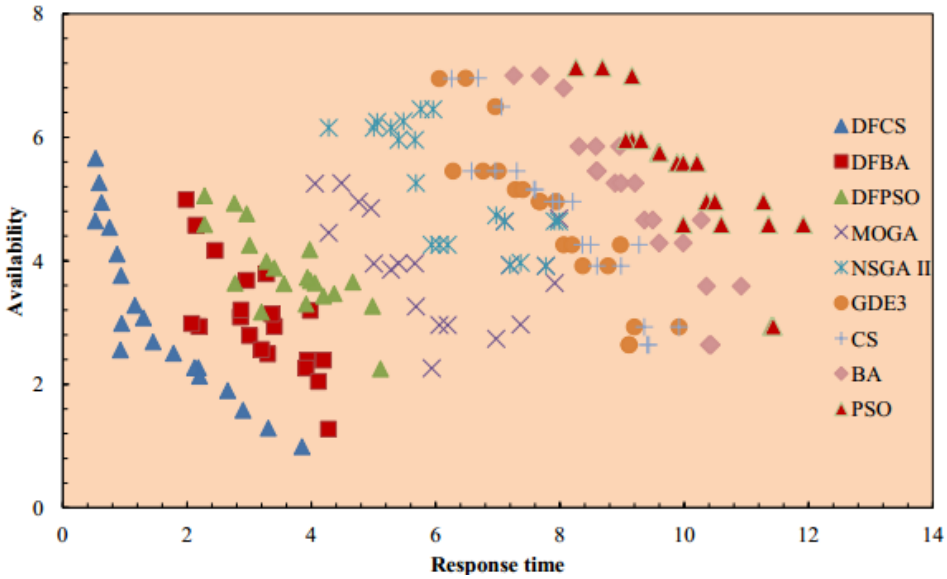

Figure 1: Solutions obtained by DFCS, DFBA, DFPSO, MOGA, NSGA-II, GDE3, CS, BA, and PSO for 2- objective optimization

From Figure 2, we observed that MOGA, GDE3, and NSGA-II yield optimal solutions with high availability, low response time, and high throughput. But some of the

solutions are dominated by NSGA-II and GDE3 algorithms. PSO, BA, and CS provide numerous optimal solutions with high availability, high throughput and low response time. GDE3, DFPSO, DFBA, DFCS algorithms are the best candidates for solving the QoS optimization problem.

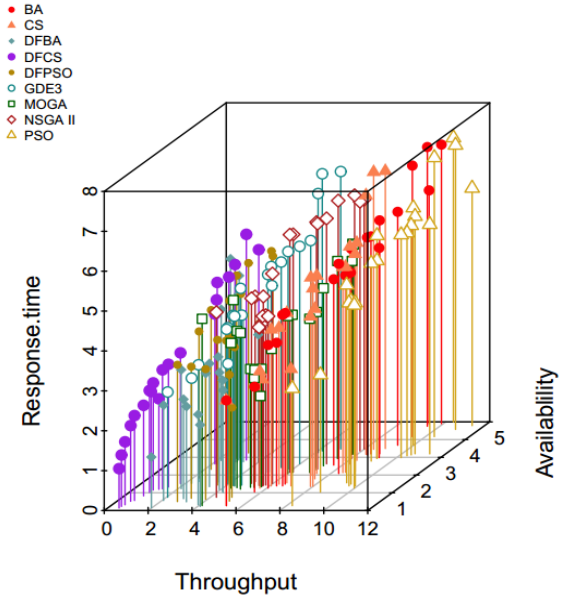

Figure 2: Solutions obtained by DFCS, DFBA, DFPSO, MOGA, NSGA-II, GDE3, CS, BA, and PSO. 3-objective optimization

Figure 3 indicates the evaluation of the fitness value of various algorithms when compared with the proposed approach. It is observed that the fitness value of DFCS outperforms various existing algorithms for 1000 iterations remarkably and consistently in around 4.8 fitness ratio.

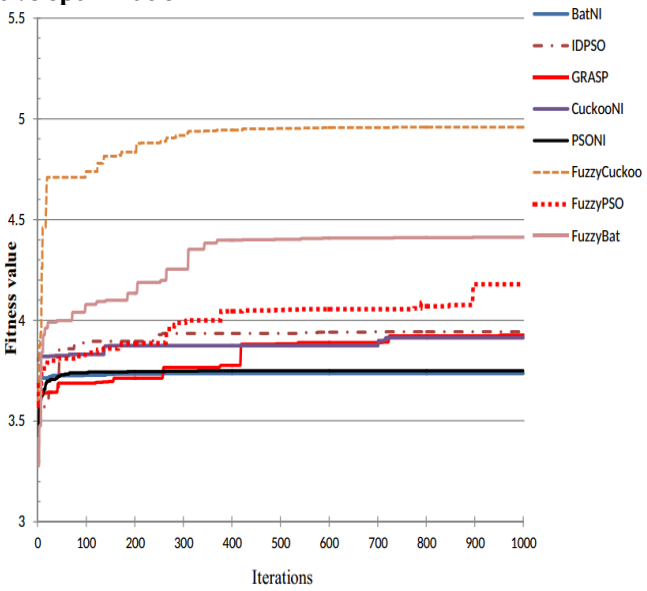

Figure 3: Evaluation of the fitness value

VI. CONCLUSION AND FUTURE WORK

In this research study, an optimal and QoS-aware CSC solved by CS using fuzzy integrated numerical encoding scheme. The fuzzy integrated numerical encoding scheme was used because continuous algorithms will not be able to derive solutions for discrete problems like QoS-aware CSC. DFCS algorithm is developed and implemented using the proposed fuzzy numerical encoding scheme. This proposed algorithm help in the context of selecting the superlative composition from a large set of real-world services viz. service repository that optimizes the composition of QoS parameters to manufacture a composite service on fulfilling the user's requirements. Based simulation studies, it is inferred that DFCS algorithm outperformed when compared with several existing algorithms consistent with the Wilcoxon signed-rank significant test which is carried out with a rate of 1% level of significance.

As a part of the future study, we plan to work on finding more effective optimization algorithms (Against Spider Monkey Optimization, Binary Grey Wolf Optimizer) and various numerical solution encoding mechanism that caters to the composition of large scale big data influenced cloud services. Further, we plan to propose a new robust algorithm to handle the unavailable services in a dynamic environment..

REFERENCES

1. R. Buyya, J. Broberg, A. M. Goscinski, Cloud computing: principles and paradigms, John Wiley & Sons, 2011.
2. H. Wang, X. Wang, X. Zhang, Q. Yu, X. Hu, Effective service composition using multi-agent reinforcement learning, Knowledge-Based Systems 92 (2016) 151–168
3. D. Ardagna, B. Pernici, Adaptive service composition inflexible processes, IEEE Transactions on Software Engineering 33(6) (2007) 369–384.
4. G. Canfora, M. Di Penta, R. Esposito, M. L. Villani, An approach for QoS-aware service composition based on genetic algorithms, in: Proceedings of the 7th annual conference on Genetic and evolutionary computation, ACM, 2005, pp. 1069–1075
5. F. Seghir, A. Khababa, A hybrid approach using genetic and fruit fly optimization algorithms for qos-aware cloud service composition, Journal of Intelligent Manufacturing 29 (8) (2018) 1773–1792.
6. A. Jula, Z. Othman, E. Sundararajan, Imperialist competitive algorithm with proclus classifier for service time optimization in cloud computing service composition, Expert Systems with applications 42 (1) (2015) 135–145.
7. Z.-Z. Liu, D.-H. Chu, C. Song, X. Xue, B.-Y. Lu, Social learning optimization (slo) algorithm paradigm and its application in qos-aware cloud service composition, Information Sciences 326 (2016) 315–333.
8. H. Jin, X. Yao, Y. Chen, Correlation-aware qos modeling and manufacturing cloud service composition, Journal of Intelligent Manufacturing 28 (8) (2017) 1947–1960.
9. W. Pang, K.-p. Wang, C.-g. Zhou, L.-j. Dong, Fuzzy discrete particle swarm optimization for solving traveling salesman problem, in: Proceedings of the 4th International Conference on Computer and Information Technology, 2004, pp. 796–800.
10. H. Liu, A. Abraham, A. E. Hassanien, Scheduling jobs on computational grids using a fuzzy particle swarm optimization algorithm, Future Generation Computer Systems 26 (8) (2010) 1336–1343.
11. P. J. Angeline, Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences, in: Proceedings of the International Conference on Evolutionary Programming, Springer, 1998, pp. 601–610. 51
12. X.-S. Yang, S. Deb, Cuckoo search via l´evy flights, in: Proceedings of the World Congress on Nature & Biologically Inspired Computing (NaBIC), 2009, pp. 210–214.
13. J. Liao, Y. Liu, X. Zhu, J. Wang, Accurate sub-swarms particle swarm optimization algorithm for service composition, Journal of Systems and Software 90 (2014) 191–203.
14. L. A. Zadeh, Outline of a new approach to the analysis of complex systems and decision processes, IEEE Transactions on Systems, Man and Cybernetics 3 (1) (1973) 28–44.
15. L. A. Zadeh, Fuzzy sets, Information and control 8 (3) (1965) 338–353.
16. W. V. Leekwijck, E. E. Kerre, Defuzzification: criteria and classification, Fuzzy Sets and Systems 108 (2) (1999) 159 – 178.
17. T. Yu, Y. Zhang, K.-J. Lin, Efficient algorithms for web services selection with end-to-end qos constraints, ACM Transactions on the Web (TWEB) 1 (1) (2007) 1–26
18. A. Parejo, S. Segura, P. Fernandez, A. Ruiz-Cortes, Qos-aware web services composition using grasp with path relinking, Expert Systems with Applications 41 (9) (2014) 4211–4223.
19. M. Liu, M. Wang, W. Shen, N. Luo, J. Yan, A quality of service (qos)-aware execution plan selection approach for a service composition process, Future Generation Computer Systems 28 (7) (2012) 1080 – 1089.
20. M. Cremene, M. Suciu, D. Pallez, D. Dumitrescu, Comparative analysis of multi-objective evolutionary algorithms for qos-aware web service composition, Applied Soft Computing 39 (2016) 124–139.
21. E. Al-Masri, Q. H. Mahmoud, Investigating web services on the world wide web, in: Proceedings of the 17th International conference on World Wide Web, ACM, 2008, pp. 795–804.
22. D. Zaharie, Influence of crossover on the behavior of differential evolution algorithms, Applied Soft Computing 9 (3) (2009) 1126–1138
23. S. M. Islam, S. Das, S. Ghosh, S. Roy, P. N. Suganthan, An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 42 (2) (2012) 482–500
24. Y. Yu, H. Ma, M. Zhang, An adaptive genetic programming approach toqos-aware web services composition, in: Proccedings of the IEEE Congress on Evolutionary Computation, 2013, pp. 1740–1747
25. S. Kukkonen, J. Lampinen, Gde3: The third evolution step of generalized differential evolution, in: Proceedings of the IEEE Congress on Evolutionary Computation, IEEE, 2005, pp. 443–450
26. H. Wada, P. Champrasert, J. Suzuki, K. Oba, Multiobjective optimization of sla-aware service composition, in: Proceedings of the IEEE Congress on Services-Part I, IEEE, 2008, pp. 368–375
27. Y. Yao, H. Chen, Qos-aware service composition using nsga-ii1, in: Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human, ACM, 2009, pp. 358–363

AUTHORS PROFILE

Mr. A V L N Sujith, Research Scholar in department of CSE JNTUA University Anantapur and Assistant Professor in SVCE, Tirupati, A.P., India. He received his B Tech and M Tech in Computer Science and Engineering, JNTUA University, Anantapuramu. Currently he is pursuing Ph.D in JNTUA university in department of CSE. His research intrests include Cloud Computing, IoT and FoG Computing.

Dr. A. Rama Mohan Reddy obtained his Bachelor Degree in Mechanical Engineering and Master's degree in Computer Science Engineering from NIT Warangal and PhD degree from Sri Venkateswara University and at present working as a Professor in Department of Computer Science and Engineering, Sri Venkateswara University College of Engineering. His areas of interest include Software Architecture and data mining. He has more than 30 years of experience in teaching and research.

Dr. K. Madhavi, working as Assistant professor in the Department of computer science and Engineering, JNTUA college of Engineering, Anantapur, Andhra Pradesh, India. She received **B.Tech** in Electronics and Communication Engineering from JNT University, Anantapur, India, **and M. Tech** in Computer Science from JNT University, Anantapur, India, and **Ph.D** from JNT University, Anantapur, India in August, 2012. Her research interests include computer networks, Cloud Computing, Mobile communications and wireless systems. She has published several papers in national and international refereed journals and conferences

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication (BEIESP)