

An Efficient QoS-aware Services Selection in IoT using A Reputation Improved- Social Spider Optimization Algorithm

Aghabi Nabil Abosaif (✉ aghape.nabeel@gmail.com)

Sudan University of Science and Technology <https://orcid.org/0000-0002-7082-1125>

Dr. Salah Elfaki Elrofai

Sudan University of Science and Technology

Research

Keywords: Services Selection Algorithms (SSA), Quality of Services (QoS), Internet of Things (IoT), Likert Scale Measurement, Social Spider Optimization (SSO) Algorithm, Reputation Improved-Social Spider Algorithm (RI-SSO)

Posted Date: September 15th, 2020

DOI: <https://doi.org/10.21203/rs.3.rs-38596/v2>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

An Efficient QoS-aware Services Selection in IoT using A Reputation Improved- Social Spider Optimization Algorithm

Aghabi N. Abu-Safe

College of Computer Science,
Sudan University of Science and Technology,
Sudan

Associate Prof. Dr. Salah Elfaki Elrofai

School of Electronic Engineering, Sudan
University of Science and Technology,
Sudan

ABSTRACT

Internet of Things (IoT) has been grown rapidly over the last years to connect a considerable number of spatially distributed objects or actuators. The connected objects create new functionality and provide various services to enhance and satisfy End-users daily lives. The issue is to provide the End-users with optimal services based on their requirements. The critical challenge is to select the optimal service from similar services functionally and various services non-functionality requirements (Quality of services). To achieve this challenge, this paper proposed a services selection model under QoS constraints in the IoT environment. The introduced model implements a meta-heuristic optimization algorithm with a friendly Likert scale measurement method. It aims to improve the performance of bio-inspired optimizing algorithms, called a Social Spider Optimization (SSO) Algorithm, by adding a reputation value to member's weight. The proposed model used a Likert scale measurement to evaluate the reputation of the services from the End-users. In the experiments, a comparative study was done between an original SSO and the proposed RI-SSO model. The results show the efficiency of the proposed RI-SSO model against the original SSO, in both maximization and minimization problems. It obtains a better outperform in terms of fitness values.

Keywords:

[Services Selection Algorithms (SSA), Quality of Services (QoS), Internet of Things (IoT), Likert Scale Measurement, Social Spider Optimization (SSO) Algorithm, Reputation Improved-Social Spider Algorithm (RI-SSO)]

1. INTRODUCTION

Internet of Things (IoT) is a coherent environment, which aims to link physical things together. It senses things by using a massive number of sensors and actuators. These actuators connected to create new applications that facilitate End-users life. The provided applications composite from any number of abstract services connected as one service for End-user to perform more complex functions[1]. In the IoT environment, the number of sensors continuously increasing, this leads to an increase in the number of services. The provided services from the same sensors type; are similar in their

functional properties, but they differ in their non-functional properties.

The service selection algorithm is fundamental in the IoT environment. It is an NP-hard problem that aims to select an optimal abstract service from a set of concrete services. The selection criteria build based on the Quality of Service (QoS) constraint that is defined by the End-users. ITU-T E.860 defined QoS [2] as "The degree of conformance of the service delivered to a user by a provider with an agreement between them." In the IoT environment, QoS formed in several architectures. Authors in[3] proposed QoS architecture to be appropriate)with IoT architecture. They divided QoS into three main layers (Sensor layer, Network layer, and Application layer). These layers integrated the traditional QoS attributes with other essential qualities in IoT Architecture (e.g., network deployment cost, Energy efficiency management, information accuracy). This paper focused on QoS from the Application Layer. It represents the highest layer in IoT architecture, which consists of many distributed services composed to be one service or application. This layer aims to deal with ranking services based on users' requirements of QoS. Examples for QoS in the application layer as (Performance time, Execution Time, Availability, Services Perform Price, and Reliability)[3].

To this end, many solutions have been developed to address the QoS aware of service selection schemes. The proposed solution aimed to solve a selection problem by using a bio-inspired Meta-heuristic Optimization Algorithm (M-HOP). M-HOP algorithms are one of the widely used to address the services selection problem. A Meta-heuristic is artificial intelligence algorithms introduced by Glover in 1986. It comes from the combination of two Greek terms meta-(metá, which means the high-level), and heuristic (euriskein, which means "I find, discover").

One of the recent M-HOP used to solve the optimization problem is a Social Spider Optimization Algorithm (SSO). The proposed solution improved the behavior of SSO by enhancing the member weights in the colony. The member weights increase or decrease based on the service reputation that takes from the End-user. The reputation value also called historical information[4] or End-user feedback information[5]. This information gathered from End-user after using a service. The proposed model introduced an easily friendly method to gather this historical information by using a Likert Scale measurement. It called a Reputation Improved-Social Spider Algorithm (RI-SSO) because it improved the behavior of SSO based on gathered Reputation information.

The rest of this paper organized as follows: First, the next section introduced the related works.

Section three and four show the services in the IoT environment and services selection process, respectively. In section five, a QoS composition model discussed. Composed services selection model and fitness function defined in section six and section seven, respectively. The Likert scale measurement explained in section eight. Section nine illustrated an original social spider optimization algorithm. The proposed model of the Reputation Improved-Social Spider Optimization (RI-SSO) algorithm explained in section ten. Section eleven shows the experiment settings and dataset. Section twelve explored the conclusion. Finally, the recommendation and future work discussed in section thirteen.

2. RELATED WORK

Select the optimal services that match the End-user requirement is a core study for many researchers over the years. There are many studies done by applying bio-inspired optimizing algorithms. In particular, the meta-heuristic algorithms proposed to enhance the QoS constraints aware a services selection process in IoT environment. Authors in [6],[7],and [8]

applying a Meta-heuristic algorithm by considering the objective QoS information which is supported by the service providers. *Li et al.* [7] introduced an optimization solution in the cloud logistics platform in IoT and Cloud computing environments. They termed the services selection problem as a constraint satisfaction problem (CSP). The proposed a dynamic model depend on Practical Swarm Optimization (PSO) algorithm. They considered four QoS constraints (availability, reliability, time, and cost). *Liu et al.* [8] proposed a Cooperative Evolution algorithm (CE) by integrating two meta-heuristics algorithms (Genetic Algorithm (GA) and Canonical PSO (CPSO) algorithm). They computed four QoS factors in their algorithm (cost, time, availability, and reliability). In their approach, authors regarded some characteristics for compositions implemented on a large scale like: ensures the diversity of the population, convergence in the global best solution, optimize a local best solution, and fit the Self-adaptive mechanism of learning rates. *M. Elhosenya et al.* [6] proposed a new model to improve health services applications (HAS) in an industry 4.0 and cloud-IoT based environment. The model aimed to optimize virtual machine selection (VMs). It executed using three different Meta-heuristic optimization algorithms (Parallel Particle swarm optimization (PPSO), Particle swarm optimizer (PSO), and Genetic Algorithm (GA)). Authors regarded five factors: reduce the performance time, adjust the required storage of patient's data, optimize the resource utilization, improve the scheduling of medical requests tasks, and provide a real-time data retrieval mechanism for HAS.

While there are some researchers[5], [4] and [10] emphasize the subjective QoS feedback information that comes from End-user after using selected services. Such as *Mejri et al.* [5]adopted a self-adaptive approach to ensure scalability in the IoT environment. The

introduced approach consists of two models: *QoS prediction model*: used an Artificial Neural Network (ANN) to predict QoS by considering three contexts (user, service, and network). A *technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) model*: to present the optimal service to the consumer of the services. They optimized two QoS factors: execution time and reliability. *Nwe et al. in*[4], presented a new model called a Flexible QoS-Based Service Selection Algorithm (FQSA) to match, rank, and select the distributed things on dynamic IoT environments. The selection factors divided into two main categories: *Objective Information category* comes from the service providers, and the *Subjective Information category* comes from the service consumers. The model allowed the End-user to request the QoS criteria in an easy and friendly manner using a flexible, user-friendly assessment form. It divided into two parts based on two previous factors: *First part to calculate the user subjective factors*, authors implemented a Similarity Aggregation Method (SAM)) to evaluate the creditability of the different End-users. Also, to understand QoS characteristics and extract End-users input meaning, they used QoS ontology, WordNet, and ontological reasoning. *In the second part of finding the objective factors*, the proposed FQSA algorithm implemented Artificial Neural Network Back-propagation Algorithm (ANN-BP) to improve a selection performance rate to be acceptable in real-time service selection.

Also, few researchers introduced the optimization solutions for the selection problem using a social spider algorithm (SSO). *Mousa et al. [9]* proposed SSO as a solution for services selection process-aware QoS constraints. The proposed model introduced in the general web services environment by considering the objective QoS information only. The authors satisfied in the three QoS factors (execution time, availability, and throughput). Their experimental results show that SSO has less execution time and better global searching than PSO.

In [10], *Divyad et al.* proposed an idea for ranking the services in the registry based on End-user evaluation or feedback to enhance a selection and discovery processes. They proposed the SSO algorithm for the first optimizing processes. Then they proposed to use the gathered information to rank services in the registry for further optimized. They calculated the fitness value by considering response time, availability, and cost as the metrics in the selection process. They proposed their idea only as steps without any computation and without any method to explain how feedback information will gather from End-User. Also, they did not show any experimental setting or any experimental results for their model.

Therefore, in this article the proposed optimization model can differentiate from the above state-of-the-art models in two main parts: Firstly, the model proposed an improved SSO algorithm for IoT applications called Reputation Improved-SSO (RI-SSO). Secondly, it offered an easy, friendly method; most of the typical End-user manipulates with it before. It proposed to use the Likert-type management to calculate subjective QoS information (the retrieval evaluation feedback information comes from End-user).

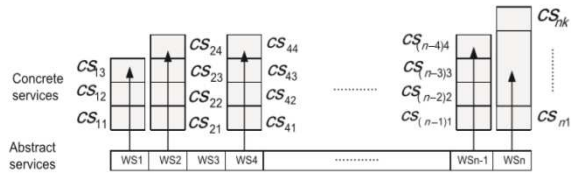
3. SERVICES SELECTION IN IoT ENVIRONMENT

In the IoT environment, there are numerous benefits and facilities provides to the End-users. These facilities provided as applications like health care, smart home, smart cities, animal tracking, manufacturing, and many critical applications. The applications build from many abstract services. These services compose together and deliver to the End-users to satisfy their requirements.

For each abstract service, there are many concrete services, which are similar in their functional properties but different in non-function properties.

The proposed model applies the *Horizontal adaptation* to optimize the selection problem.

The goal in the *Horizontal adaptation* is to find the ideal concrete service from a set of functionality equivalent candidate services for each abstract service separately [13] [7], as in Fig (1). It regards more appropriate for IoT environment, which contains the enormous number of sensors that provide services with the equivalent in functional properties and different in non-functional properties. Horizontal adaptation provides greater flexibility for user intervention[13], which enables the user to implement and modify the abstract workflow when required.



Fig(1): Services selection workflow[7]

4. QoS COMPOSITION MODEL

The selection of an optimal service that meets End-user requirements depends on QoS factors values that distinguished from one service to another. This approach identifies five QoS factors in the application layer. These factors selected from two quality types [14].

The first type is the Business Quality Type (BQT): it means an economic value that is offered by applying services. This value is using to evaluate the right service based on business value.

This paper studied two BQT factors, they are:

Reputation $q_{RP}(s)$: This is a social evaluation of service depending on the rates coming from different users after requesting services.

Execution Price $q_{EP}(s)$: It is a price value that the user pays for the service invocation to a provider during or after using the service.

The second type is the System Quality Type (SQT): it means QoS related to the system, based on process time, determines QoS

properties. Three SQT factors considered in this paper are:

Reliability $q_{RE}(s)$: It is the probability ratio to complete the services successfully.

Availability $q_{AV}(s)$: It refers to a probability ratio that the service is running and can be accessed when invoked.

Response Time $q_{RT}(s)$: It indicates the delay between the service request and the time. The service response was received. It can be measures by seconds.

To optimize the QoS value, the behavior of factors are varied from one factor to another[9]. Some factors optimized by getting the minimum values; they called Negative QoS factors. Other factors optimized by getting the maximum values; they called Positive QoS factors.

The optimal results for the Negative QoS factors (Execution Price and Response Time) are the smallest values. It evaluated as in equation (1).

$$\bar{Q}_i(CS) = \begin{cases} \frac{Q_i^{max} - Q_i(CS)}{Q_i^{max} - Q_i^{min}} & Q_i^{max} - Q_i^{min} \neq 0 \\ 1 & Q_i^{max} - Q_i^{min} = 0 \end{cases}$$

Moreover, the optimal results for Positive QoS factors (Reputation, Reliability, and Availability) are the highest values. They evaluated as in equation (2).

$$\bar{Q}_i(CS) = \begin{cases} \frac{Q_i(CS) - Q_i^{min}}{Q_i^{max} - Q_i^{min}} & Q_i^{max} - Q_i^{min} \neq 0 \\ 1 & Q_i^{max} - Q_i^{min} = 0 \end{cases}$$

Where i , ($1 < i < 5$) indicates the number of QoS factors. CS indicates concrete services. Q_i^{max} and Q_i^{min} represent the maximum and minimum values of the i -th QoS factor, respectively. QoS vector of concrete service CS is defined as follows:

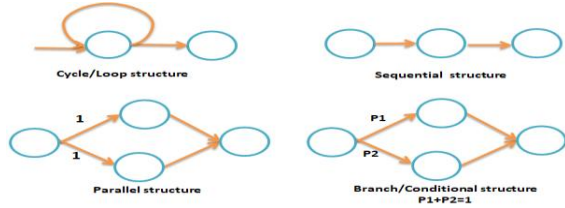
$$Q(CS) = (Q_{RP}(CS), Q_{EP}(CS), Q_{RE}(CS), Q_{AV}(CS), Q_{RT}(CS)) \quad (3)$$

5. COMPOSED SERVICES SELECTION MODEL

To compose services together using general web services technologies, there are four main composition structures (sequential, parallel, cycle, and branch)[8] [15], as shown in Fig(2). *Sequential*, is to execute the composition of services in sequential order one follows to others. *Parallel*, tasks are performed simultaneously by moving to the next task until all of these parallel tasks achieve. *Cycle*, at least one task must performed more than one time.

Branch, only one task from a set of optional tasks will be selected then moves to the next step. In the proposed model, the sequential workflow of services composition will be considered.

In a sequential workflow, the QoS value for each concrete service calculated by aggregating the corresponding values of each factor. The sequence-structure applies two types of QoS aggregation functions, as in Table (1). The additive function applies for response time $q_{RT}(s)$, execution price $q_{EP}(s)$, and reputation $q_{RP}(s)$ factors. The multiplicative function applies for availability $q_{AV}(s)$ and reliability $q_{RE}(s)$ factors.



Fig(2): Basic patterns of web service composition[16][8]

6. FITNESS FUNCTION

To get the optimal selection process, the objective function or fitness value calculated by considering the five QoS factors for candidate concrete services in all abstract services. The objective function or fitness value is calculated based on an optimization type to maximize or minimize the services selection as in equation (4):

Objective Function =

$$[\text{Min}(W_{RT} * F_{RT}(CS_{ij})) + \text{Min}(W_{EP} * F_{EP}(CS_{ij})) + \text{Max}(W_{AV} * F_{AV}(CS_{ij})) + \text{Max}(W_{RE} * F_{RE}(CS_{ij})) + \text{Max}(W_{RP} * F_{RP}(CS_{ij}))] \quad (4)$$

Where, $F_{QoS}(CS_{ij})$ represents the summation or multiplicative function for each factor. For simplicity's sake, our model calculates the summation for all QoS factors.

And W_{QoS} represents the weight for each factor, the

$$\sum_{i=1}^5 W_{QoS} = 1, \quad 0 < W_{QoS} < 1.$$

$i, (1 < i < 5)$ represents the number of QoS factors. In our model, we consider the End-users interested in services with high response time and execution price, so we give them a high weight equal to 0.25 for both of them. The reputation weight equals 0.20, while availability and reliability get less interested, so we give them low weights equal 0.15 for both of them as in equation (5), (6).

$$W_{QoS} = W_{RT} + W_{EP} + W_{AV} + W_{RE} + W_{RP} = 1 \quad (5)$$

$$W_{QoS} = 0.25 + 0.25 + 0.15 + 0.15 + 0.20 = 1 \quad (6)$$

Table (1): Aggregation Function To Compute The QoS Factors[17]

| Structure | Response Time | Execution Price | Availability | Reliability | Reputation |
|-----------|--|--|---|---|--|
| Sequence | $q_{RT}(\text{Seq}) = \sum_{i=1}^n q_{RT}(CS_i)$ | $q_{EP}(\text{Seq}) = \sum_{i=1}^n q_{EP}(CS_i)$ | $q_{AV}(\text{Seq}) = \prod_{l=1}^N q_{AV}(CS_i)$ | $q_{RE}(\text{Seq}) = \prod_{l=1}^N q_{RE}(CS_i)$ | $q_{RP}(\text{Seq}) = \frac{1}{n} \sum_{i=1}^n q_{RP}(CS_i)$ |

7. LIKERT-TYPE

Likert-Type defined in [18] as "a psychometric response scale primarily used in questionnaires to obtain participant's preferences or degree of agreement with a statement or set of statements." The scale measurement named by Dr. Rensis Likert in 1932. His goal was to improve a means of measuring psychological attitudes directly in a "scientific" method. There are many structures introduced to measure the levels of granularity. Our proposed solution implemented the most commonly used structure is a 5-point scale or levels[19]. The scales starting ranging from "Strongly Disagree", "Disagree", "Neither", "Agree", and "Strongly Agree" as in Fig (3). Each scale assigned to coding like using numeric value or alphabet value. This value used to measure the attitude under investigation, usually starting at one and incremented by one for each level.

The introduced paper uses a Likert-type to measure the agreement level of service taken from End-users after using it. The agreement level provides the feedback value about the End-user preference saved which as reputation value to each service. In the beginning, the reputation value set to equal zero. To analyze the collected feedback data, the median value is not the mean value of each service calculated to get a more precise answer. The median used because our solution used the Likert-type does not use a Likert scale [19].

For example if the service s_i , evaluated ten times from End-users as follow: 3,1,3,2,4,3,4,2,1,1. To calculate the median value, first, we need to reorder the evaluation list in ascending order as

1,1,2,2,3,3,3,3,4,4. Then find the position of middle one by divide the (list_length / 2): (10/2=5th position). The result rounded to the nearest integer number for odd list length. So the median is equal to 3, which indicates "Neither". The reputation value saved as a decimal number as in Table (2) —this the reputation value for service s_i , is equal to 0.6.

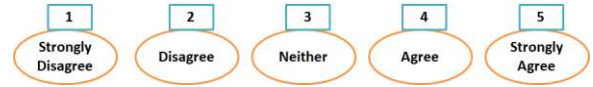


Fig (3): Five-Point scales used in Likert-type Measurement [19]

Table (2): Reputation values of 5-points Agreement levels

| Agreement levels | Equal number | Reputation Values |
|------------------------|--------------|-------------------|
| Not evaluated services | 0 | 0 |
| Strongly Disagree | 1 | 0.2 |
| Disagree | 2 | 0.4 |
| Neither | 3 | 0.6 |
| Agree | 4 | 0.8 |
| Strongly Agree | 5 | 1.0 |

8. An ORIGINAL SOCIAL SPIDER OPTIMIZATION ALGORITHM

SSO is a swarm intelligence algorithm emulates the collective behavior of spider swarms. It proposed by Cuevas et al. in [20] to find an optimal solution of complex optimization problems in continuous search space. In this artical the original SSO and proposed RI-SSO

used to solve the services selection problem in discrete search space, namely the Nearest Integer method[9].

One of the primary behavior in the SSO colony is mating behavior, females used the vibrations of the male spiders over the web to determine a heavier spider fitness.

SSO colony consists of two main components:

Communal Web: represent a search space of the optimization problem of the SSO algorithm where all spiders have a position on it. Each position represents an available solution to the optimization problem on the web. When a spider leaves the web, its position represents an unavailable solution to the optimization problem[21].

Social Members: They are spiders on the web which they an agent of SSO to perform optimization. It represents the complete population (pop_s). Social members are divided into two members group[20][21] as in Fig (4).

Females group: F represent 65% - 90% of the total colony members, $F = \{f_1, f_2, f_3, \dots, f_{N_f}\}$.

Males group: M represent 35% - 10% of the total colony members, $M = \{m_1, m_2, m_3, \dots, m_{N_m}\}$. Whereas $pop_s = F \cup M = F \cup M$, $S = \{s_1, s_2, s_3, \dots, s_N\}$, so $pop_s = \{s_1=f_1, s_2=f_2, \dots, s_{N_f}=f_{N_f}, s_{N_f+1}=m_1, s_{N_f+2}=m_2, \dots, s_N=m_{N_m}\}$. On the basis of gender, each individual is calculated through a set of different evolutionary operators that emulate different cooperative behaviors. Also spider receives a weight according to the fitness value of the solution on web.

8.1 Fitness Evaluation

On the web, each spider s_i has a weight SW_i , which represents the solution quality. The weight value of each solution s_i in population pop_s calculated by equation (7):

$$SW_i = \frac{f(s_i) - Worst_s}{best_s - Worst_s} \quad (7)$$

Where $f(s_i)$ is the fitness value obtained by the calculation of the spider position s_i concerning the fitness or objective function in equation (4), the bests and worst values are the maximum and

the minimum values of the solutions in the population.

8.2 Social Members (Spiders)

Each spider s on the web has a memory to store two types of information: *Individual situation information* used to describe the spider s , which consist of: The *position* of spider s on the population pop_s . The *fitness value* in the current position of spider s .

New positions information used to guide a spider s to *new positions*, and consist of: The *target vibration* of spider s in the previous iteration.

The *number of iterations* since spider s has last changed its target vibration. The *movement* that spider s performed in the previous iteration. The *dimension mask* that spider s employed to guide movement in the previous iteration.

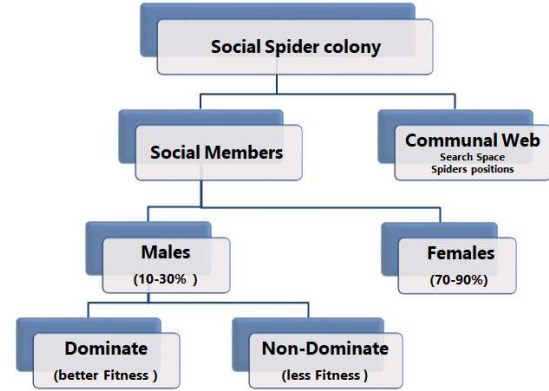


Fig (4): Social Spider Member

8.3 Vibrations through the Communal Web

The communal web is used as a mediator to transmit information among the spiders. This information is encoded as small vibrations V for the collective coordination of all individuals in the web. Vibrations strength are affected by two properties; the *intensity source*, and the *intensity attenuation*. The *intensity source* of the vibration also called the *weight factor*: it calculated by fitness function as in equation (8) in the range $[0, +\infty]$ [21]. whereas members have the highest weight generating stronger vibrations compared to members having the lowest weight. For each time t a spider s at position P_a move to a new position, it generates a vibration at its current position. This *intensity source* at the position P_a

is affected by the fitness value of its position $f(P_a)$, or spider weight at position P_a .

$$I_S(P_a, P_b, t) = \log\left(\frac{1}{f(P_a) - C} + 1\right)$$

(8)

Where, C is a confidently small constant. In minimization problems, all possible fitness values are more significant than the C value.

The spider positions with better fitness values, i.e., the larger value from maximization problems or smaller value for minimization problems, have higher vibration intensities source than those with worse fitness values.

The *intensity attenuation also called source position* I_d over distance $D(P_a, P_b)$ between spiders at position P_a and P_b calculated as in equation (9) with consideration of the physical energy phenomenon of vibration attenuation during the propagation process over the web.

$$I_d(P_a, P_b, t) = \exp\left(-\frac{D(P_a, P_b)}{\bar{\sigma} * r_a}\right)$$

(9)

Where, $\bar{\sigma}$ is a standard deviation of all spider positions.

So the *vibration intensity receives* $I(P_a, P_b, t)$ value sensed by a spider in position P_b and generated by a spider in source position P_a at time t calculated by equation (10) and (11) by regarding both intensity source I_S and intensity attenuation I_d .

$$I(P_a, P_b, t) = I_S(P_a, P_b, t) \times I_d(P_a, P_b, t)$$

(10)

$$I(P_a, P_b, t) = \log\left(\frac{1}{f(P_a) - C} + 1\right) \times \exp\left(-\frac{D(P_a, P_b)}{\bar{\sigma} * r_a}\right)$$

(11)

8.4 SSO Algorithm levels

To obtain an optimization selection solution by using SSO Algorithm, there are three levels :

Initialization level: is a start step in optimization processes done by initializing the following:

- The optimization search space which represents the hyper-dimensional spider web.

- The spider's positions (feasible solution for services selection) represent the population pop_s over the web. They generated randomly with their fitness, which represents the quality of each offered solutions.
- The objective function will use to select an optimal solution based on End-user preference.
- The End-user defined a value for the QoS factors to be used in SSA, representing female attraction.

Iteration level: SSO performs searching iteratively until finding the optimal solution between offered solutions. For each iteration; spiders on the population pop_s move to a new position and perform the following steps:

- Evaluate the fitness values of each spider S in the population pop_s .
- Generate the vibrations V for all spiders by using equation (8).
- Propagates these vibrations intensity over the web using equation (11).
- Select the strongest vibration value V^{best} from V , in maximization problems strongest vibration means the largest value and vice versa in minimization problems.
- Compare the intensity value of V^{best} with the sorted intensity value of the target vibration V^{tar} . If V^{tar} is less than V^{best} , the inactive degree IN^d reset to zero. Otherwise, V^{tar} value retained, and IN^d is incremented by one.
- Move a random walk towards V^{tar} by using dimension mask M to direct the spider movement. M is a binary vector $\in [0, 1]$ of length equal to the web dimension D of the optimization problem. Its value is changed based on the probability $1 - P_C^{IN^d}$. If the M value is changed, all vector elements have a probability of P_m to be an equal one, and a probability of $1 - P_m$ is zero.

Whereas, P_C is a user pre-defined parameter that detects the probability of changing the mask. Moreover, P_m is also a user pre-defined controlled parameter $\in [0,1]$.

- Generate a new following position $\mathbf{P}_{s,i}^{follow}$ based on the mask for \mathbf{S} As in equation (12):

$$P_{s,i}^{follow} = \begin{cases} P_{s,i}^{tar}, & M_{s,i} = 0 \\ P_{s,i}^r, & M_{s,i} = 1 \end{cases} \quad (12)$$

Where, $P_{s,i}^{tar}$ is the i th element of the source solution of V^{tar} , $P_{s,i}^r$ is a random solution's position, r is a random integer value $\in [1, |\text{pop}_s|]$, and $M_{s,i}$ is the i th dimension mask M of the spider S .

- Calculate the random walk of a new position $P_s(t+1)$ using the following formula (13):

$$P_s(t+1) = P_s(t) + (P_s(t) - P_s(t-1) \times r + P_s^{fo} - P_s) \odot R \quad (13)$$

Where, \odot operator indicates the element-wise multiplication operator, and $P_s(t)$ is a current position.

Final level: This level handles any constraint that can happen during iteration level lead to violating the optimization problem, such as spiders can move out of the web (maximum and minimum bounds) during the random walk step, which means the offered solution will be unavailable.

$$P_{s,i}(t+1) = \begin{cases} (\bar{X}_i - P_{s,i}(t)) \times r, & P_{s,i}(t+1) > \bar{X}_i \\ (P_{s,i}(t) - \underline{X}_i) \times r, & P_{s,i}(t+1) < \underline{X}_i \end{cases} \quad (14)$$

Where, \bar{X}_i is the upper bound on the search space, and \underline{X}_i is the lower bound.

9. PROPOSED MODEL

The proposed model build based on two natural behaviors in the social spider colony. These behaviors and proposed solution are arranged in the following two points

The first point, the natural behavior of mating between social spider members, done between a

females and dominant males (males with better fitness)[21]. As a result of a mating a new offspring generated with new fitness values. The generated fitness values based on the strength of the dominated male who performs a mating. Fitted male mating generates fitted offspring and vice versa[22]. To represent this behavior in the optimization selection problem, the females represent the End-Users, and Dominated males represent the candidate services.

The proposed solution aims to enhance and emulate this natural behavior. It regards a subjective factor by evaluating the feedback value take from the End-user to each service. Then add this evaluation value to the selected service as a reputation factor. The reputation value gathered by using an easy, friendly method called a Likert-type measurement, as in section 8. Then generate a new fitness value based on collected information on services reputation. The new fitness value of the service represents the new offspring with new fitness value. This added value leads to improve the next service selection process based on the collected evaluation from a previous selection process.

The second point, the behavior of attraction or dislike between females and males, has been evolved based on the propagation of the vibrations across the web from males to females. In original SSO, the strength of vibrations intensity sensed based on two properties[22] weight and distance as physical energy phenomena as in equations(8), (9), and (11). The more energetic vibration generated either by large spiders or neighboring members on the web.

Generally, in physics, the intensity defined as "the quantity of energy the wave conveys per unit time across a surface of unit area" [23]. The intensity formula is:

$$I = \frac{P}{A} \quad (15)$$

Where P is the power, and A is the area.

From the above equation, the relation between intensity and power is directly proportional. So the increase in the power value leads to an

increase in the intensity value. In a social spider colony, the power represents by members' weight. So the proposed approach focuses on the effect of members weights based on the collected evaluation information. If the End-users satisfy with selected services, they evaluate it with a high agreement point, for example (Strongly Agree, or Agree) and vice versa. This evaluation converts to reputation value to be (1.0, 0.8), respectively. The reputation value added to the member's weights or its fitness value as in equation (16). The service with a high reputation value meets the spider with high intensity that enhances the attraction of dominated males and vice versa. Our optimization model summarizes in the following steps as in Fig (6) where blue processes represent the Original SSO, and the red processes are the addition improvement:

Step 1: In the initial steps, the selection process implements by the Original SSO as in section 9.4, and the reputation value set to equal zero.

Step 2: The selected services deliver to the End-user, who evaluates it based on QoS constraints. The Likert scale measurement is used based on 5-points scales to get the evaluation value from the End-user.

Step 3: The reputation value calculated from the evaluation information as in Table (2). The value is one from the following (0.2, 0.4, 0.6, 0.8, and 1.0). The reputation value added to each service as a reputation property.

Step 4: Calculate the *intensity source* value by update the original SSO equation (16) to calculate as following:

$$I_s(P_a, P_b, t) = \log\left(\frac{1}{(f(P_a) + (f(P_a) * Rep_val)) - C} + 1\right) \quad (16)$$

Where, $f(P_a)$ is a fitness value of spider at position a , C is a confidently small constant.

Step 5: Evaluate the *vibration intensity* value by update the original SSO equation (17) to calculate as following:

$$(P_a, P_b, t) = \log\left(\frac{1}{(f(P_a) + (f(P_a) * Rep_val)) - C} + 1\right) \times \exp\left(-\frac{D(P_a, P_b)}{\bar{\sigma} * r_a}\right) \quad (17)$$

Then calculate the best *vibration intensity* receive in population and save it.

Step 6: Add the *best vibration intensity* receive value to the *target vibration* position as in this equation:

$$\text{Target vibration} = ((\text{Target vibration} * \text{Target Matrix}) + (\text{Max Position} * 1 - \text{Target Matrix})) + \text{Best Intensity Recive} \quad (18)$$

Note, in original SSO, the *target vibration* calculated without adding the *best vibration intensity* receives.

The introduced solution RI-SSO obtains good results compare with the original SSO for both optimization problems: maximization and minimization problems, as in section 11. It leads to useful and accurate response generation.

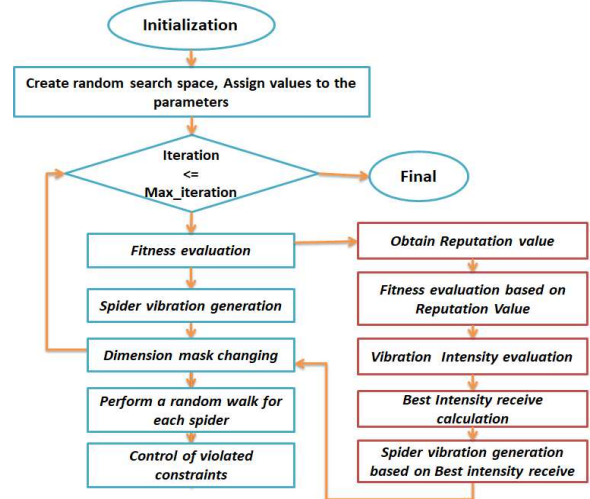


Fig (6): Flow Chart Of Original SSO and RI-SSO operations.[24]

10. EXPERIMENT SETTINGS AND DATASET

To evaluate the performance of proposed approach RI-SSO, it compared with the original SSO proposed by (Mousa and Bentahar 2016)[10] to solve the selection problem. The dimension of selection problem represented by the number of items in an integer array-coding

schemes. The indexes of candidate service are represented by the array elements.

The parameter's value using SSO and RI-SSO are[21]:

- $r_a = 1$ represents the parameter that controls the attenuation rate of the vibration intensity over the distance.
- $p_c = 0.7$, represent the user-defined attribute that describes the probability of changing mask.
- $p_m = 0.1$, represent the user-controlled parameter defined between (0, 1).

In the experiments, the number of stopping criteria (iterations) set to 50 and the population size to optimize the fitness value is sets to 20 for five abstract services. Each type of abstract service has a group of concrete services, as in Fig (1), and Fig (7). These services have similar functional properties but differ in non-function properties (QoS factors). The introduced model, regards five QoS factors (Execution Time, Availability, Cost, Reliability, and Reputation) for each concrete service. The dataset values for first four factors (Execution Time, Availability, Cost, and Reliability) take from(Li 2013)[7], see Fig (7), and the values of reputation factor generated randomly based on five Likert-type values(0.2, 0.4, 0.6, 0.8, 1.0).

The experiments tests implemented on a laptop with Windows 10, 2.90 GHz processor, and 8-GB RAM. The algorithms coded in MATLAB R2017b.

| | | | | |
|--|------|------|------|------|
| <p>CS= Concrete Service AS= Abstract Service</p> | | | | |
| CS13 | CS23 | CS33 | CS43 | CS53 |
| CS12 | CS22 | CS32 | CS42 | CS52 |
| CS11 | CS21 | CS31 | CS41 | CS51 |
| AS1 | AS2 | AS3 | AS4 | AS5 |

Fig (7): Data Set of Services selection workflow [7]

10.1 Performance Comparison

To analyze the performance of approach RI-SSO, the behavior of Original SSO compared with RI-SSO in two behaviors, one when the reputation value equals 0.2 the first or minimum point in Likert-type measurement. Other when the reputation value equals 0.8, the fourth or previous maximum point in Likert-type measurement.

The comparison is made based on three evaluations: firstly, it recorded the best intensity receive values obtain from the three behaviors for minimization and maximization problems. Results show that the increases in reputation value leads to an increase in the best intensity received values in maximization problem as in Fig (8.a). Furthermore, lead to a decrease in the best intensity receives values in minimization problems, as in Fig (8.b).

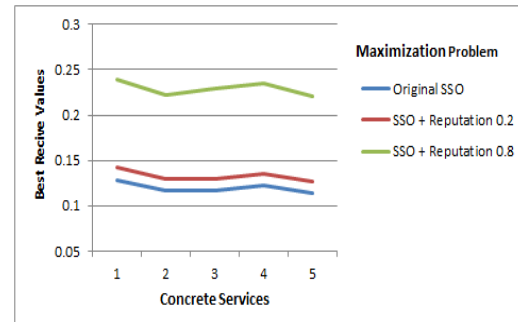


Fig (8.a) Maximization Problem

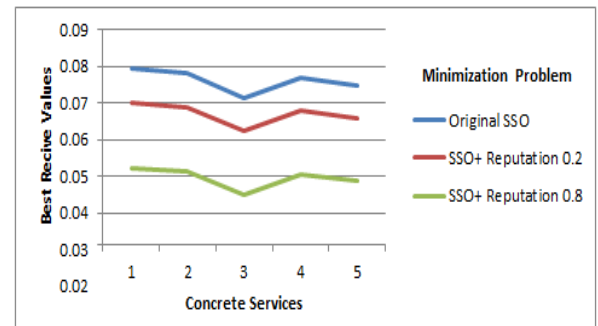


Fig (8.b) Minimization Problem

Fig (8): Evolution curves of Best intensity Receive values

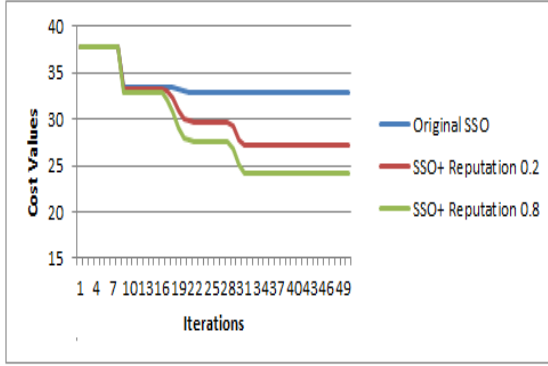


Fig (9.a) Evaluation of Cost Fitness value

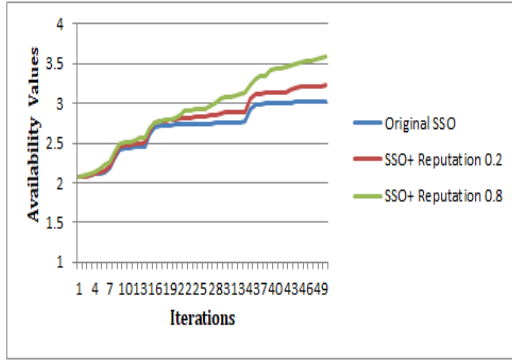


Fig (9.b) Evaluation of Availability Fitness value

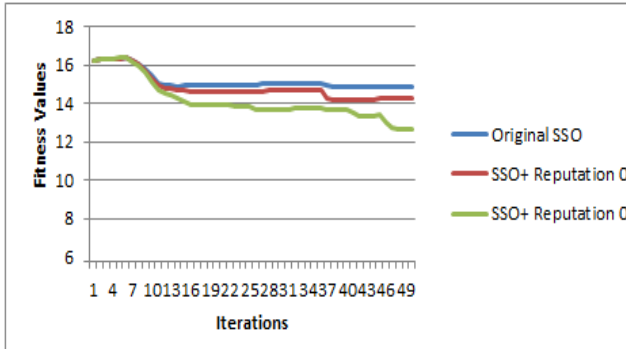


Fig (10): Comparing Results of Fitness Function Values

Secondly, it observed the evolution of fitness value over the two optimization types: For a maximization problem, it observed the fitness value of the cost factor. Results show in Fig (9.a) the RI-SSO with reputation 0.8 gets the best performance by obtaining the less fitness value than RI-SSO with reputation 0.2 and Original SSO.

For the minimization problem, it observed the fitness value of the availability factor. Results

show in Fig (9.b) the RI-SSO with reputation 0.8 gets the best performance by obtaining the highest fitness values than reputation 0.2 and Original SSO.

Thirdly, to approve the feasibility and efficiency of RI-SSO, it calculated the fitness values to optimize the five QoS factors by using equation (4). The result shown in Fig (10) demonstrates that SSO with a high reputation value can achieve better fitness values in this selection problem than SSO with low reputation value and original SSO. Thus, the RI-SSO can be used to optimize a selection problem by getting high availability and reliability while keeping a low cost and execution time.

11. CONCLUSION

To facilitate End-users life, IoT environments content a massive number of actuators that provide different services for them. The similar actuators can provide services with similar functionality but different in non-functional requirement. So the different End-users request the same service with different QoS criteria. Thus make a process of selecting the required service with the optimal QoS is an NP-hard problem. This paper proposed a meta-heuristic algorithm to solve the services selection problem. The proposed approach classified into two parts: *First part* is collecting feedback information about services from End-users who used it—this information collected through a user-friendly tool named Likert-type measurement. The median value of feedback saves as reputation value for each service. *The second part* is to find an optimal services selection solution by using a Reputation Improved-Social Spider Optimization (RI-SSO) algorithm, which regards reputation information. The model uses the collected information to improve the behavior of Original SSO. It adds value to new offspring in the SSO model by updating a fitness weight based on an average calculation of reputation information. The RI-

SSO upgrades the reliability of the searching process by selecting a service that appropriates with the End-user preference.

The comparison studies find that the proposed approach RI-SSO has an extreme fitness function value than Original SSO. The simulation results approve the efficiency of using the Likert-type with SSO in services selection is much higher than using SSO only.

12. RECOMMENDATION AND FUTURE WORK

Future work, aims to increase the dataset scale and implement a real-world case study for IoT application. Also recommends using a prediction tool that predicts the quantities and qualitative QoS which End-user can require based on evaluation information collected from Likert-type measurement.

13. LIST OF ABBREVIATIONS

| | |
|--------|--|
| BQT | Business Quality Type |
| CPSO | Canonical PSO Algorithm |
| FQSA | Flexible QoS-Based Service Selection Algorithm |
| GA | Genetic Algorithm |
| HAS | Health Services Applications |
| I_d | Intensity Attenuation |
| I_s | Intensity Source |
| IoT | Internet of Things |
| M-HOP | Meta-heuristic Optimization Algorithm |
| PPSO | Parallel Particle swarm optimization |
| PSO | Practical Swarm Optimization |
| QoS | Quality of Service |
| RI-SSO | Reputation Improved-Social Spider Algorithm |
| SSO | Social Spider Optimization |
| SQT | System Quality Type |
| VMs | Virtual Machine Selection |

14. DECLARATIONS

Availability of data and materials

The data and materials were available online from 2019 to submitted date.

Competing interests

The authors declare that they have no known competing financial interests or personal

relationships that could have appeared to influence the work reported in this paper.

Funding

There is no funding supporter for this research.

Authors' contributions

The two main contributions of this article are

- Collect the End-user feedback information using a friendly method Likert scale measurement in Services selection Algorithms and keep it as Reputation value for each service.
- Improve the performance of Social Spider Optimization Algorithm by using the collected Reputation value and add it to the weight of each spider in the web.

Acknowledgements

Authors thank our University (Sudan University of Science and Technology) who provided insight and expertise that assisted the research.

Authors' information



Aghabi Nabil Abu-Safe was born in Sudan in 1987. He received the B.E. and M.E. degrees from University of Khartoum (UoFK), Sudan, in 2008, and 2013, respectively.

She study the B.E. in the Information Technology Department and done a M.E. in Software Engineering.

Now she wants to get PhD in Computer science department from Sudan University of science and Technology.

She worked as Research Assistant a programmer in Nile Center for Technology Research (NCTR) in Sudan for one year. Then she worked as teaching assistant in UoFK for two years. Now she works as lecturer in University of science and technology in Sudan in Software Engineering department. Her main areas of research

interest are services Selection Algorithms in IoT environment, smart Bed in hospitals.

15. REFERENCES

- [1] P. Asghari, A. M. Rahmani, H. Haj, and S. Javadi, "Service Composition approaches in IoT: A Systematic Review," *J. Netw. Comput. Appl.*, 2018.
- [2] ITU-T, "Recommendation E.860: Framework of a Service Level Agreement," *Ser. E Overall Netw. Oper. Teleph. Service, Serv. Oper. Hum. Factors*, 2002.
- [3] L. Li, S. Li, and S. Zhao, "QoS-Aware scheduling of services-oriented internet of things," *IEEE Trans. Ind. Informatics*, vol. 10, no. 2, pp. 1497–1507, 2014.
- [4] N. W. E. Nwe, H. Win, B. A. O. Jianmin, and C. U. I. Gang, "Flexible user-centric service selection algorithm for Internet of Things services," *J. China Univ. Posts Telecommun.*, vol. 21, no. July, pp. 64–70, 2014.
- [5] M. Mejri and N. Ben Azzouna, "Scalable and Self-Adaptive Service Selection Method for the Internet of Things," *Int. J. of Computer Appl.*, vol. 167, no. 10, pp. 43–49, 2017.
- [6] M. Elhoseny, A. Abdelaziz, A. S. Salama, A. M. Riad, K. Muhammad, and A. K. Sangaiah, "A Hybrid Model of Internet of Things and Cloud Computing to Manage Big Data in Health Services Applications," *Futur. Gener. Comput. Syst.*, 2018.
- [7] W. Li, Y. Zhong, X. Wang, and Y. Cao, "Resource virtualization and service selection in cloud logistics," *J. Netw. Comput. Appl.*, pp. 1–9, 2013.
- [8] J. Liu *et al.*, "A Cooperative Evolution for QoS-driven IoT Service Composition," *Autom. – J. Control. Meas. Electron. Comput. Commun.*, vol. 54, no. 4, pp. 438–447, 2013.
- [9] V. Priyadarshini, P. Divya, D. Preethi, N. Pazhaniraja, and P. V. Paul, "A novel Web service publishing model based on social spider optimization technique," in *4th IEEE Sponsored International Conference on Computation of Power, Energy, Information and Communication, ICCPEIC 2015*, 2015, pp. 373–387.
- [10] A. Mousa and J. Bentahar, "An Efficient QoS-aware Web Services Selection using Social Spider Algorithm," *Procedia - Procedia Comput. Sci.*, vol. 94, no. MobiSPC, pp. 176–182, 2016.
- [11] M. Gigli and S. Koo, "Internet of Things: Services and Applications Categorization," *Adv. Internet Things*, vol. 01, no. 02, pp. 27–31, 2011.
- [12] M. Burhan, R. A. Rehman, B. Khan, and B. S. Kim, "IoT elements, layered architectures and security issues: A comprehensive survey," *Sensors (Switzerland)*, vol. 18, no. 9, pp. 1–37, 2018.
- [13] and T. I. Ahlem Ben Hassine, Shigeo Matsubara, "A Constraint-Based Approach to Horizontal Web Service Composition," *Int. Semant. Web Conf.*, vol. 4273, no. November, pp. 172–186, 2006.
- [14] OASIS, "Web Services Quality Factors Version 1.0 Candidate OASIS Standard 01," *OASIS Open 2012*, 2012.
- [15] J. Zhou and X. Yao, "A hybrid artificial bee colony algorithm for optimal selection of QoS-based cloud manufacturing service composition," *Int. J. Adv. Manuf. Technol.*, 2016.
- [16] Y. Li, Z. Luo, and J. Yin, "A LOCATION-AWARE SERVICE SELECTION MODEL," *Serv. Trans. Serv. Comput.*, vol. 1, no. 1, pp. 52–63, 2013.
- [17] X. Mei, F. Zheng, A. Jiang, and S. Li, "QoS aggregation evaluation of web services composition with transaction," in *Proceedings - 2009 International Conference on Information Technology and Computer Science, ITCS 2009*, 2009, vol. 2, pp. 151–155.
- [18] D. Bertram, "Likert Scales. Topic Report," *Fac. Math. – Univ. Belgrade – Serbia*, 2009.
- [19] H. N. Boone, D. A. Boone, and W. Virginia, "Analyzing Likert Data Likert-

- Type Versus Likert Scales,” *J. Ext.*, vol. 50, no. 2, 2012.
- [20] E. Cuevas, M. A. Díaz-Cortés, and D. A. Oliva Navarro, “A swarm global optimization algorithm inspired in the behavior of the social-spider,” in *Studies in Computational Intelligence*, vol. 629, no. 16, 2016, pp. 9–33.
 - [21] J. J. Q. Yu and V. O. K. Li, “A social spider algorithm for global optimization,” *Appl. Soft Comput. J.*, vol. 30, pp. 614–627, 2015.
 - [22] E. Cuevas, M. Cienfuegos, E. Cuevas, and M. Cienfuegos, “A new algorithm inspired in the behavior of the social-spider for constrained optimization,” *Expert Syst. Appl.*, 2013.
 - [23] “Byjus learning App,” *Byjus learning App*. [Online]. Available: <https://byjus.com/intensity-formula/>. [Accessed: 30-May-2020].
 - [24] U. P. Shukla and S. J. Nanda, “A Binary Social Spider Optimization algorithm for unsupervised band selection in compressed hyperspectral images,” *Expert Syst. Appl.*, vol. 97, pp. 336–356, 2018.

Figures

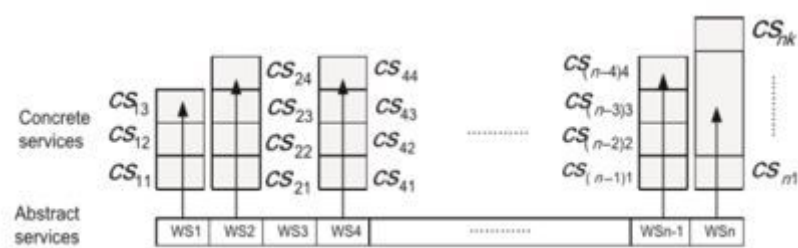


Figure 1

Services selection workflow [7]

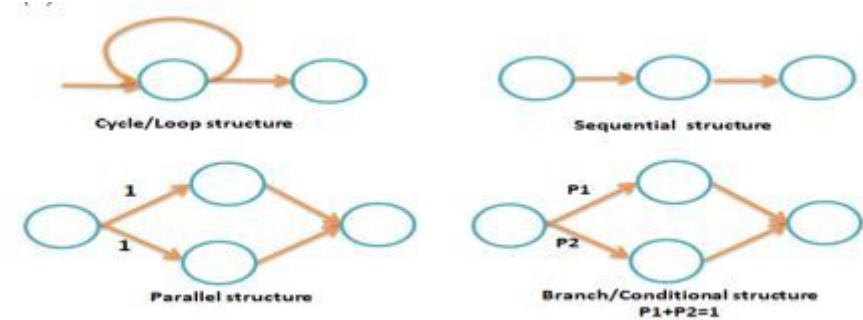


Figure 2

Basic patterns of web service composition[16][8]

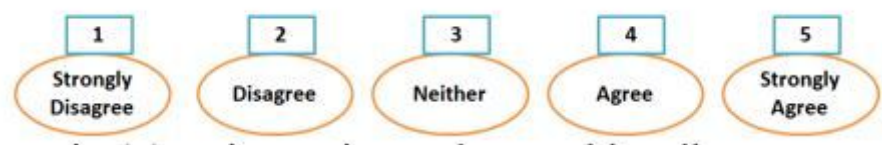


Figure 3

Five-Point scales used in Likert-type Measurement [19]

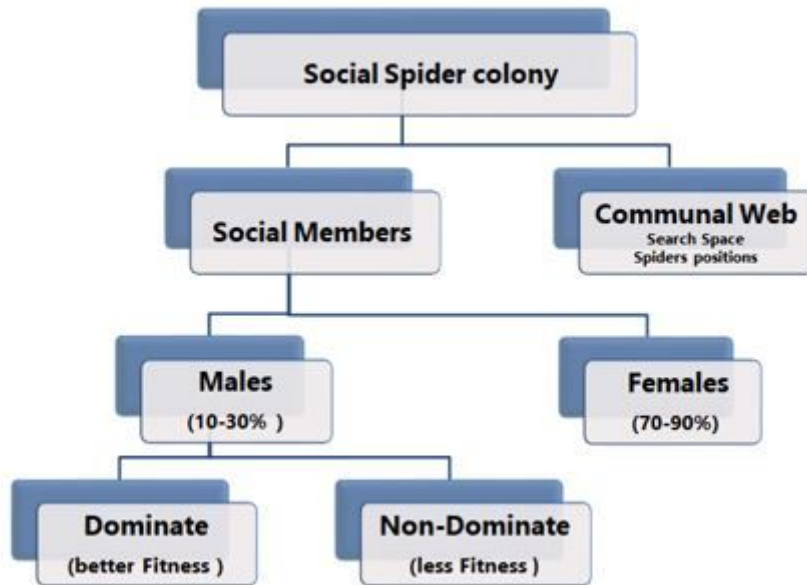


Figure 4

Social Spider Member

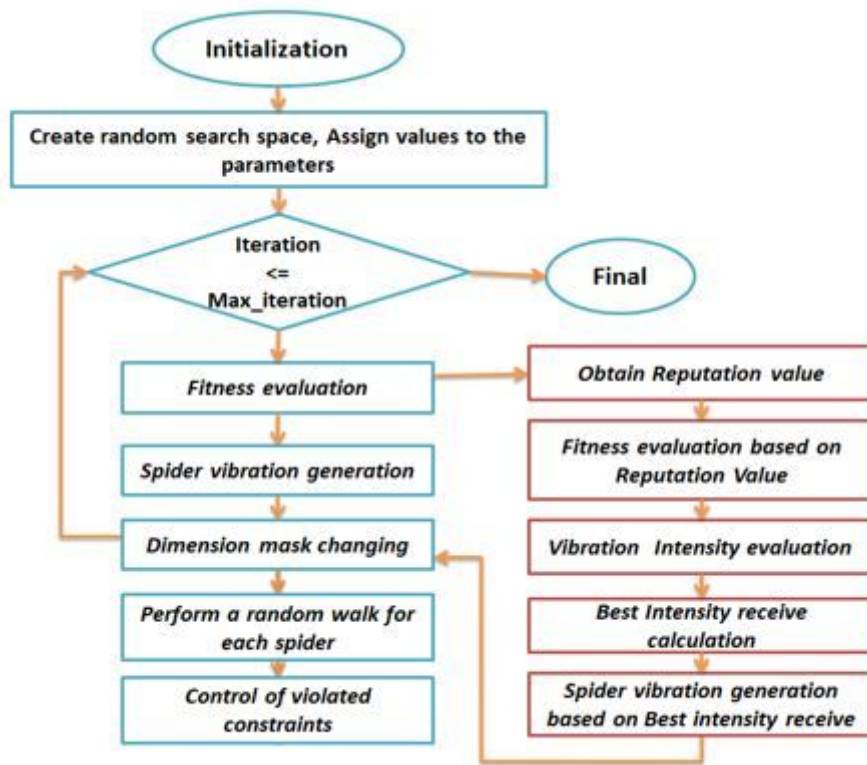


Figure 5

Flow Chart Of Original SSO and RI-SSO operations.[24]

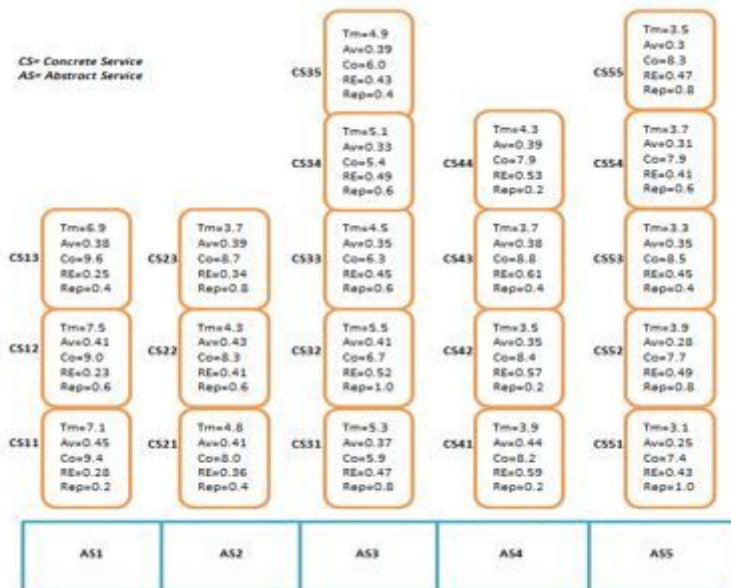


Figure 6

Data Set of Services selection workflow [7]

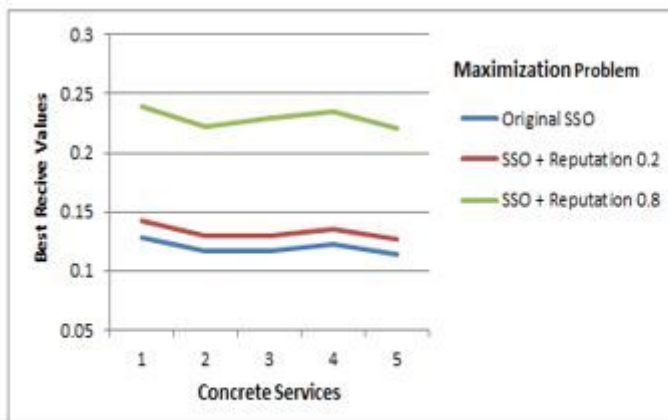


Fig (8.a) Maximization Problem

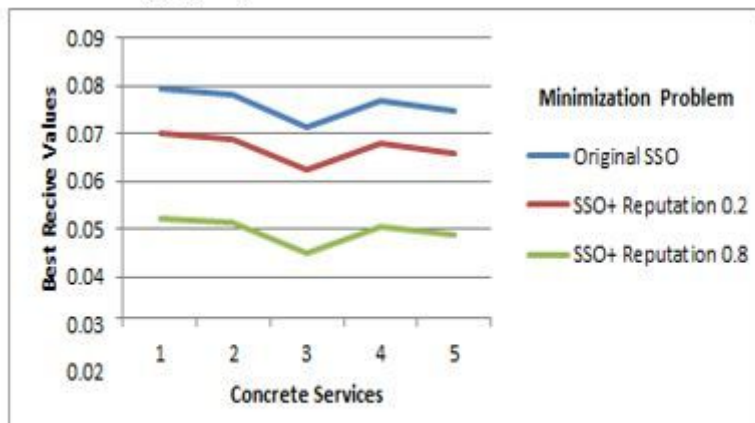


Fig (8.b) Minimization Problem

Figure 7

Evolution curves of Best intensity Receive values

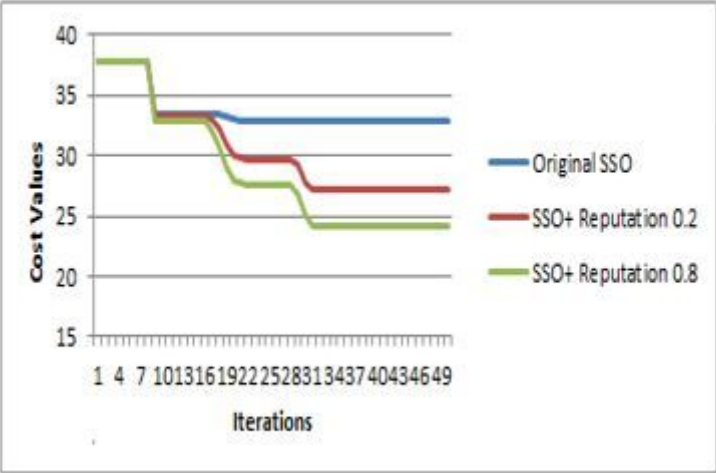


Fig (9.a) Evaluation of Cost Fitness value

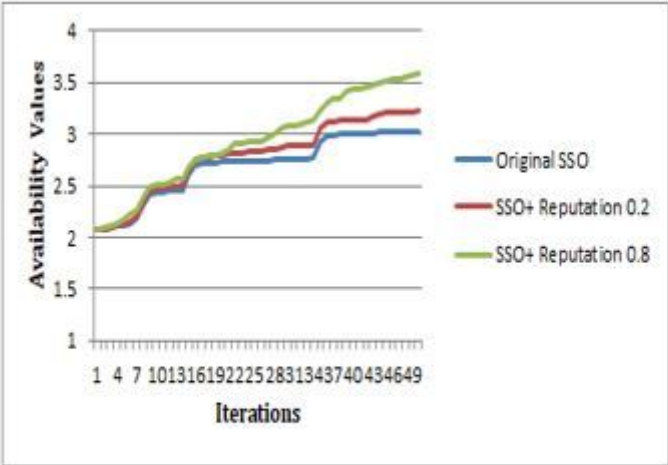


Fig (9.b) Evaluation of Availability Fitness value

Figure 8

Figure 9 a, b

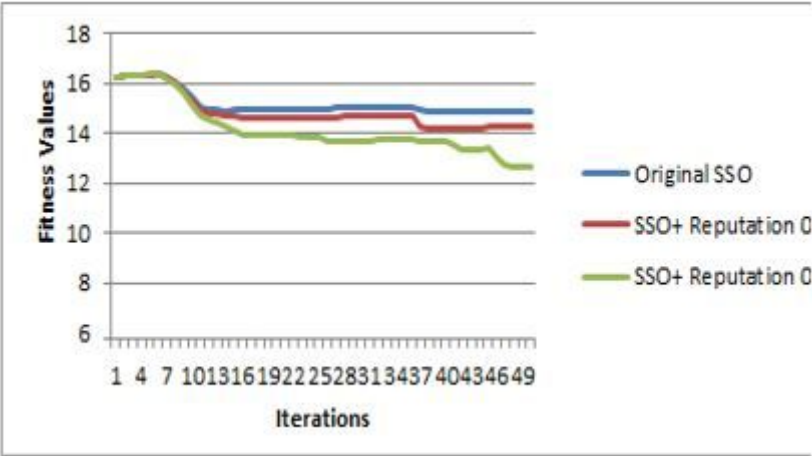


Figure 9

Comparing Results of Fitness Function Values