Retraction Notice

Title of retracted article:	**An Efficient QoS-Aware Services Selection in IoT Using a Reputation Improved-Social Spider Optimization Algorithm**
Author(s):	Aghabi N. Abosaif, Salah Elfaki Elrofai
*Corresponding author's Email:	aghape.nabeel@gmail.com
Journal:	Open Access Library Journal (OALib Journal)
Year:	2022
Volume:	9
Number:	7
Pages (from - to):	1-22
DOI (to PDF):	https://doi.org/10.4236/oalib.1108224
Paper ID at SCIRP:	118782
Article page:	https://www.scirp.org/journal/paperinformation.aspx?paperid=118782
Retraction date:	2022-08-03

Retraction initiative (multiple responses allowed; mark with **X**):

X All authors
☐ Some of the authors:
☐ Editor with hints from
- O Journal owner (publisher)
- O Institution:
- O Reader:
- O Other:

Date initiative is launched: 2022-08-01

Retraction type (multiple responses allowed):

☐ Unreliable findings
- O Lab error
- O Inconsistent data
- O Analytical error
- O Biased interpretation
- O Other:

☐ Irreproducible results
☐ Failure to disclose a major competing interest likely to influence interpretations or recommendations
☐ Unethical research

☐ Fraud
- O Data fabrication
- O Fake publication
- O Other:

☐ Plagiarism ☐ Self plagiarism ☐ Overlap ☐ Redundant publication *
☐ Copyright infringement ☐ Other legal concern:

☐ Editorial reasons
- O Handling error
- O Unreliable review(s)
- O Decision error
- O Other:

X Other: Authors requested withdrawal.

Author's conduct (only one response allowed):

☐ honest error
☐ academic misconduct
X none (not applicable in this case – e.g. in case of editorial reasons)

* Also called duplicate or repetitive publication. Definition: "Publishing or attempting to publish substantially the same work more than once."

Comment:
The Editorial Board would like to extend its sincere apologies for any inconvenience this retraction may have caused.

Open Access Library Journal
2022, Volume 9, e8224
ISSN Online: 2333-9721
ISSN Print: 2333-9705

An Efficient QoS-Aware Services Selection in IoT Using a Reputation Improved-Social Spider Optimization Algorithm

Aghabi N. Abosaif[1]*, Salah Elfaki Elrofai[2]

[1]College of Software Engineering, University of Science and Technology, Omdurman, Sudan
[2]School of Electronic Engineering, Sudan University of Science and Technology, Khartoum, Sudan
Email: *Aghape.nabeel@gmail.com

RETRACTED

How to cite this paper: Abosaif, A.N. and Elrofai, S.E. (2022) An Efficient QoS-Aware Services Selection in IoT Using a Reputation Improved-Social Spider Optimization Algorithm. *Open Access Library Journal*, **9**: e8224.
https://doi.org/10.4236/oalib.1108224

Received: April 15, 2022
Accepted: July 24, 2022
Published: July 27, 2022

Abstract

Internet of Things (IoT) has 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 end-user daily life. The critical challenge is to select the optimal service that satisfies the end-user requirements from similar services functionally and different non-functionality requirements (Quality of Service). This paper proposed a services selection model under QoS constraints in the IoT environment to achieve this challenge. The introduced model implemented a meta-heuristic optimization algorithm and a friendly Likert-type measurement method. It aims to improve bio-inspired optimizing algorithms, called a Social Spider Optimization (SSO) Algorithm, by adding a reputation value to members' weight. It used a Likert-type measurement to evaluate the reputation value of the service. In the experiments, a comparative study was done between an original SSO and the proposed RI-SSO model. The experimental results show the proposed RI-SSO model's efficiency against the original SSO in maximization and minimization problems. It obtains a better outperform in terms of fitness values and execution time.

Subject Areas

Complex Network Models

Keywords

Services Selection Algorithms (SSA), Quality of Services (QoS), Internet of Things (IoT), Likert-Type 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 actuators. These actuators are connected to create new services that facilitate end-user life. The offered services are composite of sub-abstract services connected as one service to perform more complex functions [1]. The provided service from the same actuator type is similar in their functional properties, but they differ in their non-functional (QoS) properties. So, there is a set of concrete services for each abstract service, similar in their functional properties and different in their QoS properties.

The service selection process aims to select an optimal service from a set of concrete services based on Quality of Service (QoS) constraints [2]. ITU-T E.860 defined QoS [3] 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 is formed in several architectures. Authors in [4] proposed QoS architecture to be appropriate for 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 in the Application layer. It represents the highest layer in IoT architecture. Examples of QoS in the application layer are Performance Time, Execution Time, Availability, Services Perform Price, and Reliability [4].

The proposed solution aimed to solve a selection problem using a bio-inspired Meta-heuristic Optimization Algorithm (M-HOP). Meta-heuristic is an artificial intelligence algorithm introduced by Glover in 1986. It comes from combining two Greek terms Meta, which means high-level, and heuristic, which means finding or discovering [2].

Social Spider Optimization Algorithm (SSO) is an M-HOP used by Mousa *et al.* in 2016 [5] to solve the selection problem. Our proposed solution aimed to improve the behavior of SSO to obtain more optimal results in a selection problem. It enhanced the members' weights in the spider colony by increasing or decreasing it based on the service reputation value that the end-users evaluated. The reputation value is also called historical information [6] or end-user feedback information [7]. This information is gathered from the end-user after using a service. The introduced model used an efficiently friendly method to gather this historical information by using a Likert-type measurement. It is called a Reputation Improved-Social Spider Algorithm (RI-SSO) because it improved the behavior of SSO based on gathered reputation information.

The rest of the paper is organized as follows: Section 2 introduced the related works. Sections 3 and 4 show the services in the IoT environment and QoS composition model, respectively. In Section 5, a composed services selection model is discussed. The Likert-type measurement and fitness function are defined in Section 6 and Section 7, respectively. The original social spider optimization algorithm is explained in Section 8. The proposed model of the Reputation Improved-Social Spider Optimization (RI-SSO) algorithm is described in Section 9. Section 10 shows the experiment settings and dataset. Section 11 explored the conclusion—finally, the recommendation and future work discussed in Section 12.

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 of a services selection process in the IoT environment. Authors in [8] [9] [10] [11] applying a Meta-heuristic algorithm by considering the objective QoS information, which is supported by the service providers. Li *et al.* [9] introduced an optimization solution in IoT and Cloud computing environments in the cloud logistics platform. They termed the services selection problem as a constraint satisfaction problem (CSP). The proposed dynamic model depends on Practical Swarm Optimization (PSO) algorithm. They considered four QoS constraints (availability, reliability, time, and cost). Abosaif *et al.* [10] presented their Likert-Improved-PSO model and evaluated its performance by compare it with the original PSO and Improved-PSO. The introduced model had a lower execution time and it obtained better fitness value. For future research, they recommended testing more QoS factors and combining with more than one meta-heuristic algorithm with respect to the end user conditions. Liu *et al.* [11] 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.* [8] 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 [6] [7] [12] emphasize the subjective QoS feedback information that comes from end-user after using selected services. Mejri *et al.* [7] adopted a self-adaptive approach to ensure scalability in the IoT environment. The introduced method 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 services' consumer. They optimized two QoS factors: execution time and reliability. Nwe *et al.* in [6] showed 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 are 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.* [5] proposed SSO as a solution for services selection process-aware QoS constraints. The proposed model was introduced in the general web services environment by considering the objective QoS information only. The authors satisfied 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 [12], Divyad *et al.* proposed ranking the registry services based on end-user evaluation or feedback to enhance selection and discovery processes. They presented 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 selection process metrics. 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 Process in IoT Environment

In the IoT environment, there are numerous benefits and facilities provides to the end-users. These facilities are healthcare, 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.

There are many concrete services for each abstract service, similar to their functional properties but different in non-function properties.

The proposed model applies 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 [9] [13], as in **Figure 1**. It is more appropriate for the IoT environment, containing 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], enabling the user to implement and modify the abstract workflow when required.

4. QoS Composition Model

The selection of an optimal service that meets end-user requirements depends on QoS factor values distinguished from one service to another. The introduced model identified five QoS factors in the application layer. These factors are selected from two quality types [14].

The first type is the Business Quality Type (BQT): it means an economic value 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:

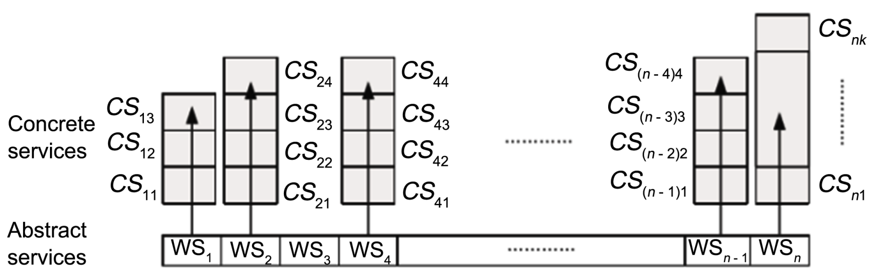


Figure 1. Services selection workflow [9].

Availability $q_{AV}(s)$ refers to a probability ratio that the service is running and accessing 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 measured by seconds.

Also, the proposed model regarded other factor:

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

To optimize the QoS value, the behavior of factors is varied from one factor to another [5]. Some factors are optimized by getting the minimum values; they are called Negative QoS factors. Other factors are optimized by getting the maximum values; they are called Positive QoS factors.

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

$$\bar{Q}i(CS)=\begin{cases}\dfrac{Qi^{\max}-Qi(CS)}{Qi^{\max}-Qi^{\min}} & Qi^{\max}-Qi^{\min}\neq 0\\ 1 & Qi^{\max}-Qi^{\min}=0\end{cases} \tag{1}$$

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

$$\bar{Q}i(CS)=\begin{cases}\dfrac{Qi(CS)-Qi^{\min}}{Qi^{\max}-Qi^{\min}} & Qi^{\max}-Qi^{\min}\neq 0\\ 1 & Qi^{\max}-Qi^{\min}=0\end{cases} \tag{2}$$

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

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

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) [11] [15], as shown in **Figure 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 perform 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 is calculated by aggregating each factor's corresponding values. 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

Table 1. Aggregation function to compute the QoS factors [19].

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

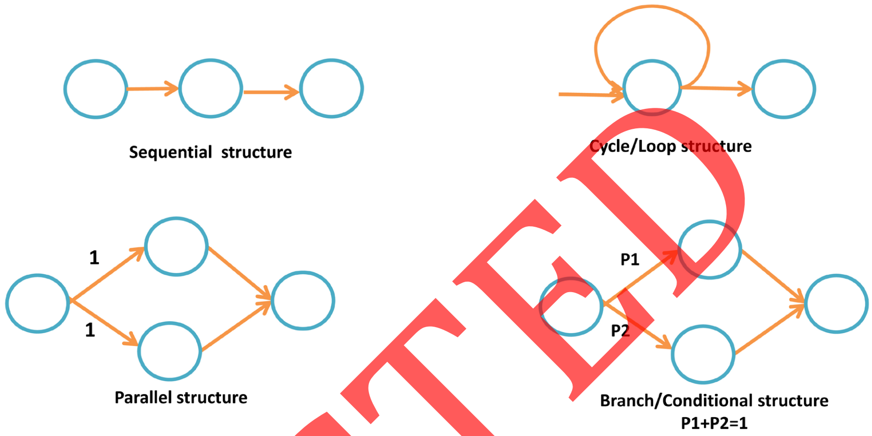


Figure 2. Basic patterns of web service composition [11] [18].

$q_{RP}(s)$ factors. The multiplicative function applies for availability $q_{AV}(s)$ and reliability $q_{RE}(s)$ factors.

6. Fitness Function

The objective function or fitness value is 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

$$\begin{aligned} &= \text{Min}\left(W_{RT} * F_{RT}\left(CS_{ij}\right)\right) + \text{Min}\left(W_{EP} * F_{EP}\left(CS_{ij}\right)\right) + \text{Max}\left(W_{AV} * F_{AV}\left(CS_{ij}\right)\right) \\ &\quad + \text{Max}\left(W_{RE} * F_{RE}\left(CS_{ij}\right)\right) + \text{Max}\left(W_{RP} * F_{RP}\left(CS_{ij}\right)\right) \end{aligned} \tag{4}$$

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

W_{QoS} represents the weight for each factor. It is calculated as in Equations (5) and (6).

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

$$W_{QoS} = W_{RT} + W_{EP} + W_{AV} + W_{RE} + W_{RP} = 1 \tag{6}$$

where i represents the number of QoS factors ($1 < i < 5$).

7. Likert-Type

Likert-type is defined in [16] 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." Dr. Rensis Likert named the scale measurement 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 [17]. The scales starting ranging from "Strongly Disagree", "Disagree", "Neither", "Agree", and "Strongly Agree" as in **Figure 3**.

Each scale is assigned to coding like using numeric value or alphabet value. This value is used to measure the attitude under investigation, usually starting at one and incremented by one for each level.

They introduced the 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 is a reputation value to each service. In the beginning, the reputation value is set to equal zero. The median value is not the mean value of each service calculated to get a more precise answer to analyze the collected feedback data. The median used because our solution used the Likert-type does not use a Likert scale [17].

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 middle one's position by dividing 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 is the reputation value for service s_i, is equal to 0.6.

8. An Original Social Spider Optimization Algorithm

SSO is a swarm intelligence algorithm that emulates the collective behavior of

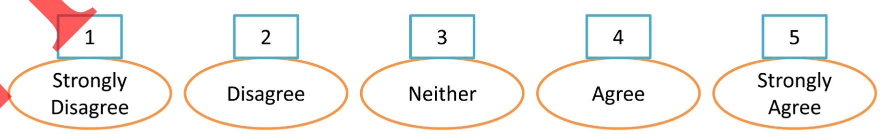


Figure 3. Five-point scales used in Likert-type measurement [17].

Table 2. Reputation values of 5-point 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

spider swarms. It proposed by Cuevas *et al.* in [20] to find an optimal solution to complex optimization problems in continuous search space. In this article, the original SSO and proposed RI-SSO used to solve the services selection problem in discrete search space, namely the Nearest Integer method [5].

One of the SSO colonies' primary behaviors is mating behavior; females used the male spider's vibrations over the web to determine 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 **Figure 4**. *Females group*: F represent 65% - 90% of the total colony members, $F = \{f_1, f_2, f_3, \cdots, f_{Nf}\}$. *Males group*: M represent 35% - 10% of the total colony members, $M = \{m_1, m_2, m_3, \cdots, m_{Nm}\}$. Whereas $pop_s = F \cup M$, $S = \{s_1, s_2, s_3, \cdots, s_N\}$, so $pop_s = \{s_1 = f_1, s_2 = f_2, \cdots, s_{Nf} = f_{Nf}, s_{Nf+1} = m_1, s_{Nf+2} = m_2, \cdots, s_N = m_{Nm}\}$. 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

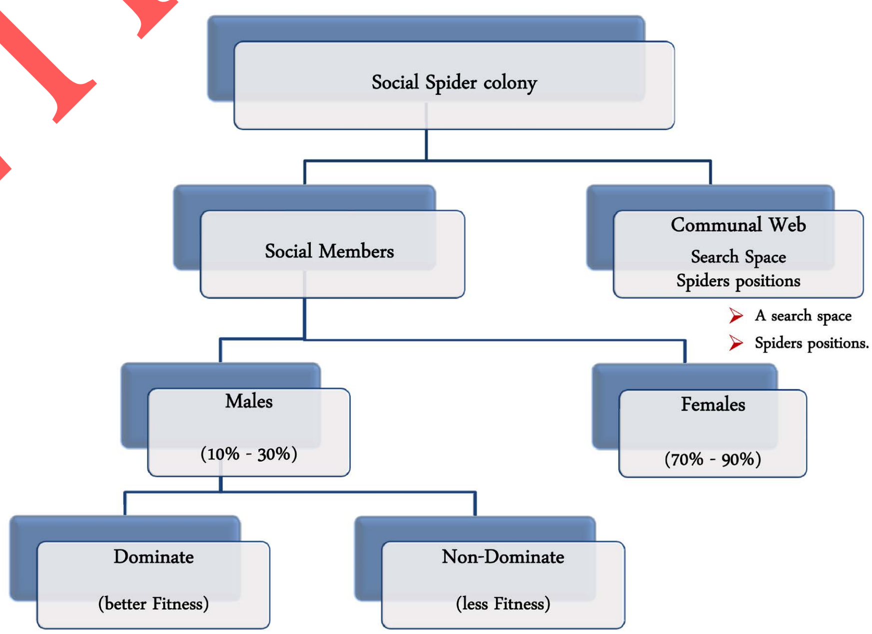


Figure 4. Social spider member.

quality. The weight value of each solution s_i in population pop_s calculated by Equation (7):

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

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 population's solutions.

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 is used to guide a spider *s* to new positions, consisting 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 during the last iteration.

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 on the web. Vibrations' strength is affected by two properties; the *intensity source and the intensity attenuation*. The *intensity source* of the vibration is also called the *weight factor*: it is calculated by fitness function as in Equation (8) in the range [0, +∞] [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.

$$I_S(P_a, P_b, t) = \log\left(\frac{1}{f(P_a) - C} + 1\right) \tag{8}$$

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) considering the physical energy phenomenon of vibration attenuation during the web propagation process.

$$I_d\left(P_a, P_b, t\right) = \exp\left(-\frac{D\left(P_a, P_b\right)}{\bar{\sigma} * r_a}\right) \tag{9}$$

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

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

$$I\left(P_a, P_b, t\right) = I_S\left(P_a, P_b, t\right) \times I_d\left(P_a, P_b, t\right) \tag{10}$$

$$I\left(P_a, P_b, t\right) = \log\left(\frac{1}{f\left(P_a\right) - C} + 1\right) \times \exp\left(-\frac{D\left(P_a, P_b\right)}{\bar{\sigma} * r_a}\right) \tag{11}$$

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 most robust vibration value V^{best} from *V*, in maximization problems, the 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 $P_{s,i}^{folw}$ based on the mask for S As in Equation (12):

$$P_{s,i}^{folw} = \begin{cases} P_{s,i}^{tar}, & M_{s,i} = 0 \\ P_{s,i}^{r}, & M_{s,i} = 1 \end{cases} \tag{12}$$

where, $P_{s,i}^{tar}$ is the *ith* element of the source solution of V^{tar}, $P_{s,i}^{r}$ is a random solution's position, r is a random integer value $\in \left[1, |pop_s|\right]$, and $M_{s,i}$ is the *ith* dimension mask M of the spider S.

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

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

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.

$$P_{s,i}(t+1) = \begin{cases} \left(\overline{X_i} - P_{s,i}(t)\right) \times r, & P_{s,i}(t+1) > \overline{X_i} \\ \left(P_{s,i}(t) - \underline{X_i}\right) \times r, & P_{s,i}(t+1) < \underline{X_i} \end{cases} \tag{14}$$

where, $\overline{X_i}$ is the upper bound on the search space, and $\underline{X_i}$ is the lower bound.

9. Proposed Model

The proposed model builds 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 has done between 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 are 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. 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 improves the next service selection process based on the previous selection process's collected evaluation.

The second point, the behavior of attraction or dislike between females and males has been evolved based on propagating the vibrations across the web from males to females. In the 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 is defined as "the quantity of energy the wave conveys per unit time across a surface of unit area" [23]. The intensity formula is:

$$I = \frac{P}{A} \tag{15}$$

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, enhancing the attraction of dominated males and vice versa.

Our optimization model summarizes in the following steps as in **Figure 5** where blue processes represent the Original SSO, and the red processes are the addition improvement:

Step 1: In the initial step, the selection process implements as in the Original SSO, and the reputation value is set to equal zero.

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

Step 3: The reputation value is 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: Evaluate the intensity source value by adding the reputation value to the QoS fitness value as in Equations 16(a)) and 16(b). This update applied in the Equation (11) of the original SSO and is calculated as follow:

For maximization optimization problem, the intensity source is calculated as:

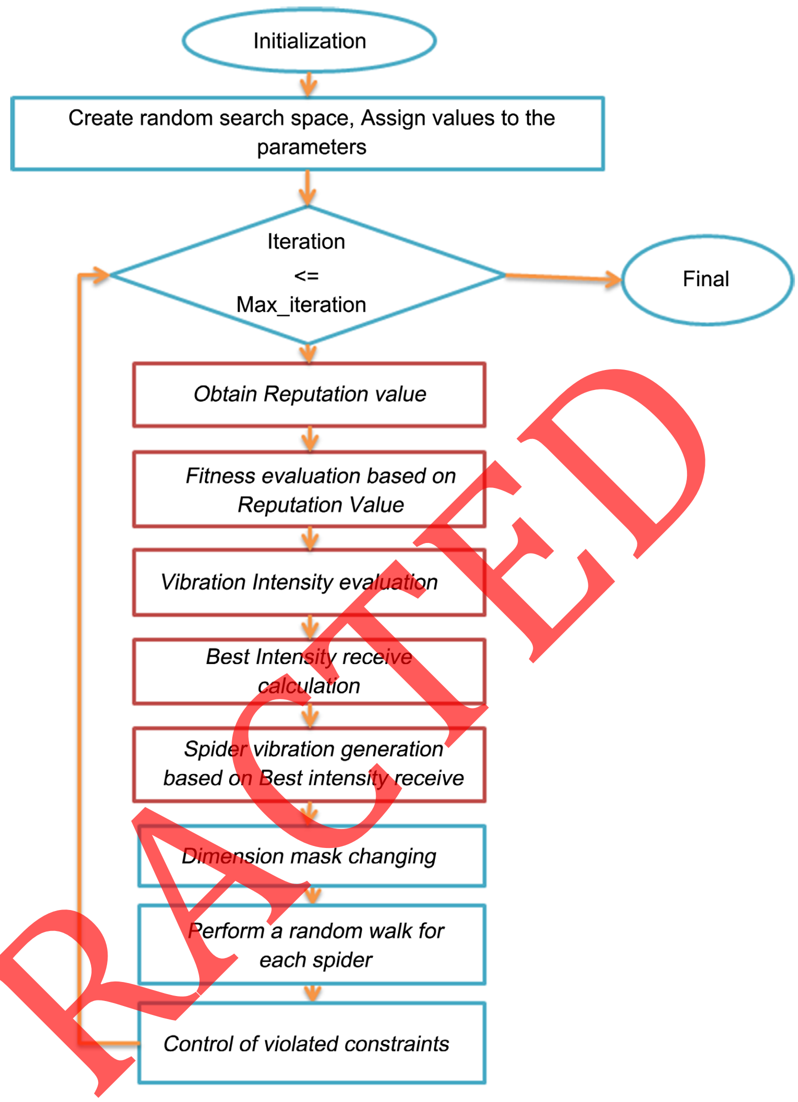


Figure 5. Flow chart of original SSO and RI-SSO operations [24].

$$I_S\left(P_a, P_b, t\right) = \log\left(\frac{1}{\left(f\left(P_a\right) - \left(f\left(P_a\right) * Rep\_val\right)\right) - C} + 1\right) \tag{16(a)}$$

For minimization optimization problem the intensity source is calculated as:

$$I_S\left(P_a, P_b, t\right) = \log\left(\frac{1}{\left(f\left(P_a\right) + \left(f\left(P_a\right) * Rep\_val\right)\right) - C} + 1\right) \tag{16(b)}$$

where, $f\left(P_a\right)$ 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 follows:

$$\left(P_a, P_b, t\right) = I_S\left(P_a, P_b, t\right) \times \exp\left(-\frac{D\left(P_a, P_b\right)}{\bar{\sigma} * r_a}\right) \tag{17}$$

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

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

$$\begin{aligned}&\text{Target viberation} = \big(\big(\text{Target viberation} * \text{Target Matrix}\big)\\&+\big(\text{Max Position} * 1 - \text{Target Matrix}\big)\big) + \text{Best Intensity Recive}\end{aligned} \tag{18}$$

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

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

10. Experiment Settings and Dataset

To evaluate the performance of the proposed approach RI-SSO, it is compared with the original SSO proposed by (Mousa and Bentahar, 2016) [5] to solve the selection problem. The dimension of the selection problem is represented by the number of items in an integer array-coding scheme. The array elements represent the indexes of candidate service.

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 **Figure 1** and **Figure 6**. 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 the first four factors (Execution Time, Availability, Cost, and Reliability) take from (Li, 2013) [9], see **Figure 6**, 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 were implemented on a laptop with Windows 10, 2.90 GHz processor, and 8-GB RAM. The algorithms coded in MATLAB R2017b.

Performance Comparison

To analyze the RI-SSO model's performance, the behavior of Original SSO compared with RI-SSO in two situations, one when the reputation value equals 0.2, which is the minimum point in Likert-type measurement. Other when the reputation value equals 0.8, which is the previous maximum point in Likert-type measure.

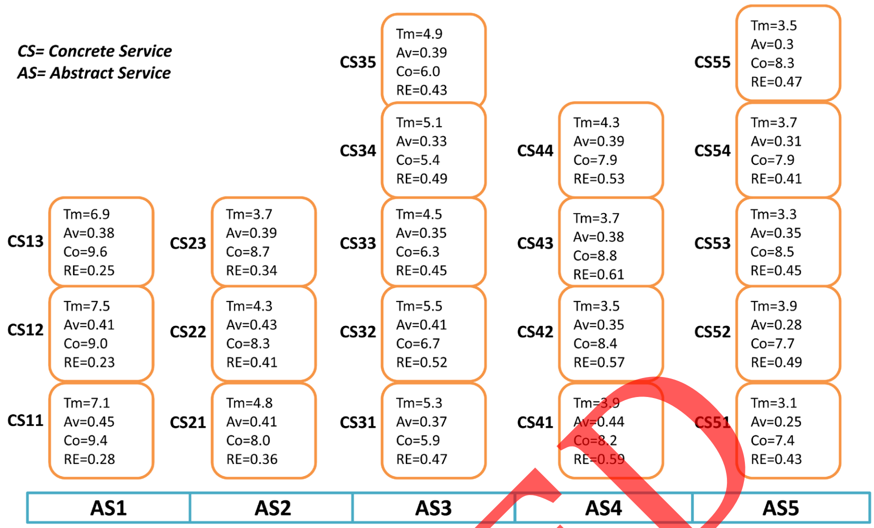


Figure 6. Data set of services selection workflow [9].

The comparison is made based on three evaluations: firstly, it recorded the best intensity receive values obtain from the three comparisons in minimization and maximization problems. Results show that the increases in reputation value lead to an increase in the best intensity received values in the maximization problem, as in Figure 7(a). Furthermore, it leads to a decrease in the best intensity receives values in minimization problems, as in Figure 7(b).

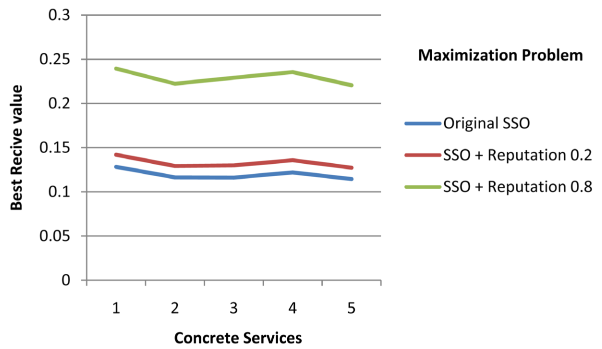
Secondly, we observed the fitness values over the two optimization types: We attended the availability factor's fitness value for a maximization problem. Results are shown in Figure 8(a); the RI-SSO with reputation 0.8 gets the best performance by obtaining the highest fitness values than reputation 0.2 and Original SSO. The RI-SSO with reputation value equals 0.2 improves the availability of fitness value by 2% - 6%, and when a reputation value equals 0.8, the fitness value improves by 17% - 57%.

For the minimization problem, we observed the fitness value of the cost factor. Results are shown in Figure 8(b); 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. The RI-SSO with reputation value equals 0.2 improves the cost fitness value by 9% - 17%, and when a reputation value equals 0.8, the fitness value improves by 16% - 26%.

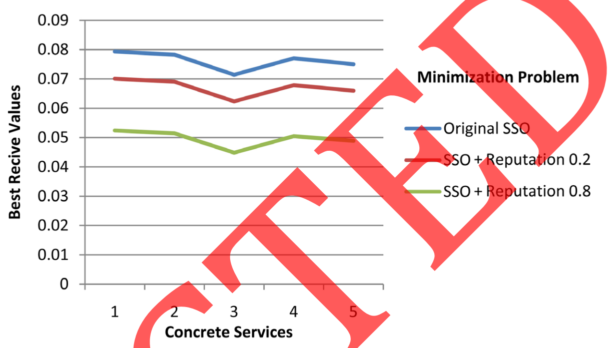
Thirdly, to approve the feasibility and efficiency of RI-SSO, the QoS fitness values are calculated to optimize five factors. The weights of factors are 0.25, 0.25, 0.15, 0.15, and 0.20 for response time, availability, execution price, reliability, and reputation, respectively, as in Equation (19).

$$\begin{aligned}&\text{QoS fitness function}\\&=\mathrm{Min}\left(0.25*F_{RT}\left(CS_{ij}\right)\right)+\mathrm{Max}\left(0.25*F_{AV}\left(CS_{ij}\right)\right)\\&\quad+\mathrm{Min}\left(0.15*F_{EP}\left(CS_{ij}\right)\right)+\mathrm{Max}\left(0.15*F_{RE}\left(CS_{ij}\right)\right)\\&\quad+\mathrm{Max}\left(0.20*F_{RP}\left(CS_{ij}\right)\right)\end{aligned}\tag{19}$$

where, $F_{QoS}(CS_{ij})$ represents the summation function for each factor.

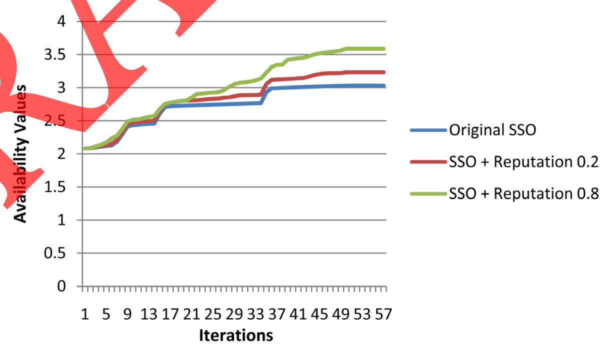


(a)

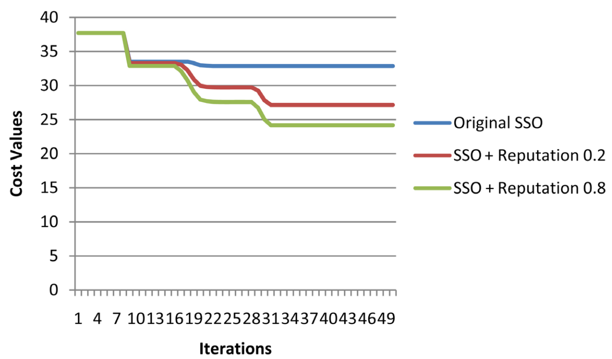


(b)

Figure 7. (a): Evolution curves of best intensity receive in maximization problem; (b) Evolution curves of best intensity receive in minimization problem.



(a)



(b)

Figure 8. (a): Evaluation of availability fitness value; (b): Evaluation of cost fitness value.

The result is shown in **Figure 9** 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. As shown in **Figure 9**, SSO with a high reputation obtained the fitness value 14.6 in iteration eleven, the SSO with a low reputation obtained it at iteration eighteen, and the original SSO obtained it at iteration forty-four. The results show that the RI-SSO with a reputation value equal to 0.2 can improve the selection process by 7% - 10%. While it can increases the performance by 21% - 26% when a reputation value equals 0.8.

Also, to validate the RI-SSO's obtained results, its fitness value was compared with the obtained fitness value proposed by (Wenfeng *et al.*, 2013) [9]. The implementation applied the same dataset, shown in **Figure 6**, and QoS factors' exact weights. The weights of QoS factors are 0.28, 0.24, 0.3, and 0.18 for response time, availability, cost, and reliability. They are thus demonstrating that the RI-SSO can perform more efficiency to optimize a selection problem. It got high availability and reliability values while keeping low cost and execution time values.

The results show that the RI-SSO is performed better than PSO, which was proposed in [9]. In [9], the minimum objective function value is 18.08 when the number of iterations is 15. This value 18.08 is obtained in iteration 8 when applied to the RI-SSO, as in **Figure 10**. This means that the number of iteration in PSO is more than that in RI-SSO.

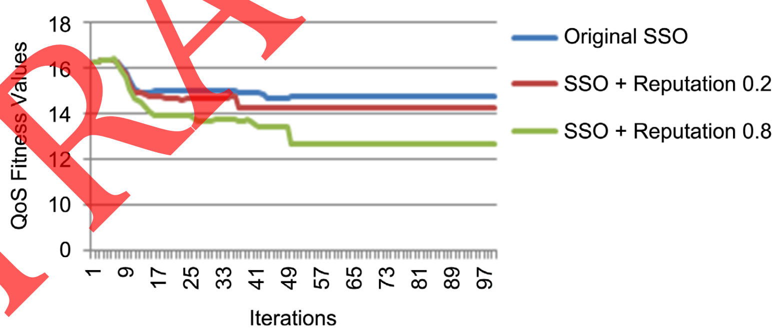


Figure 9. Comparing results of fitness function values.

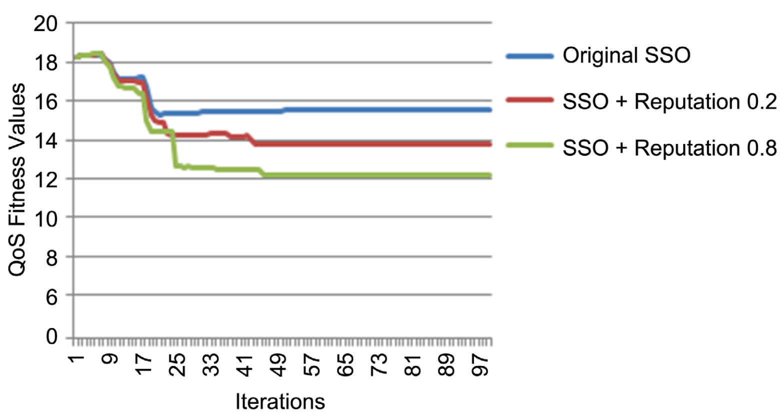


Figure 10. Results of fitness function values.

11. Conclusions

To facilitate the end-user lives, IoT environments contain a massive number of actuators that provide different services. Similar actuators can provide services with similar functionality but various non-functional requirements. So the different end-users request the same service with different QoS criteria. Thus making the 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 is classified into two parts: *First part* is collecting feedback information about services from end-users who used it—this information is 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 the 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 search process's reliability by selecting a service that appropriates the end-user preference.

The comparison studies find that the proposed approach RI-SSO has an extreme fitness function value than the original SSO. The simulation results approve that 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 applications. It 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.

Authors' Contributions

The two main contributions of this article are:

- Collect the end-user feedback information using a friendly method of Likert-type measurement in Services selection Algorithms and keep it as a reputation value for each service.
- Improve the Social Spider Optimization Algorithm's performance by using the collected Reputation value and adding it to the weight of each spider on the web.

Acknowledgements

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

Availability of Data and Materials

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

Conflicts of Interest

The authors declare that they have no competing financial interests or personal relationships that could have influenced the work reported in this paper.

References

[1] Asghari, P., Rahmani, A.M., Haj, H. and Javadi, S. (2018) Service Composition Approaches in IoT: A Systematic Review. *Journal of Network and Computer Applications*, **120**, 61-77. https://doi.org/10.1016/j.jnca.2018.07.013

[2] Abosaif, A.N. and Hamza, H.S. (2020) Quality of Service-Aware Service Selection Algorithms for the Internet of Things Environment: A Review Paper. *Array*, **8**, Article ID: 100041. https://doi.org/10.1016/j.array.2020.100041

[3] ITU-T (2002) Recommendation E.860: Framework of a Service Level Agreement. Overall Network Operation, Telephone Service, Service Operation and Human Factors.

[4] Li, L., Li, S. and Zhao, S. (2014) QoS-Aware Scheduling of Services-Oriented Internet of Things. *IEEE Transactions on Industrial Informatics*, **10**, 1497-1507. https://doi.org/10.1109/TII.2014.2306782

[5] Mousa, A. and Bentahar, J. (2016) An Efficient QoS-aware Web Services Selection using Social Spider Algorithm. *Procedia Computer Science*, **94**, 176-182. https://doi.org/10.1016/j.procs.2016.08.027

[6] New, N.H.W., Bao J.-M. and Cui, G. (2014) Flexible User-Centric Service Selection Algorithm for Internet of Things Services. *Journal of China Universities of Posts and Telecommunications*, **21**, 64-70. https://doi.org/10.1016/S1005-8885(14)60510-0

[7] Mejri, M. and Ben Azzouna, N. (2017) Scalable and Self-Adaptive Service Selection Method for the Internet of Things. *International Journal of Computers and Applications*, **167**, 43-49. https://doi.org/10.5120/ijca2017914542

[8] Elhoseny, M., Abdelaziz, A., Salama, A.S., Riad, A.M., Muhammad, K. and Sangaiah, A.K. (2018) A Hybrid Model of Internet of Things and Cloud Computing to Manage Big Data in Health Services Applications. *Future Generation Computer Systems*, **86**, 1383-1394. https://doi.org/10.1016/j.future.2018.03.005

[9] Li, W., Zhong, Y., Wang, X. and Cao, Y. (2013) Resource Virtualization and Service Selectionin Cloud Logistics. *Journal of Network and Computer Applications*, **36**, 1696-1704.

[10] Abosaif, A.N. and Elrofai, S.E. (2020) QOS—Aware Meta-Heuristic Services Selection Algorithm and Likert Scale Measurement for IOT Environment. *International Journal of Computer Science Trends and Technology*, **8**, 1-8.

[11] Liu, J., *et al.* (2013) A Cooperative Evolution for QoS-Driven IoT Service Composition. *Automatika*, **54**, 438-447. https://doi.org/10.7305/automatika.54-4.417

[12] Priyadharshini, V., DIvya, P., Preethi, D., Pazhaniraja, N. and Paul, P.V. (2015) A Novel Web Service Publishing Model Based on Social Spider Optimization Technique. *4th IEEE Sponsored International Conference on Computation of Power, Energy, Information and Communication*, Melmaruvathur, 22-23 April 2015, 373-387. https://doi.org/10.1109/ICCPEIC.2015.7259488

[13] Hassine, A.B., Matsubara, S. and Ishidaa, T. (2006) A Constraint-Based Approach to Horizontal Web Service Composition. In: *International Semantic Web Conference*, Vol. 4273, Springer, Berlin, Heidelberg, 130-143. https://doi.org/10.1007/11926078_10

[14] OASIS (2012) Web Services Quality Factors Version 1.0 Candidate OASIS Standard 01. *OASIS Open* 2012.

[15] Zhou, J. and Yao, X. (2016) A Hybrid Artificial Bee Colony Algorithm for Optimal Selection of QoS-Based Cloud Manufacturing Service Composition. *International Journal of Advanced Manufacturing Technology*, **88**, 3371-3387. https://doi.org/10.1007/s00170-016-9034-1

[16] Bertram, D. (2009) Likert Scales. Topic Report, University of Belgrade, Beograd, Serbia.

[17] Averin, A.D., Yakushev, A.A., Maloshitskaya, O.A., Surby, S.A., Koifman, O.I. and Beletskaya, I.P. (2017) Synthesis of Porphyrin-Diazacrown Ether and Porphyrin-Cryptand Conjugates for Fluorescence Detection of Copper(II) Ions. *Russian Chemical Bulletin*, **66**, 1456-1466. https://doi.org/10.1007/s11172-017-1908-3

[18] Li, Y., Luo, Z. and Yin, J. (2013) A Location-Aware Service Selection Model. *International Journal of Services Computing*, **1**, 52-63. https://doi.org/10.29268/stsc.2013.1.1.5

[19] Mei, X., Zheng, F., Jiang, A. and Li, S. (2009) QoS Aggregation Evaluation of Web Services Composition with Transaction. *Proceedings of the* 2009 *International Conference on Information Technology and Computer Science*, **2**, 151-155. https://doi.org/10.1109/ITCS.2009.171

[20] Cuevas, E., Díaz-Cortés, M.A. and Oliva Navarro, D.A. (2016) A Swarm Global Optimization Algorithm Inspired in the Behavior of the Social-Spider. In: *Studies in Computational Intelligence*, Vol. 629, Springer, Cham, 9-33. https://doi.org/10.1007/978-3-319-28503-0_2

[21] Yu, J.J.Q. and Li, V.O.K. (2015) A Social Spider Algorithm for Global Optimization. *Applied Soft Computing*, **30**, 614-627. https://doi.org/10.1016/j.asoc.2015.02.014

[22] Cuevas, E., Cienfuegos, M., Cuevas, E. and Cienfuegos, M. (2013) A New Algorithm Inspired in the Behavior of the Social-Spider for Constrained Optimization. *Expert Systems with Applications*, **40**, 6374-6384. https://doi.org/10.1016/j.eswa.2013.05.041

[23] (2020) BYJUS Learning App. https://byjus.com/intensity-formula/

[24] Shukla, U.P. and Nanda, S.J. (2018) A Binary Social Spider Optimization Algorithm for Unsupervised Band Selection in Compressed Hyperspectral Images. *Expert Systems with Applications*, **97**, 336-356. https://doi.org/10.1016/j.eswa.2017.12.034

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