RESEARCH ARTICLE OPEN ACCESS

QOS–Aware Meta-Heuristic Services Selection Algorithm and Likert Scale Measurement for IOT Environment

Aghabi Nabil Abosaif [1], Salah Elfaki Elrofai [2]
Department of Computer Science [1],Sudan University of Science and Technology, Sudan
School of Electronic Engineering Department [2],Sudan University of Science and Technology, Sudan

ABSTRACT

Internet of Things (IoT) has been evolved continuously over the last decade to connect hundreds of objects/devices to provide end-users with services that enhance their daily lives. End-users require services that are typically composed of other services to fulfill their needs. Moreover, users may demand specific Quality-Of-Service (QoS) when they request their services. Accordingly, with the increase in the number of services with diverse QoS, the problem of selecting and composing services that match the required QoS constraints is becoming critical, yet challenging. To this end, this paper attempts to solve the services selection problem by taking into account the end-user feedback in a user-friendly way.

It proposed to use a Likert scales measurement with Improved Practical Swarm Optimization algorithm (Improved-PSO) to enhance the End-user requirement. Our proposed approach aimed to improve the performance of the selection process and selection time. It was to be done based on user evaluation and enhance the behavior of it to be more intelligent and faster. The effectiveness and efficiency of the proposed approach were verifying by comparing the performance of the proposed approach with the performance of PSO and Improved-PSO only without regard to the end-user evaluations. The results show that the proposed approach gets a more optimal solution and has a less execution time than a compared PSO, and Improved-PSO algorithms.

Keywords: — Services Selection Algorithms (SSA), Quality of Services (QoS), Internet of Things (IoT), Optimization Objectives, Likert Scales Measurement.

I. INTRODUCTION

In recent years, the Internet of things (IoT) technology has widely concerned by researchers. The IoT environment, sensors, and devices exchange data and information through IoT networks using standard, global protocols. The Web Services Description Language (WSDL) allows an interoperable machine-to-machine interaction over the network[1] through a web service (WS).

Web services[2] defined as self-contained programs that can execute through the Web. Mostly a WS depend on three critical standard concepts Web Service Definition Language (WSDL), Simple Object Access Protocol (SOAP), and the Universal Description, Discovery, and Integration (UDDI). There are three main components in the WS architecture[3], as in Fig.[1]:

Service Provider (SP): who provides different services through the interfaces to create implement and publish a web service by using the (UDDI) specification.

Service Consumer (SC): who request, consume services and regards the end-user of a web service. SC uses the service registry to gain information about services and access to it.

Service Registry (SR): contain information about different services provided based on the UDDI specification where services are listed and advertised for search.

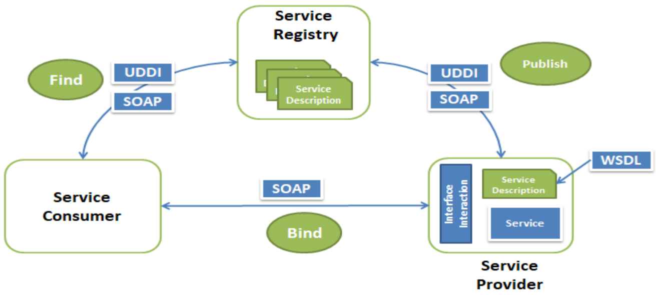

Fig.[1]: Web Service Components [3]

Services composition is the process of integrating and collecting more than one services into a single service to perform more complex functions[4]. A service selection process regards the heart of service composition[5].

These selected services consist of more than one candidate service, which meets the user preference of QoS constraints. Therefore, the optimal matched services must be selected and grouped into constructed composite service collection. To this end, many types of research have been developed to address QoS aspects for service selection schemes for different purposes. Meta-heuristic optimization algorithm (MHOP) used widely to solve the services selection problem. Our approach intended to use a Meta-heuristic optimization algorithm (MHOP) with Likert scale measurements as a friendly way to evaluate the end-user feedback. A Meta-heuristic is artificial intelligence and dependent-problem optimization algorithm that was introduced by Glover in 1986.

It comes from the combination of the Greek prefixes meta- (metá, which means the high-level) plus heuristic (heuriskein or euriskein, which means "I find, discover"). It is a high-level independent-optimization problem that identifies a set of concepts grouped in an algorithmic framework to provide a set of strategies or rules that develop and improve heuristic optimization algorithms to find a suitable solution for a specific problem.[6][7]. One of the most using Meta-heuristic algorithms to solve the optimization problem for services selection is Practical Swarm Optimization (PSO). PSO is a subset of evolutionary computation, which is a subfield of artificial intelligence; it will explain in section seven.

The rest of this paper organizes as follows. The next section formally introduces the related works. Section three shows a service selection workflow. In section four, a QoS in the IoT environment discussed. Quality of service model and objective functions explain in section five. The Likert scale measurement defines in section six. Section seven explains the Practical Swarm Optimization algorithm (PSO). Section eight proposed a services selection model. The experiment settings and experimental results show in section nine. Finally, section ten contains the conclusion.

II. RELATED WORK

There are many solutions proposed for a service selection problem as show in Table1. This section presents some related work on IoT and CMfg environments, using meta-heuristic algorithms.

Authors in[8] and [9] designed their algorithms based on EV Algorithms without regard to any feedback or experience that comes from users in their solution. In 2016, Anas et al. [8] aimed to take benefits from both human thinking to make a decision when facing multiple choices and data collected from IoT sensors. They took an example of a human decision in driving to trade-offs in time, distance, or cost in general. The main point is how to capture and use human heuristics information. Two main qualities they obtained from the final solution: reduce the total time, and get more accurate results. They proposed the Heuristic-IoT framework for enhancing heuristic search algorithms to collect data from IoT sensors. They implemented their framework with the GA optimization algorithm using data collected from sensors to solve the Travelling Salesman Problem (TSP) with hidden edge costs.

The researcher in [9]focused on the area of CMfg by introduced a solution for composited CMfg Service Optimal Selection (CCSOS) to select the optimal one under multi-objective with four QoS (time, cost, availability, and reliability). They introduced the Hybrid Artificial Bee Colony (HABC) algorithm for (CCSOS) problems. They improved the onlooker strategy, by update great solutions through the chaos algorithm, which has the irregular property of all states and can help the worst bees to overcome.

Also, in[10], [11], and [12] Authors designed their algorithms based on Evolution Algorithms (EA), especially using a PSO evolutionary algorithm without any feedback or experience come from users in their solution. Liu et al., in 2013, [10] designed a cooperative EA based on integrating GA and CPSO algorithms for services composition and selection to solve *MOoPs*. They computed four non-functional attributes in their algorithm (cost, time, availability, and reliability).

Li et al. , 2013 [11] focused on the cloud logistics platform, which comes from IoT and Cloud computing environments. In this study author named a services selection approach as a constraint satisfaction problem (CSP). They aimed to find the best concrete WS by applied the rules of (Canfora et al., 2005), which defined the rules of the aggregation function. The four QoS (time, cost, availability, and reliability) took as QoS constraints. Then, they proposed a dynamic service selection model based PSO.

M. Elhosenya et al. [12] concerned with health services applications (HAS), they proposed a new model for Cloud-IoT based HSA to efficiently manage a large amount of data in an integrated industry 4.0 environment. Their proposed model aims to improve the performance of the healthcare systems by providing five factors: Reduce the execution time, waiting time, and turnaround time of medical requests tasks, Optimize the required storage of patient’s data efficiently, improve the scheduling tasks, provide a real-time data retrieval mechanism for healthcare applications, and Maximize utilization of resources. The authors proposed a new model to optimize virtual machine selection (VMs). They used three meta-heuristic optimization algorithms (Genetic Algorithm (GA), Particle swarm optimizer (PSO), and Parallel Particle swarm optimization (PPSO)) to build the proposed model.

In [13] and [14], authors keep the history's feedbacks come from users and regard it in their approaches. They proposed to solve the problem of services selection using an Artificial Neural Network Back-propagation Algorithm (ANN-BP). Nwe et al. [13] introduced a matching, ranking, and selecting a model to meet the distributed things on dynamic networks in IoT environments. They concerned with select the services based on two factors to optimize QoS information: *Objective Information* supported by the service providers and *Subjective Information* provided by the service consumers. To achieve the services selection, they proposed a Flexible QoS-Based

Service Selection Algorithm (FQSA). Their algorithm divided into two parts based on two previous factors: *Firstly*, to calculate the user subjective factors, they implemented a Similarity Aggregation Method (SAM)). *Secondly*, to find the objective factors FQSA algorithm, they performed an Artificial Neural Network Back-propagation Algorithm (ANN-BP) to improve a selection performance rate to be acceptable at real-time service selection. Also, they offered a flexible, user-friendly assessment form in a comfortable and friendly manner to allow users to request any number of QoS criteria for service. In [14], Mejri et al. regarding the scalability for services selection in IoT. They adopted a self-adaptive approach which combined two models: *QoS prediction model:* this model considers a user context, service context, and network context, by using the Artificial Neural Network (ANN). *The Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) model*: to introduce the best service to the consumer of the service. They optimized two QoS parameters: response time and reliability. Also, in recent researches, authors regard the feedback come from End-users but with fewer details of its calculation with Non-EA.

Authors in this paper[15] doing their study in the cloud manufacturing (CMfg) environment. They proposed their model based on composition design deployment, execution monitoring, and composition evaluation feedback. To analyze the optimization problem of service composition, they proposed the Gray Relational Analysis Method (GM). The QoS indexes of service composition in this paper divided into primary and secondary indexes. They studied six primary QoS indexes are: time, compos-ability, quality, usability, reliability, and cost. The weight of each index value in QoS evaluation is determined using an improved fuzzy comprehensive evaluation method. For service evaluation feedback: They pointed to evaluate the execution results by users. Then the QoS values updated according to user satisfaction and fed back the results to the cloud service library. For the next service call, the system will recompose services that had unsatisfactory results according to user requirements. In their paper, they did not mention any details of how to calculate the feedback.

Authors in [16] ranking the services in registry-based on user feedback to enhance a selection process. They proposed to use a social spider optimization algorithm for optimizing publishing operation. They calculated the fitness value based on a few factors. They considered response time, availability, and cost as metrics in the selection process. In their paper, researchers explain the steps to calculate the feedback and proposed algorithms, but they did not mention how to take the feedback from the End-user. Our approach regards the feedback by taking it directly from End-user in a relaxed and friendly manner. It ranked the services in the registry based on feedback information to be a principal value in the future selection process.

III. SERVICES SELECTION WORKFLOW

The processes of services selection workflow illustrated in Fig.[2]. End-user requests a service, which composes from more than one abstract service. Each abstract service contents from several concrete services with the same functionality, but different non-functional properties and QoS factors. The services composition is doing by composing those abstract services together. The selection of the ideal concrete service for each abstract service requires an effective selection algorithm that must be implemented to satisfy the different End-users requirements.

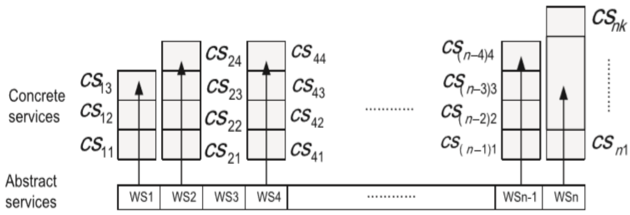

Fig .[2]: Services selection workflow[11]

Table1: Analysis of the existing state-of-the-art researches

Authors	Year	User preferences (feedback)	QoS Factors	Optimization Algorithm	User-friendliness
Anas et al. [8]	2016	X	▪ Reduce time ▪ Get accurate results	GA	X
Zhou et al. [9]	2016	X	▪ Time ▪ Cost ▪ Availability ▪ Reliability	Hybrid Artificial Bee Colony (HABC)	X
Liu et al.[10]	2013	X	▪ Cost ▪ Time	Integrating GA and CPSO	X

			▪ Availability ▪ Reliability		
Li et al. [11]	2013	X	▪ Cost ▪ Time ▪ Availability ▪ Reliability	Dynamic selection model based PSO	X
Elhoseny et al. [12]	2018	X	▪ Time. ▪ Storage space. ▪ Utilization of resources.	Meta-Heuristic algorithms ((GA), (PSO), and (PPSO))	X
Nwe et al. [13]	2014	✓	▪ Subjective Factors ▪ Objective Factors	FQSA by implement: SAM + ANN-BP	✓
Mejri et al. [14]	2017	✓	▪ Response time ▪ Reliability	QoS prediction model + TOPSIS model	X
Yuan et al. [15]	2019	✓	▪ Time ▪ Compos-ability ▪ Quality ▪ Usability ▪ Reliability ▪ Cost	Gray relational analysis Method(GM)	X
Divyad et al. [16]	2015	✓	▪ Response time ▪ Availability ▪ Cost	Social Spider Algorithm	X

IV. QOS IN IOT ENVIRONMENT

As mention in section two, the services requester need to consume the service from services provider by search in services registry. Also, in the services registry, there are many service providers with the same duty or functionality but with different QoS parameters or non-functional properties. (Robert Pirsig, 1974) Defined QoS as "Even though quality cannot be defined, you know what it is." Also, ITU-T E.860 defines QoS [17] as " The degree of conformance of the service delivered to a user by a provider with an agreement between them."

Our research follows the QoS Based on IoT Architecture introduced by (Ling Li. el. , 2014) in[18] to be appropriate with IoT architecture (Sensor layer, Network layer, and Application layer). The architecture integrated the traditional QoS attributes with other essential qualities in IoT Architecture (e.g., the cost of a network deployment, the information accuracy, energy consumption).

Mainly our research focused on QoS based on Application Layer. The application layer represents the highest layer in IoT architecture, which consists of many distributed services composed together to be one service to the End-user or an application. In this layer, the application is selected to establish a connection, and the End-user and the QoS constraints make the selection decisions of composing services.

In this architecture, the application layer aims to deal with ranking services based on users' requirements of QoS.

V. QUALITY OF SERVICE MODEL AND OBJECTIVE FUNCTIONS

The proposed approach studies five QoS parameters of complete service workflow (WF). The parameters or factors considered by two quality groups. *The first group* is the Business Quality Group (BQG). They are reputation $q_{RP}(s)$ and execution price $q_{EP}(s)$. *The second group* is the System Quality Group (SQG). It considers reliability $q_{RE}(s)$, availability $q_{AV}(s)$, and response time $q_{RT}(s)$.

The QoS parameters q(S) of a service S calculated as an aggregation of five factors:

$$\mathbf{q(S)=Min/Max\ [(q_{RP}(s),q_{EP}(s),q_{RE}(s),q_{AV}(s)q_{RT}(s))]} \quad \textbf{(1)}$$

There are five basic structures of WS to be composed (sequential, cycle, parallel, and branch)[10] [9]. Our study follows the sequential workflow of SC. The sequential workflow executes the composition in sequential order one follows to others, as in Fig.[2] The sequential workflow for five QoS parameters evaluated as aggregation functions[19] in Table2.

To calculate reputation q_{RP} (WF), response time q_{RT} (WF), and execution price q_{EP} (WF) of services WF, the average of them will calculate for each single service value. To calculate reliability q_{RE} (WF), and availability q_{AV} (WF), of services WF, the amount of them will calculate for every single service.

To optimize the QoS value, the behavior of factors is diverse from one element to another[20]. Some factors optimize when getting their maximum values; this called positive factors. The others optimized when getting their minimum values, and they called negative factors. Take our five parameters as an example, the optimal results for the cost, and response time are the smallest values. Moreover, the optimal results for reputation, reliability, and availability are the highest values.

There are two equations to calculate each of them. Positive factors evaluated as in equation (2):

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

Negative factors evaluated as in equation (3):

$$\bar{Q}i = \begin{cases} \dfrac{Qi^{max} - Qi}{Qi^{max} - Qi^{min}} & Qi^{max} - Qi^{min} \neq 0 \\ 1 & Qi^{max} - Qi^{min} = 0 \end{cases} \tag{3}$$

Where $\bar{Q}i \in [0.1]$

To get the optimal solution for all five parameters the objective function is:

Objective Function = [Max(q_{RE}), Max(q_{RP}), Max(q_{AV}), Min(q_{RT}), Min(q_{EP})] **(4)**

VI. LIKERT SCALE

It was named by Dr. Rensis Likert, in 1932. His goal was to improve a means of measuring psychological attitudes directly in a "scientific" way. It defined in [21]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."

Most commonly used as a 5-point scale or levels. The scales starting ranging from "Strongly Agree", "Agree", "Neither", " Disagree", and "Strongly Disagree". There is some researcher use of the seven and 9-point scales, which add additional levels of granularity. Other researchers use a 4-point, the point determination based on questionnaire requirements and deeps. Each scale assigned to codings like using alphabet value or a numeric value. This value used to measure the attitude under investigation, usually starting at one and incremented by one for each level, as in Fig.[3].

Fig.[3]: Sample scale used in Likert scale questions[22]

In this research Likert scale used to measure the agreement of service that taken from End-users after using the services. Scales help to rank services in the registry-based on End-user feedback and services reputation.

VII. PRACTICAL SWARM OPTIMIZATION ALGORITHM (PSO)

PSO is a nature-inspired MOoPs Algorithm developed by J. Kennedy and R. Eberhart in 1995[23]. PSO is a stochastic optimization technique, commonly used on continuous nonlinear optimization function[24] [25]. It mimics the behavior of a fish or birds swarm to search for food[10] in search space. The Swarm represents the population, and when an individual practical finds a direction for food in search space, it shares this information with other practices in a swarm. The other particles direct to the correct direction toward the food. The PSO directed by personal (individual) practical solution (xPBest), global practical solution (xGBest), and the present movement of the particles to decide their next positions in the search space. It means that if a particle finds a new solution, all the other particles will move closer to it, exploring the region more thoroughly in the process.

The improved-PSO (canonical PSO, CPSO) proposed by Shi and Eberhart in 1995 [10]. It hurries up the progress of traditional PSO in a dynamic environment. The Improved-

Table 2: Aggregation QoS Evaluation[19]

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

PSO has an inertia weight W, which balances between exploitation and exploration.

The basic equations of PSO are:

$$xi(t+1) = xi(t) + vij(t+1) \qquad (5)$$

$$v_{ij}(t+1) = w(t) \times v_{ij}(t)+c_1\times r_{1j}(t) \times (xPBest_{ij}(t)-x_{ij}(t)) + c_2\times r_{2j}(t) \times (xGBestj(t)-x_{ij}(t)) \qquad \textbf{(6)}$$

The iterations directed by two factors c1 is a variable to weigh the particle's knowledge, and c2 is a variable to weigh the swarm's knowledge, they control how far a particle will move in a single iteration, and two random numbers r1 and r2 generated between [0, 1]. In contrast, the present movement multiplied by an inertia factor w varying between $[w_{min},w_{max}]$ which used to weigh the last velocity, t a point in time or iteration number, xBest is the best solution the particle ever visited, and xGBest is the best location any particle in the swarm ever visited.

The initial population of size N and dimension j is denoted as $X = [X_1,X_2,...,X_N]^T$, where 'T' denotes the transfer operator. Each individual (particle) X_i ($i = 1,2,...,N$) is given as $X_i=[X_{i,1},X_{i,2},...,X_{i,j}]$ x represents a particle and i denotes the particle's number.

Also, the initial velocity of the population is denoted as $V = [V_1,V_2... V_N]^T$. Thus, the velocity of each particle X_i ($i = 1,2,...,N$) is given as $V_i=[V_{i,1},V_{i,2},...,V_{i,j}]$.

VIII. PROPOSED SERVICES SELECTION MODEL

In the normal process, a customer requested the service and selected it from the services registry. The registry content thousands of services from different providers with the same function but different in non-function properties.

In our solution, we proposed to rank services in a registry-based on its reputation.

The services classify on five Likert scale measurements (Strongly Agree, Agree, Neither, Disagree, and Strongly Disagree). It means the services with the same function classify in five parts with the five values (A, B, C, D, and F) sequentially based on its measurement value. This value reduces the search space when a customer requests a service; the selected will be from a part only meet requirement constraints.

For example, if a customer requests a services S with QoS q and value A, the search will be only in part A from the same services. This behavior improves performance, increases the response time, and reduces the cost.

For the first time, when a new customer requests a service, it selected from the registry in a usual way. Our research proposed to use an improved PSO algorithm discussed in section seven. This step is done with all new customers (customer's request service for the first time) only before they did any service evaluation. After a customer uses a service, they evaluate the QoS for each service based on a Likert scale. Our research depends on five QoS (Cost, Time, Reputation, Reliability, and Availability). This evaluation keeps the reputation of each service.

If there is more than one customer, use the same service from the same provider for the first time — the mean of reputation scales calculated and kept with a service.

The next time, when a customer requests a service, it will be selected based on its previous feedback from the registry.

IX. EXPERIMENT SETTINGS AND RESULTS

A. Experiment Settings and Dataset

This section discusses the experimental settings of our proposed approach to using the Likert scale measurement with improved PSO. The proposed approach is implemented using MATLAB to ensure its consistency. The evaluation of our approach in terms of feasibility and efficiency, an integer array-coding scheme designed where the number of items in the array denotes the dimension of our problem, and each element is an index of candidate service. The dataset generated randomly in a period between 0 and 1.

We compare the performance of regular PSO, Improved-PSO, and the proposed approach that uses a Likert scale with Improved-PSO to improve the performance of PSO by decrease the search space. The values of PSO's parameters are w= 1, c1= 1.50, and c2=2. The values of improved PSO's parameters are w= 0.729 and c1=c2= 1.49. In the experiments, the population size sets to 20. The experiments tests performed on a laptop with Windows 10, 2.90 GHz processor, and 8GB Ram, and the algorithms implemented in MATLAB.

B. Experimental Results

To analyses the efficiency of our approach, the execution time through the number of iterations of the three compared algorithms (PSO, Improved-PSO, and Likert Scale with Improved-PSO) calculated. The results show in Fig.[4] our proposed approach takes a few execution times than use Improved-PSO and PSO only. This result approves the efficiency of the proposed approach. Moreover, the results show that the gradual increase in the number of iterations increases the disparity of time between our approach, Improved-PSO, and PSO.

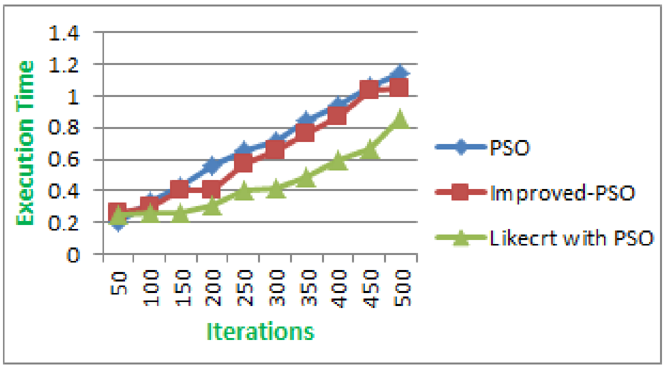

Fig.[4]: Evolution curves of Execution Time for PSO, Improved-PSO, and the proposed approach

In order to validate the feasibility of the proposed approach, we compare its optimizing results with PSO and Improved-PSO over the previous setting. The results in Fig.[5]shows, Likert Scale with PSO achieves better fitness values than compared algorithms.

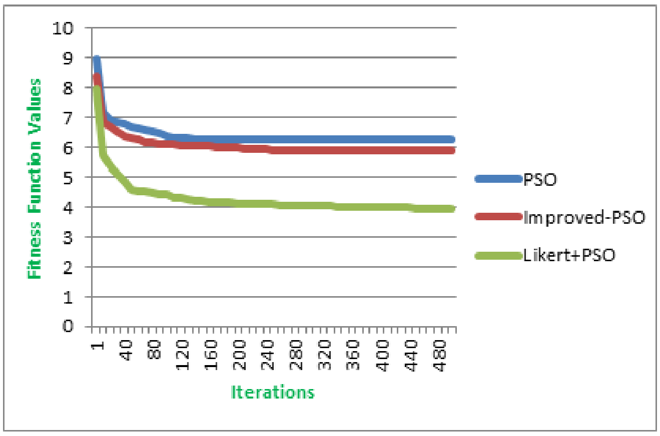

Fig.[5]: Evolution curves of Fitness Value for PSO, Improved-PSO, and the proposed approach.

X. CONCLUSION

With the increasing number of presented services available to the End-user in the IoT environment, that has similar functionality properties but different in non-functionality properties. The services selection problem becomes an NP-hard problem. This paper proposed a selection approach by using Likert scale measurement with Improved-Practical Swarm Optimization. In this approach, the services are ranking in the services registry based on its reputation. The End-user who uses the service evaluates it by using a user-friendly way. The approach regards the user preferences and their feedbacks as reputation information saved with each used service. This information improves the selection performance in the next selection and upgrades the reliability of the searching process by selecting a service wanted by the End-user. The comparisons found that the proposed approach has fewer execution times and excessive fitness value than PSO and Improved-PSO. The simulation results show that the efficiency of using the Likert Scale with PSO in services selection is much higher than using PSO only.

XI. RECOMMENDATION AND FUTURE WORK

For future work, we aim to test more QoS factors. Also, we plan to combine more than one meta-heuristic algorithm with regard the customer feedback.

REFERENCES

[1] W. Hugo Haas and M. (until J. 2002) Allen Brown, “Web Services Glossary,” *W3C Working Group*. [Online]. Available: https://www.w3.org/TR/ws-gloss/. [Accessed: 20-Jan-2018].
[2] F. Mustafa, “Dynamic Web Service Composition,” pp. 463–467, 2009.
[3] IBM® and I. K. Center, "Web services: Key roles.".
[4] Q. Z. Sheng, X. Qiao, A. V. Vasilakos, C. Szabo, S. Bourne, and X. Xu, “Web services composition: A decade’s overview,” *Inf. Sci. (Ny).*, vol. 280, pp. 218–238, 2014.
[5] A. Maaradji, “End-user service composition from a social networks analysis perspective,” 2012.
[6] S. Kenneth, M. Sevaux, and F. Glover, “A History of Metaheuristics ∗,” pp. 1–16.
[7] M. Yaghini, "What is a Metaheuristic?" 2009.
[8] M. Anas, S. Khatab, and A. Badr, “HeuristicIoT : A Framework for Augmenting Heuristic Search Algorithms

by Internet-of-Things Data," vol. 12, no. 1, pp. 133–141, 2016.
[9] 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.
[10] J. Liu *et al.*, "A Cooperative Evolution for QoS-driven IoT Service Composition," 2013.
[11] W. Li, Y. Zhong, X. Wang, and Y. Cao, "Resource virtualization and service selection cloud logistics," *J. Netw. Comput. Appl.*, pp. 1–9, 2013.
[12] 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.
[13] N. W. E. Nwe, H. Win, B. A. O. Jian-min, and C. U. I. Gang, "Flexible user-centric service selection algorithm for the Internet of Things services," *J. China Univ. Posts Telecommun.*, vol. 21, no. July, pp. 64–70, 2014.
[14] 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.
[15] M. Yuan, Z. Zhou, X. Cai, C. Sun, and W. Gu, "Service composition model and method in cloud manufacturing," *Robot. Comput. Integer. Manuf.*, vol. 61, no. July 2019, p. 101840, 2020.
[16] V. P. Divyad, P. Pazhanirajap, and V. Paul, "A NOVEL WEB SERVICE PUBLISHING MODEL BASED ON SOCIAL SPIDER OPTIMIZATION TECHNIQUE," pp. 373–387, 2015.
[17] ITU-T, "Recommendation E.860: Framework of a Service Level Agreement," *Ser. E Overall Netw. Oper. Teleph. Service, Serv. Oper. Hum. Factors*, 2002.
[18] L. Li, S. Li, and S. Zhao, "QoS-Aware Scheduling of Services-Oriented Internet of Things," vol. 10, no. 2, pp. 1497–1505, 2014.
[19] X. Mei, "QoS Aggregation Evaluation of Web Services Composition with Transaction," 2009.
[20] 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.
[21] D. Bertram, "Likert Scales. Topic Report," *Fac. Math. – Univ. Belgrade – Serbia*, 2009.
[22] H. N. Boone, D. A. Boone, and W. Virginia, "Analyzing Likert Data Likert-type Versus Likert Scales," *J. Ext.*, vol. 50, no. 2, 2012.
[23] M. N. Alam, "Particle Swarm Optimization : Algorithm and its Codes in MATLAB Particle Swarm Optimization : Algorithm and its Codes in MATLAB," no. March 2016.
[24] A. L. Ballardini, "A tutorial on Particle Swarm Optimization Clustering," 2016.
[25] S. A. Ludwig, "Applying Particle Swarm Optimization to Quality-of-Service-driven Web Service Composition."