Computers and Electrical Engineering 86 (2020) 106693

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Optimal resource selection framework for Internet-of-Things[☆]

Monika Bharti[a,*], Rajesh Kumar[b], Sharad Saxena[b], Himanshu Jindal[a]

[a] Computer Science and Engineering Department, Jaypee University of Information Technology, Waknaghat, Solan, Himachal Pradesh 173234, India

[b] Computer Science and Engineering Department, Thapar Institute of Engineering and Technology, Patiala, Punjab 147004, India

ARTICLE INFO

Article history:
Received 19 March 2019
Revised 30 April 2020
Accepted 1 May 2020
Available online xxx

Keywords:
Internet-of-Things
Discovery and selection
Optimization
Shared ontology
Fuzzy based rules

ABSTRACT

The fundamental requirement for communication and computation across distinct application areas on Internet-of-Things is the resource discovery that demands appropriate reasoning for the optimal selection. With exponential growth of resources and their produced huge amount of heterogeneous data, various activities with respect to foraging and sense-making loops face challenges due to interoperability. Hence, interoperability emerges as a major bottleneck for the requirement. Therefore, to eliminate the challenge, the paper has proposed an "Optimal Resource Selection Framework for Internet-of-Things" that deals with the interoperability and ease the resource discovery and selection. The framework facilitates formation of semantic knowledge base as Shared Virtual Composite Ontology for capturing dynamic *IoT* heterogeneous data. Moreover, it supports optimal resource selection through the proposed algorithms, *namely*, Resource discovery Algorithm and Improved Firefly Algorithm. Both algorithms target coordination and optimization with Shared Ontology, respectively. The feasibility of the framework is checked against data collected from Sutlej river, Ludhiana, Punjab, India. The proposed framework is evaluated using benchmark functions with respect to metrics such as mean, standard deviation, processing and execution time. The obtained results are compared with the existing Nature-Inspired algorithms to confirm the efficiency of the proposed framework.

1. Introduction

Internet-of-Things (*IoT*) envision a networked infrastructure through communication, computation and interaction among heterogeneous resources to sense, process and interpret *via* Internet-connected infrastructures. A resource is defined as an intelligent service that can be either software or hardware with fundamental characteristics such as physical embodiment, unique identifiers, offered service, location, processed information, operating system, languages and modes of communication. These resources as predicted by International Data Corporation (*IDC*) projects would grow to *212* billion by *2020* that would drive *40* zettabytes *IoT* data approximately. The data is not limited to sensors and machines but data from social networks, the web, and other user submitted physical observations and measurements. Such huge amount of data from real or virtual world will be available globally and in vast amount which is to be shared among applications without human intervention. Such a system has to be reactive, efficient, and effective as it will have to continuously respond to changes

☆ This paper is for regular issues of CAEE. Reviews processed and recommended for publication to the Editor-in-Chief by Associate Editor Dr. F. Picciali.

* Corresponding author.
E-mail addresses: monika.bharti@juit.ac.in, monika.bharti@thapar.edu (M. Bharti), rakumar@thapar.edu (R. Kumar), sharad.saxena@thapar.edu (S. Saxena), himanshu.jindal@juit.ac.in (H. Jindal).

https://doi.org/10.1016/j.compeleceng.2020.106693

in the user's situation. However, such a system can also be proactive, where its behavior is based on predicted situations evaluated with some level of confidence and probability. Therefore, there is a need to understand the resource discovery mechanisms on *IoT* platform [1].

The resource discovery mechanism enables users from different applications to access *IoT* data wherein users do not require to know the originating source, location, time, description of the data. It leads discovery mechanism to emerge as a challenging task since activities such as data acquisition, modeling, integration, assessment and reasoning on *IoT* varies with respect to data providers and end publishers or brokers. These activities help to facilitate data linking, knowledge representation and context-driven search and therefore, on their basis, resource discovery mechanism is broadly categorized in to two successive loops, *namely*, foraging and sense-making [2]. In former loop, originating sources are identified and assessed for knowledge extraction which is further formatted into consumable form. In latter, *i.e.*, sense-making loop, the extracted knowledge is analyzed, interpreted and exploited for the provision of service in accordance to a particular query. In addition to it, an implicit property across these activities of *IoT* is heterogeneity imposed by plethora of resources. It is due to diversity in resources with respect to communication, computational capabilities, representation, storage, search types and data formats. Such diversity leads to unmanaged big data driven by its velocity, variety, value, and volume which poses significant challenges to realize the vision of *IoT* [3]. Moreover, to ensure availability of the resources, data would require to be efficiently stored in widely distributed and heterogeneous information systems. It would lead to another challenge related to data retrieval from the information systems which is a non-trivial task without a common machine-readable data representation. Hence, interoperability is to be addressed to design efficient mechanisms for discovering available resources and capabilities.

To address the challenge, the first requirement is to have mechanism for knowledge representation that should (i) describe resources, their properties and capabilities (ii) define building blocks of the physical things such as configuration management, registration, un-registration, or idle resource and (iii) scope of discovery with respect to location, time and dimension. The second requirement is to interpret the formed knowledge and to access resources in accordance to the user's query. This in turn would be beneficial for the optimal selection of the resource among the discovered ones. These requirements, if not fulfilled against interoperability challenge to resource discovery mechanism, would act as a major bottleneck towards the vision of *IoT*.

In summary, to provide value-added services through *IoT* platforms, resources need to be discovered. For the purpose, it require (i) sophisticated techniques for managing meta-data, (ii) mechanism to discover meta-data and the resources, (iii) to automate resource management, and (iv) sophisticated techniques for resources' communication. As one possible solution, the paper has proposed an "Optimal Resource Selection Framework on Internet-of-Things (*ORSF-IoT*)" which provides an efficient and optimized way of selection of resource in minimal time. It uses semantic based description to represent huge amount of complex heterogeneous data as knowledge (termed as Shared Virtual Composite Ontology). The formed knowledge is interpreted through Fuzzy control rules which decompose the axioms of the ontology into sub-axioms according to a degree of match. Through fuzzy concept and roles, it results into set of discovered resources having similarity matches in accordance to the queries' parameters. However, the number of discovered resources would be directly proportional to degree of match between input parameters and knowledge. Therefore, to optimize the selection of resource, another technique, *i.e.*, an Improved Firefly Algorithm is proposed which selects the resource optimally in minimal time and has maximum context information against query.

The organization of the remaining paper is as follows. Section 2 provides a literature survey on the resource discovery and selection on *IoT*. Section 3 presents the proposed framework, *i.e., ORSF-IoT* and its components. The proposed scheme is evaluated on various parameters and its results are depicted in Section 4. Finally, the conclusion of the paper is summarized in Section 5.

2. Related work

On *IoT*, the approaches for resource discovery and selection are broadly categorized with respect to knowledge formation, representation, interpretation and optimization. For the same, some of the most comprehensive contributions are provided by the researchers in the field. Thus, the paper has provided a general, comprehensive and structured overview of existing techniques in order to understand the work done in the area of resource discovery and selection on *IoT*.

2.1. Resource discovery and interoperability

IoT envision integration among various disciplines such as healthcare, telecommunication, agriculture, semantic web, *etc.*, which raise interoperability as a key challenge to resource discovery due to the heterogeneity of the resources. Moreover, the volume, velocity and volatility of the data generated by highly distributed and heterogeneous resources add complexity to the interoperability challenge. This implies that providing interoperability among the interconnected resources is the prerequisite to support knowledge formation, knowledge representation, storage, and exchange. To address the challenge, various approaches for discovery mechanism with distinct requirements are needed to support *IoT* environment at both local and remote servers; in terms of location and network. As an example, in order to address various challenges of selecting sensors (where large numbers of sensors with overlapping are involved), a Context-Aware Sensor-search, Selection,and

Ranking Model (*CASSARAM*), is provided that works on the principle of user priorities [4]. Also, a framework to provide complete solution for resource discovery is formulated that provides storage of configured registries of resources, index them by grouping their location and provides lifetime attribute to remain discoverable [5]. The framework limits the discovery to authorized resources only and has limitations such that no proper syntax for describing resources, content ranking, does not provide interoperability while discovering of vehicular and smart resources. To reduce waiting time and energy consumption, a neighbor discovery approach is suggested that operate nodes asynchronously having low energy. It lacks in providing discovery for large data of resources and does not handle collisions during data transmission [6]. For capturing *IoT* data, a heuristic framework has been developed [7]. The gathered data further undergoes transformation and filtering for efficient search using genetic algorithm. The framework has its limitations that it has not focused on gathering and managing heterogeneous data on *IoT*.

To its improvement, a crawler is designed to collect *IoT* data automatically from different data sources [8]. The crawler provides interface for both human users and machines. However, only those data sources are chosen where the sensor data is represented through a map. Few researchers have designed an adaptive discovery algorithm, "Speed and Time based Energy Efficient Probing (*STEEP*)", which helps in studying the impact on discovering resources, when nodes turn off the radio interface to conserve power [9]. The suggested approach helps in conserving energy by *30%* for discovery in delay tolerant networks. The approach has its limitation for frequent turning off radio interfaces which may in turn loses the stored data during transmission to surface station. Some authors have presented an *IoT* based healthcare system for cancer care services that uses business analytics/cloud services for actionable insights, decision making, data transmission and reporting for enhancing cancer treatments [10]. It provides complete healthcare solution for cancer patients to help increase the quality of life. Few authors have suggested *IoT* based smart home management system that offers interoperability with exceptionally reliable connections [11]. The system is user-specific to control the home gadgets and adding up other security features. Its has its disadvantages to turn-off the gadgets when not in use, making it to be an inefficient system.

2.2. Knowledge formation and representation

Though, the researchers have addressed various issues to interoperability challenge but has not incorporated challenges such as syntax, waiting time, energy consumption collision detection on *IoT* at various levels like semantic, radio access and context with respect to resource discovery. Further, to discover the resources continuously and tracing its localization position in a network, there is a need to metadata and semantic tagging of information. To address the concern, a mechanism, *i.e.* Ontology is generally referred that defines vocabulary for the data with its meaning. Ontology is the meta-knowledge that describes everything known about problem domain. It allows interoperability for the resource' meta-data, enables automatic configuration and management which together results into generation of actionable intelligence and enabling resources' interaction [12]. For example, the paper has suggested Internet-of-Things Directory System *(IoT-DS)* that performs the semantic description, discovery, and integration of objects. The system has a limitation that it does not handle the exponential growth of the objects on the *IoT* [13]. A resource discovery algorithm based on preference and movement pattern similarity in disconnected and delay-tolerant Social Internet of Things (*SIoT*) has been postulated [14]. The algorithm implements a three dimensional cartesian coordinate system with the aim of enhancing the search efficiency over the *SIoT*. It is based on both preference and movement pattern similarity to achieve higher search efficiency and to reduce the system overheads of *SIoT*. It lacks in modeling effective behavior prediction model to reduce the wait time of the nodes in mobile *SIoT*. To provide privacy to the data over the Internet, D. Hussein *et al.* have formulated a novel service framework based on a cognitive reasoning approach for dynamic *SIoT* service discovery in smart spaces [15]. The reasoning about users' situational needs, preferences, and other social aspects along with environment is suggested for generating a list of situation-aware services which matches users' needs. This reasoning is implemented as a proof-of-concept prototype, *namely*, Airport Dynamic Social, within a smart airport. An empirical study to evaluate the reasoning shows the improved services and efficient adaptability to situational needs.

To model Ontology, the major challenges are due to heterogeneity, multi-modality and volume of data. Moreover, semantic descriptions alone do not provide semantic interoperability and will not resolve all the issues regarding discovery, management of data, and supporting autonomous interactions. The semantic description still needs to be shared, processed, and interpreted by various methods and services across different domains. To eliminate the challenges, techniques are required which use the meaning and information about the context of request to semantically match it with the meaning of the offered services. Here, service requesters and providers utilize ontology to discover similarity between two concepts or services and determine semantic distance between concepts. For example, a Regularized Newton Method (*RNM*) has been postulated without line search [16]. The method controls a regularization parameter instead of a step size in order to guarantee the global convergence. It is shown that by using the suggested method, the tightness of the global complexity bounds are easily constructed which solves subproblem inexactly. A fuzzy consensus model to destroy redundancy of discordant information related to the same phenomenon provided by dynamic nodes. The model limits the search and selection of nodes into restricted area as local only [17]. To its improvement, another approach based on fuzzy set is suggested for selection criteria considering parameters for nodes, *namely*, connectivity degree, link quality, and the distance in terms of hop. It works on centralized approach of Software Defined Network *(SDN)* and provides local controller selection. Another approach, *i.e.*, Artificial Potential Fields (*APFs*) is presented that is based on the search and selection of the decentralized service approach

[18]. It has its drawback that it does not consider multiple variables in generating artificial potential fields, such as hop count.

2.3. Knowledge interpretation for optimal resource selection

Since, the domain for searching on *IoT* landscape is categorized in to (i) Searching around me, (ii) Searching on My network, (iii) Searching on directories, and (iv) Accessing Thing Meta-data on the basis of interaction patterns. These patterns help application developers in decision making and would decrease standards' dependency. It removes various constraints for discovery like bootstrapping, search, range, ranking and rich queries [19]. Few authors have reviewed various search techniques, categorized on the basis of complex nature of *IoT* and data produced by resources as fundamental search principles, data/knowledge representation and contents being searched [20]. They have suggested that these classifications will benefit the researchers to deeply understand searching methods on *IoT*. The suggested approaches of discovery have their major disadvantages of dependency on application or system for connecting intelligent resources dynamically and impose restriction to process an optimal decision. To resolve this dependency for better communication through the selection of a rightful resource, various approaches have been suggested. As an example, for selecting multiple nodes as autonomous nodes, a co-operative decision making mechanism is presented that increase network's performance to meet Quality-of-Service (*QoS*) [21]. It limits decision making with finite set of nodes and performance is decreased if nodes are added dynamically to the system during the process. For providing services to remote nodes and to dynamically add new nodes with updated information, distributed consensus decision making is introduced. It minimizes the dependency on matching value and uses clustering approach for decision [22]. K. H. N. Bui *et al*. have introduced a new approach for smart traffic light control at intersection [23]. The researchers have suggested a connected intersection system where every objects such as vehicles, sensors, and traffic lights will be connected and sharing information to one another. By this way, the controller is able to collect effectively and mobility traffic flow at intersection in real-time. The algorithm has its own disadvantages that it takes into account the priority of vehicles which are at same level of emergency. To consider priority at different level, A. Garcia-de Prado *et al.* have postulated a COLLaborative ConText aware service oriented architecture (*COLLECT*), which facilitates both the integration of *IoT* heterogeneous domain context data through the use of a light message broker, easy data delivery among several agents and collaborative participants in the system, and making use of an enterprise service bus [24]. The approach helps to avoid additional resource consumption to edge devices and saving costs in cloud hosting. The architecture is not feasible for real time prediction using contextual information to improve intelligent decision making in the domain. Few authors have suggested self-selection decision tree based on hand-off probability distribution that select nodes efficiently by generating decision feedback. It has a drawback of its dependency on user's input. Also, there is need of automated optimal decision making that select nodes to provide maximum information [25]. A novel Lifetime Maximizing optimal Clustering Algorithm (*LiMCA*) for battery-powered *IoT* devices is suggested [26]. The algorithm helps in analyzing the maximum lifetime of network and the requirements for maximizing the lifetime. It uses stochastic deployment scheme for nodes acting as cluster heads and members related to one cluster. The algorithm has its drawback as this is not tested for mobility of cluster head. Moreover, being connected for *IoT* devices, the algorithm is tested for *200* nodes.

Considering both requirements, another paper has been suggested by the authors [27]. In this paper, the fundamental challenge of resource discovery on *IoT* is addressed through framework, *namely*, “Intelligent Resource Inquisition Framework on Internet-of-Things (*IRIF-IoT*)”. Its main features are to link resources that are linked through shared ontology as conceptual sets on large scale *via* semantic description and are authenticated at centralized server that handles databases collected from distinct local servers. Here, resources are discovered with “Semantic Matchmaking Engine using Bipartite Graph (*SMEBG*)” that emphasizes on semantics for knowledge representation and automated reasoning. The technique performs semantic matchmaking which is further optimized with Hungarian approach having strong polynomial time bound complexity of $O(|V_x|^3)$. Though, *SMEBG* performs efficient searching with minimal operational complexity and enhances system performance significantly but it has not accounted for various constraints to the resources. The constraints are availability of the resources, waiting time for idle resources, queuing length, lag time, priority based scheduling, cost and frequency of completed cycles. Moreover, it has evaluated performance of the *SMEBG* with respect to searching time and evaluation parameters of the network such as packet delivery rate, throughput, packet loss, latency and power consumption. Also, it has compared the searching time of *SMEBG* with Fuzzy Control Logic (*FCL*) and Genetic Algorithm (*GA*) against dataset of *100* resources collected from Ladowal Toll Plaza, Ludhiana, Punjab, India. For the same, it has used software, *namely*, ‘*AIMSUN*’, that provides environment for Traffic Network Editor (*TEDI*).

In the present paper, the challenge of interoperability to resource discovery and selection on *IoT* is addressed through *ORSF-IoT* that performs knowledge representation through semantic networks as ontology. The ontology has inbuilt rules that are described using predicate logic. The framework does knowledge interpretation *via* semantic matchmaking using the concept of Fuzzy Control Logic (*FCL*). It addresses various constraints to the resources using *FCL* and optimal resource is selected through proposed Improved Firefly Algorithm *(IMPFF)*. The performance of *IMPFF* is evaluated for benchmark functions against Nature Inspired Algorithms, *i.e., PSO, ABC, CA* with respect to metrics such as *Mean, Standard Deviation, Processing Time* and *Execution Time*. The evaluation is tested on collected data sets of physical parameters such as Electrical Conductivity, *pH*, Temperature, Chloride and Dissolved oxygen of Sutlej river, Ludhiana, Punjab, India. The data values of physical parameters are considered as fireflies and process is run for *30* times, each with *1000* iterations on *MATLAB* for

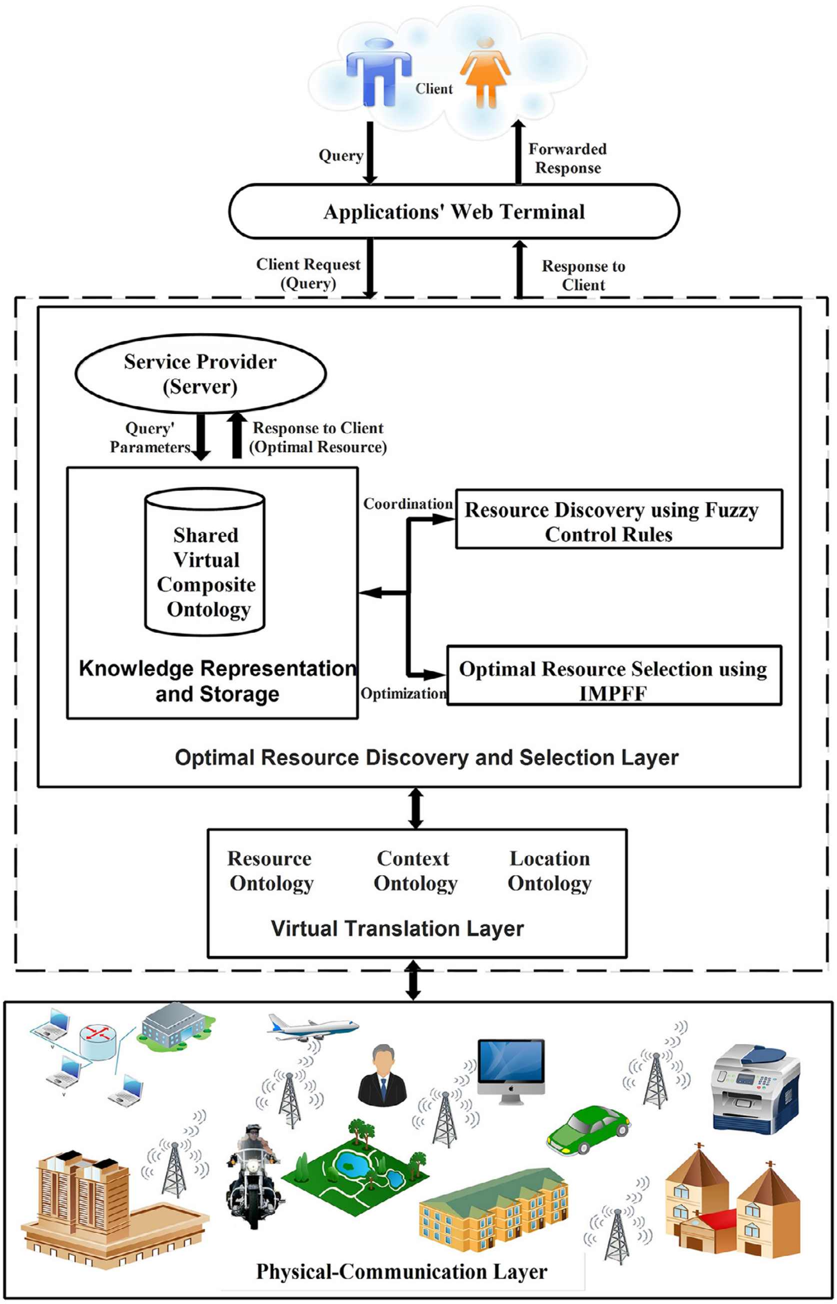

Fig. 1. REST-CoRE based Framework for Optimal Resource Selection on Internet-of-Things.

computation. The test is run to detect water's quality deterioration and automating monitoring stations by sending timely alarms for preventive measures.

3. Proposed optimal resource selection framework for Internet-of-Things

The proposed "Optimal Resource Selection Framework for Internet-of-Things (*ORSF-IoT*)" presented in Fig. 1 includes four layers, *namely*, (i) Physical-Communication, (ii) Virtual Translation, (iii) Optimal Resource Discovery and (iv) Application Web Terminal.

The first and second layer collects data from heterogeneous resources using platform of *CoRE* and translates collected data into useful information as different ontologies, respectively. The third layer is divided into three sub-layers: knowledge representation and storage, resource discovery and optimal resource selection. Here, knowledge representation is done

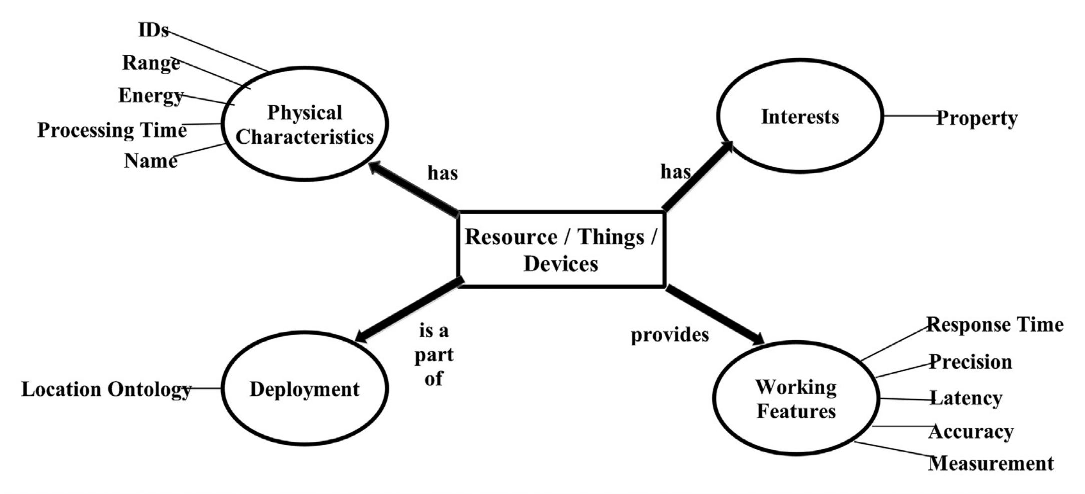

Fig. 2. Resource Ontology.

through the concept of ontology that frames the relations among the resources and concepts using defined properties, logical notations and produces knowledge. The formed knowledge is interpreted *via FCL* that outputs redundant-free data and time constraints and helps in discovering resources. Finally, discovered resources undergoes the process of optimization *via IMPFF* that helps in selecting a particular resource against query. The fourth layer provides a platform for clients to put their query as well as server to respond back to client. Each layer is briefed below.

3.1. Physical communication layer

This layer is responsible for collecting data from the heterogeneous resources such as Smart Homes, Intelligent Transportation Systems, *etc.* through the Constrained RESTful Environments (*CoRE*) [28]. *CoRE* is based on Representational State Transfer (*REST*) architecture having 8-bit constrained nodes with limited memory. It supports Low-Power Wireless Personal Area Networks (*6LoWPANs*) and is suitable for interactions among Machine-to-Machine (*M2M*) applications. *CoAP* is a specialized web transfer protocol for use with constrained nodes and constrained (e.g., low-power, lossy) networks. It is designed to easily interface with *HTTP* for integration with the Web while meeting specialized requirements such as multicast support, very low overhead and simplicity for constrained environments. *CoAP* logically uses a two-layer approach, a *CoAP* messaging layer that deals with *UDP*, the asynchronous nature of the interactions, and the request/response interactions layer that uses method and response codes. As, *CoAP* run over *UDP*, it supports the use of multicast *IP* destination addresses, enabling multicast *CoAP* requests. The resource discovery can be performed either unicast or multicast. When a server's *IP* address is already known, either a priori or resolved *via* the Domain Name System *(DNS)*, unicast discovery is performed in order to locate the entry point to the resource of interest. This is performed using a GET to "/.well-known/core" on the server, which returns a payload in the *CoAP* Link Format. A user would then match the appropriate Resource Type, Interface Description, and possible media type for its application. These attributes may also be included in the query string in order to filter the number of links returned in a response. Multicast Resource Discovery is useful when a user needs to locate a resource within a limited scope, and that scope supports IP multicast. A GET request to the appropriate multicast address is made for "/.well-known/core". In order to limit the number and size of responses, a query string is recommended with the known attributes. Typically, a resource would be discovered based on its Resource Type and/or Interface Description, along with possible application-specific attributes. In *REST* architecture, resources are registered in resource directory using *CoRE* Link Format. Also, the registered resources are provided a link in lookup interface table of resource directory, upon which a user can request for a resource *via* centralized knowledge repository. The gathered data at this layer is transmitted to Virtual Translation Layer (*VTL*) wherein data is processed into valuable information.

3.2. Virtual translation layer

VTL performs the knowledge representation and storage on the gathered data through *CoRE*. For the same, it performs semantic modeling based on standard design principles such as lightweight, completeness, compatibility and modularity, in order to hide the heterogeneity of real world entities, *i.e.*, resources. This semantic modeling processes data into valuable information as different ontologies, *i.e.,* Resource, Location and Context using CoRE Link format. These ontologies are discussed.

The Resource Ontology describes physical or composite object along with their characteristics. Fig. 2 is an example of resource ontology having four subclasses, *i.e.*, physical characteristics, interests, deployment and working features. Some of the

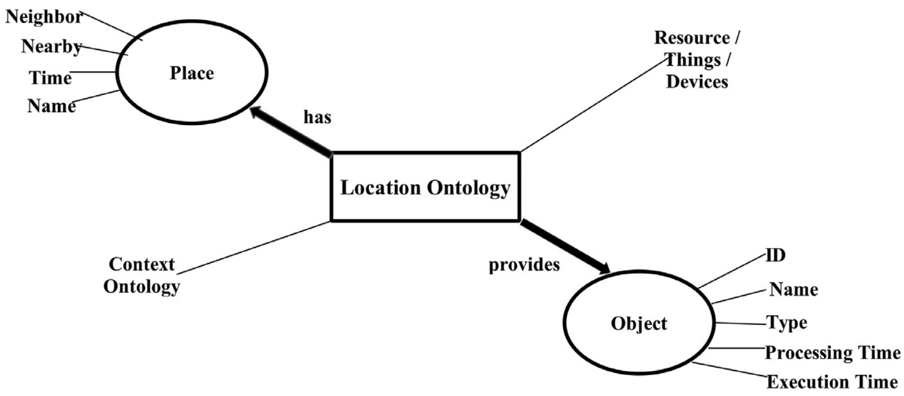

Fig. 3. Location Ontology.

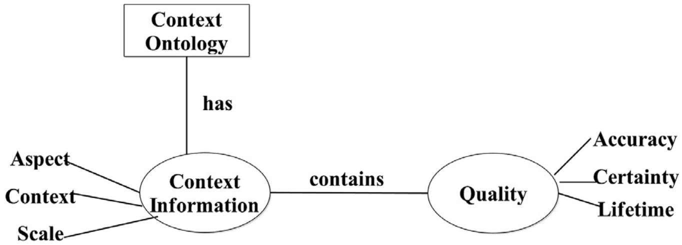

Fig. 4. Context Ontology.

properties that are exposed using this ontology are resource *ID*, range, energy, processing time, name, response time, precision, latency, accuracy, and measurement. These properties share common information with respect to Internet-connected resources. The deployment subclass has location ontology which is described further.

The Location Ontology adds geospatial information semantically for linking *IoT* resources and services. Location ontology has three sub-classes, *viz.*, place, object and context ontology. The sub-class place has two major components, *i.e.*, neighbor and nearby. In former, it describes neighbors of resources. In latter, it defines nearby locations to location class. The ontology shown in Fig. 3 exposes properties like place-time, place-name, object-id, object-name, object-type, processing and execution time of object. Location ontology is used to perform reverse geocoding, to identify a place using latitude, longitude and *vice versa*, using geospatial semantic contextual information. The location ontology has subclass as context ontology, which is described next.

Context Ontology represents contextual information that enables awareness and interoperability at the time of service discovery and composition. Contextual information is obtained using direct or indirect methods. In former, the user's gathered knowledge and sensed information is used to describe contextual information. In latter, the inference mechanisms and reasoning helps to obtain the information. Contextual information can also be described from obtained sensor data. In short, it is described as any information which helps to distinguish states of entity related to specific scenario (aspect). The scenario is based on value-range that can have reachable states in all dimensions (scale). Each scenario aggregates one or two dimensions to provide interrelated contextual information (one-to-one mapping, one-to-many mapping). As *IoT* resources are constrained specific and contextual information obtained from sensor data might not be accurate due to faultiness of the sensors. Taking into consideration, context ontology considers quality constraints like accuracy, certainty and lifetime. Fig. 4 describes the context ontology of such kind used for *IoT* resources. The gathered information from *VTL* as ontologies is transferred to Optimal Resource Discovery and Selection Layer.

3.3. Optimal resource discovery and selection layer

This layer is subdivided into three stages, *i.e.*, Knowledge Representation and Storage, Resource Discovery using Fuzzy control rules and Optimal Resource Selection using Improved Firefly Algorithm. Each stage is discussed in detail below.

3.3.1. Knowledge representation and storage

The information sent from *VTL via* Web Links is gathered and stored in centralized server repository (constrained specific). Web Links provides a link in lookup interface table of resource directory, upon which a user can request for a resource *via* centralized repository. The gathered information from different ontologies at *VTL* creates a centralized Shared Virtual Composite Ontology by linking them. This Shared Virtual Composite Ontology is generic and can be applied to any *IoT* applications, *e.g.*, Home Automation, Intelligent Transportation System, Smart Agriculture, *etc.*

Table 1
Definition for $\chi-$ Relation .

$\chi-$ Relation	Description
Segment	$r_c \subseteq r'_c$ if $pqs(r_c) = pqs(r'_c)$
Weak	r_c *(weak relation)* r'_c if $cts(r_c) \subseteq cts(r'_c)$
Strong	r_c *(strong relation)* r'_c if $pqs(r_c) = pqs(r'_c)$ and $cts(r_c) = cts(r'_c)$

Table 2
Logical relation between concepts .

Condition	*Notation*
$s\ H_{C_x}\ t$	$s \mapsto t$
$s\ C_r\ t$	$s,\ t \mapsto$
$s\ I_r\ t$	$\mapsto s,\ t$

Table 3
Properties for Ontology Morphism .

Criteria	*New Relation*
$s\ H_{C_x}\ t$	$K(s)\ H'_{C_x}\ K(t)$
$s\ C_r\ t$	$K(s)\ C'_r\ K(t)$
$s\ I_r\ t$	$K(s)\ I'_r\ K(t)$
$\varrho(r_c) = (s, t)$	$\varrho'(L(r_c)) = (K(s), K(t))$

It performs knowledge representation through Mapping Process that deduce logical relations between concepts to link the ontologies and generate knowledge. Before initiating the mapping process, it is required to describe gathered data from the resources with few definitions and notations that helps to transform it into information.

(A) **Definition for $\chi-$Relation:** A $\chi-$ relation describes the semantic properties of the concepts and is defined as $Relation : I_x \longrightarrow I'_x$, where I_x and I'_x are instances for various concepts. Considering r_c and r'_c to be any element of I_x and I'_x, the relation could be one of the following types as shown in Table 1. where, $pqs(r)$ and $cts(r)$ represents prerequisite and context sets of relation r, respectively.

(B) **Definition for Ontology:** An ontology, O_k is expressed mathematically as:

$$O_k = \{R_x, C_x, H_{C_x}, \varrho, C_r, I_r\} \tag{1}$$

where, R_x and C_x are the *sets that identify relation and concept*, respectively, H_{C_x} represents *partial ordered pair on* C_x and is known as *concept hierarchy taxonomy*, $\varrho : R_x \longrightarrow C_x \times C_x$ defines *signature of relation among two concepts,* C_r tells *relation associated to concepts and is called divergent, i.e., no relation between associated resources* and I_r represents *relation associated to concepts and is called convergent, i.e., at least a single relation exists between associated resources.*

(C) **Logical relation between concepts:** The following logical notations define relation between concepts, *say,* s and t, is expressed as follows (*see* Table 2):

(D) **Degree of Convergence:** It measures the convergence among the concepts and is given as:

$$D_{c_{s,t}} = \frac{\text{number of common instances between s and t}}{min(\mid s \mid, \mid t \mid)} \tag{2}$$

(E) **Definition for Ontology Morphism:** If the comparison of images for two concepts in two distinct ontologies is equivalent to each other, then, ontology morphism between two ontologies, $O_k = \left(R_x, C_x, H_{C_x}, \varrho, C_r, I_r\right)$ and $O'_k = \left(R'_x, C'_x, H'_{C_x}, \varrho', C'_r, I'_r\right)$ is the couple of function *(K,L)* such that $K : C_x \longrightarrow C'_x$ and $L : R_x \longrightarrow R'_x$. Suppose, s and t are two elements of C_x then relations generated are shown in Table 3.

With these definitions and notations, two ontologies, *say,* O_k and O'_k are mapped into one another, *i.e.*, each entity concept, s in O_k has same intended corresponding concept, t in O'_k. The mapping process undergoes phases, *namely*, similarity computation, similarity generation, interpretation and benefits in generating information by comparing instances and produces relations among them, *i.e.*, convergent and divergent. These phases are discussed.

(A) **Similarity computation:** Similarity computation, SM_c, is an iterative process that provides similarity among concepts. Also, it uses linguistic tools for comparing the concepts. It produces similarities in (*i-1*) iterations and inference rules are applied for comparisons. To find similarities among ontologies, different similarity measures or methods are required that are discussed below.

- **WordNet**: WordNet is used to compute the uniform distance among edges, *i.e.*, to describe relations among two concepts [29]. It also helps in judging the similarity among concepts as information content by comparing either syn/antonym or hol/hyponym synset (synonym set) information stored in WordNet. Since, hol/hyponym relations are same as synonyms, then all concepts of synset share same information and path among concepts is either single

Table 4
Degree of Match .

Criteria	Match
$c_{x_{out}}$, $d_{m_{out}}$ equivalent to $q_{s_{out}}$	Correct
$c_{x_{out}}$, $d_{m_{out}}$ SuperClass of $q_{s_{out}}$	Correct
$c_{x_{out}}$, $d_{m_{out}}$ Subsumes $q_{s_{out}}$	Segment
$c_{x_{out}}$, $d_{m_{out}}$ Subclass $q_{s_{out}}$	Contain In
None of the above	Failure

entity (*i.e., 1*), or hyponym link (*i.e.*, any value between *0* or *1*) or no relation (*i.e., 0*). Based on methodology for WordNet, the relations for two concepts, say M and N are described below.

- $M \equiv N \Rightarrow sim(M, N) = 1$. It exists if meaning of M has synonym to meaning of N.
- $M \supseteq N$ $i.e. sim(M, N) = 0.7$. It exists if meaning of M has hyponym to meaning of N.
- $M \subseteq N$ $i.e. sim(M, N) = 0.3$. It exists if meaning of N has hyponym to meaning of M.
- $M \perp N \Rightarrow sim(M, N) = 0$. It exists if there is no relation among M and N.

WordNet provides the similarity relation among concepts but most words have multiple senses. As WordNet has defined conceptual similarity, so using it, the hypnym values are assumed and described as *0.7* and *0.3* for concepts.

- **String Equability:** It performs comparisons of two or more strings. Its similarity measure is given as:

$$Str_{sim}(s, t) = \begin{cases} 1, & \text{if } s.char(i) = t.char(i) \forall i \in [0, \mid s \mid] with \mid s \mid = \mid t \mid \\ 0, & otherwise \end{cases} \tag{3}$$

- **String Equality:** It measures similarity computation among two strings on scale of 0 to 1 and is given as.

$$Str_{sim} = max\left(0, \frac{min(s, t) - euc_{dis}(s, t)}{min(s, t)}\right) \tag{4}$$

where, euc_{dis} is euclidean distance among two concepts.

(B) **Similarity Generation:** It uses a Rule Base to compute similarities between ontologies and has sets of rules. These rules benefit in the judgment on whether two concepts are similar but none provides the mapping for itself. Also, they give only a similarity weight between two compared entities. A threshold is defined on the similarity values to determine the correspondence or the non-correspondence. Moreover, these rule using the definition for ontology morphism, would generate new similarity relations at iteration *i* from the *i-1*. For the progressive and dynamic similarity generation, consider a similarity function, $F_{nc} : C_x \times C'_x \longrightarrow [0, 1]$, which associate for each couple of concept, a degree of similarity comprise between *0* and *1*. The following rules illustrate the mechanism of similarity computation as follows:

Rule 1 IF $F_{nc}(s, s')$ increases value THEN $\forall (t, t') \in C_x, C'_x$ such that $(s \mapsto t$ and $s' \mapsto t') : F_{nc}(t, t') = F_{nc}(t, t') + \left(\frac{F_{nc}(s,s')}{nb_c(s)}\right)$

Rule 2 IF $F_{nc}(s, s')$ increases value THEN $\forall t' \in C'_x$ such that $\mapsto s', t' : F_{nc}(s, t') = F_{nc}(s, t') + \left(F_{nc}(s, s') \times R_{x_{s',t'}}\right)$ and $\forall t \in C_x$ such that $\mapsto s, t : F_{nc}(s', t) = F_{nc}(s', t) + \left(F_{nc}(s, s') \times R_{x_{s,t}}\right)$

Here, $nb_c(s)$ as sub-concept of C_x.

(C) **Interpretation:** The mapping values derived from similarity computation and generation are assigned on the basis of semantic matching of the concepts. The query as the input from the client and the output is a Matching Set, M_n, according to the Degree of Match, D_{mt}. The algorithm iterates in its repository in order to determine a match between input and output. The matching of output is defined as:

$$\begin{aligned} &\forall C_x \epsilon Q_{s_{out}}, \exists D_{mt} \epsilon C_x \text{such that} \\ &\quad match(C_x, D_{mt}) \neq \text{Failure} \end{aligned} \tag{5}$$

Let $Q_{user_{in}}$, M_n be the list of input concepts, then matching of inputs is calculated by:

$$\begin{aligned} &\forall C_x \epsilon M_{n_{in}}, \exists D_{mt} \epsilon Q_{user_{in}} \text{such that} \\ &\qquad match(C_x, D_{mt}) \neq \text{Failure} \end{aligned} \tag{6}$$

Thus, the match, C_x and D_{mt} returns the degree of match between two concepts. For conceptual sets, $q_{s_{out}} \epsilon Q_{s_{out}}$, $c_{x_{out}} \epsilon C_x$ and $d_{m_{out}} \epsilon D_{mt}$, the match function is described in Table 4.
D_{mt} is ranked as Correct $<$ Segment $<$ ContainIn $<$ Failure.

For a machine to machine interaction, there is no intervention of humans and data override is not possible at the time of resource discovery by a constrained server. Moreover, *IoT* is an abstraction of resources as services, they are described through information provided by resource constrained servers having limited computational capacity, less bandwidth, limited memory and energy consumption. These services/resources are available to clients *via* multicast Web Resource Discovery through Shared Virtual Composite Ontology. This ontology provides homologous and scalable means of accessing *IoT*

Table 5
Fuzzy Control Rules .

Condition	*Match*
$Q_{user_{in}}$: if $S_{mat} == SM_c$	$Q_{user_{out}}$: 1 (True)
$Q_{user_{in}}$: if $S_{mat} <= SM_c$	$Q_{user_{out}}$: 1 (True)
$Q_{user_{in}}$: if $S_{mat} < SM_c$	$Q_{user_{out}}$: 0/1
$Q_{user_{in}}$: if $S_{mat} >= SM_c$	$Q_{user_{out}}$: 1/0
$Q_{user_{in}}$: if $S_{mat} > SM_c$	$Q_{user_{out}}$: 0/1
$Q_{user_{in}}$: if $S_{mat} \neq SM_c$	$Q_{user_{out}}$: 0

resources coupled with service oriented computing. It makes use of semantic modeling for describing the relations of *IoT* resources, their attributes, properties, *etc*. These services or resources are further processed for resource discovery using Fuzzy control rules as described below.

3.3.2. Resource discovery

In this subsection, *FCL* is used for resource discovery through the network. It takes input query as matched set, S_{mat}, to retrieve large set of data from Shared Virtual Composite Ontology. The input S_{mat} is processed using control rules under fuzzy operation to generate fuzzy value on the basis of SM_c as shown in Table 5. The output is calculated as total search time for processing the SM_c. Finally, the accurate value is generated by fuzzy estimation. The operation is processed for all possible inputs to avoid redundancy in SM_c. Also, S_{mat} have resources that are already allocated to multiple activities over the server. Moreover, these resources have complex logical dependencies and stochastic duration. This leads to selection of resources with increase in idle time of other resources to come in queue as similarity set. Further, these resources upon completion of one activity are released to be allocated to other process on sever. Thus, the resources in S_{mat} face certain constraints with respect to time dependency such as available resource, waiting time, queuing length, lag time, duration factor and completion of cycle, while computing the degree of match.

In order to eliminate these constraints, resources in S_{mat} need to be allocated considering multiple criteria as objectives, O_j. The O_j are (i) giving priority to resources having larger quantity ratio with respect to quantity of required parameters; (ii) minimizing waiting time for idle resources; (iii) minimizing queuing length for idle resources; (iv) reducing lag time among current and immediate processes; (v) providing priority for critical activities; and (vi) balancing frequency of completed cycles. Here, first objective satisfies activities having less resources for next operation cycle and reflects the various utilization of different resources. Second objective minimizes the waiting time and cost for incurred resources. Third avoids excessive queue of idle resources. Few activities are started upon completion of preceding activities. Fourth avoids lag time in order to increase efficiency of process. Fifth weighs longer and heavy processes as these are critical to process. Last objective balances the work by rearranging the frequency of completed cycles.

Now, considering the above discussed O_j, resources in S_{mat} are expressed as fuzzy sets that satisfy l objectives: $Oj_i(i = 1, ., l)$. Also, it defines the associated fuzzy membership $\vartheta_l(Oj_i)$ that indicates how the Oj_i are satisfied. The ϑ_l helps to resolve the time dependency constraints and takes less computation time for processing S_{mat}. The intersection of constraint free S_{mat} would help in taking decision, D_F for degree of match and is given as:

$$D_F = \vartheta_1 \bigcap \vartheta_2 \bigcap \ldots . \bigcap \vartheta_l \tag{7}$$

The membership function for combined decision, ϑ_{D_F} is expressed as:

$$\vartheta_{D_F} = \sum_{i=1}^{l} \beta_i \times \vartheta_i \tag{8}$$

where, $\sum_{i=1}^{l} \beta_i = 1$. β_i reflects the relative importance of $\vartheta_1, \ldots, \vartheta_l$. If the weights *(wt)* are not normalized for ϑ_ls, then it is obtained using,

$$\beta_i = \frac{wt_i}{\sum_{p=1}^{l} wt_p} \tag{9}$$

The D_F selects the ϑ_l with largest membership value after determining membership of ϑ_{D_F} and is given by:

$$\vartheta_{FinalD_F} = \{S_{mat} \mid \vartheta_{D_F} is\ max\} \tag{10}$$

To satisfy the degree of match, the fuzzy based six membership functions ϑ_l, $l = 1, \ldots, 6$ based on the objectives, are considered detailed below.

(1) The membership function for removing constraint related to availability of resource (first objective) is described as:

$$\vartheta_1 = \begin{cases} 1, AR_u(Av_k) \leq R_u(Av_k) \\ 0, AR_u(Av_k) > R_u(Av_k) \end{cases} \tag{11}$$

where, $Av_k \longrightarrow$ activity, $AR_u \longrightarrow$ *available wait current resource*, $R_u \longrightarrow$ *required wait current resource*.

(2) The membership function for handling waiting time (second objective) is

$$T_w(Av_k) = \sum_{k=1}^{NR_T(AV_k)} \sum_{l=1}^{AR_u(Av_k)} wait_{kl}(Av_k) \tag{12}$$

where, $T_w(Av_k)$ tells total waiting time of resources for activity, NR_T represents total number of different resource types and wait is the waiting time.

$$\vartheta_2 = \begin{cases} 1, T_w(Av_k) \le MTR \\ 0, T_w(Av_k) > MTR \end{cases} \tag{13}$$

where, $T_w(Av_k) \longrightarrow$ *total waiting time of resources for activity* Av_k, as calculated in Eq. 12, $MTR \longrightarrow$ *maximum waiting time allowable for resources*, $NR_T \longrightarrow$ *total number of different resource types*, *wait* $\longrightarrow$ *waiting time*.

(3) The membership function for queuing length (third objective) is:

$$Q_R(Av_k) = \sum_{k=1}^{NR_T(Av_k)-1} Q_{R_k}(Av_k) \tag{14}$$

where, $Q_R(Av_k)$ tells queuing length for Av_k, $NR_T(Av_k) - 1$ is the current lacking type of resource causing other resources to be in queue.

$$\vartheta_3 = \begin{cases} 1, Q_R(Av_k) \le MQR \\ 0, Q_R(Av_k) > MQR \end{cases} \tag{15}$$

MQR is the maximum queuing-occupy ration allowable for resources.

(4) Membership function ϑ_4: It is defined using lag time $L_T(Av_k)$ (fourth objective). The $L_T(Av_k)$ at activity Av_k is calculated as:

$$L_T(Av_k) = \sum_{k=1}^{N_T(Av_k)} \sum_{l=1}^{C_c(Av_k)} L_{T_{k,l}}(Av_k) \tag{16}$$

where, $N_T(Av_k)$ is the number of types of resources at Av_k, $C_c(Av_k)$ tells completed cycles at Av_k.

$$\vartheta_4 = \begin{cases} 1, L_T(Av_k) \le ML_w \\ 0, L_T(Av_k) > ML_w \end{cases} \tag{17}$$

where, ML_w is the maximum lag time.

(5) Membership function ϑ_5: It is defined using duration factor D_{rf} (fifth objective) which is calculated as:

$$D_{rf} = mean_{dr}(Av_k) + \frac{std_{dr}(Av_k)}{mean_{dr}(Av_k)} \tag{18}$$

where, $mean_{dr}$ is the mean duration and std_{dr} is the standard deviation. Using the factor, membership function ϑ_5 is computed as:

$$\vartheta_5 = \frac{D_{rf}(Av_k) - min_{df}}{max_{df} - min_{df}} \tag{19}$$

here, max_{df} and min_{df} is the maximum and minimum duration factor.

(6) Membership function ϑ_6 for completion of cycle: X_{cr} as completion ratio of activities (sixth objective) is calculated as:

$$X_{cr} = \frac{X_{co}(Av_k)}{S_{co}(Av_k)} \tag{20}$$

where, X_{co} tells the completed cycles, S_{co} are the stipulated operation cycles. For the completion of cycle, membership function is given as:

$$\vartheta_6 = \frac{max_{cr} - X_{cr}(Av_k)}{max_{cr} - min_{cr}} \tag{21}$$

where, max_{cr} is the maximum completion ratio of resources and min_{cr} is the minimum completion ratio of resources.

Once all the membership functions are determined, the weighted membership value is determined using Eq. 8. Finally, the resource is selected using Eq. 10. Thus, the combined decision based on its membership function using Eq. 8 and is expressed as:

$$FinalD_F = \vartheta_{FinalD_F} \times \sum_{i=1}^{6} \beta_i \times \vartheta_i \tag{22}$$

where, ϑ_{FinalD_F} ensures the limited queuing capacity taken by the current resource at activity ϑ_i. The ϑ_{FinalD_F} is determined using:

$$\vartheta_{FinalD_F} = \begin{cases} 1, \ (Av_u(S_{mat}) + ALLU) \le MQR(S_{mat}) \\ 0, \ (Av_u(S_{mat}) + ALLU) > MQR(S_{mat}) \end{cases} \tag{23}$$

where, $Av_u(S_{mat})$ is available unit of current resource and *ALLU* is allocated unit. Thus, it results into redundant and constraint free resources in S_{mat} having maximum information. The process of resource discovery is explained in Algorithm 1.

The process initializes *n* to *100*, Q_1 to resources and Q_2 to C_x. It initializes a loop to prioritize similarity computation, SM_c, by matching C_x with resources. It computes the time consumed, t_x, for SM_c and condition is checked to generate matching sets, S_{mat}, from C_x with in prescribed time. It prints S_{mat} and t_x. A loop is initialized for matching S_{mat} and SM_c using *FCL* as shown in Table 5 and results out as Qs_{out} with true or false statement. As S_{mat} faces various constraints with respect to time, therefore, six membership functions are described using Eqs. 11–21. Now, another loop computes the decision, D_F, for degree of match and combined decision, ϑ_{D_F}, is calculated using Eqs. 7 and 8, respectively. Finally, ϑ with largest membership value is calculated and combined final decision, $FinalD_F$, is generated using Eqs. 10 and 22, respectively. The process ends up with the display of redundant and constraint free, S_{mat}. S_{mat} is further processed for optimal selection of the resource using Improved Firefly algorithm (*IMPFF*).

3.3.3. Optimal resource selection

The discovered resources (data values) will act as brightest fireflies having maximum similarity match determined with best direction in which brightness of firefly increases and is selected in minimal time. The *IMPFF* is nature inspired multiple heuristic algorithm that helps in optimizing S_{mat} as maximization problem and works through its three basic components, *namely*, optimization function, solution set for selecting variable and rule for optimization. The optimization function is defined as:

$$\max f(x_i) \text{such that} x_i \in S_{mat} \tag{24}$$

A solution for this function is a member of S_{mat} that provides the maximum value of $f(x_i)$ compared to all matched resources in S_{mat}. The optimizibility lies in selecting x_i^* such that $f(x_i^*) \ge f(x_i) \forall x_i \in S_{mat}$. The *IMPFF* aims to select such x_i^*. The working of *IMPFF* is explained as under.

It works on the principle of flashing at night with its three rules to be followed. The first rule suggest to attract unisex fireflies to brighter one. The second rule determines the brightness of firefly using encoded function. The third rule describes that attractiveness that is fully dependent upon brightness and is decreased with increase in distance. If there is no brighter firefly then firefly will move randomly. The light intensity is reciprocal to square of distance (r_c), that is going farther from source. It also describes that if light passes through medium having absorption coefficient (λ_k), then intensity (I_l) changes with r_c and is given as:

$$I_l(r_c) = I_1 \times e^{-\lambda_k \times r_c} \tag{25}$$

where, I_1 is source point intensity. Using physics principle, the $I_l(r_c)$ becomes

$$I_l(r_c) = I_1 \times e^{-\lambda_k \times r_c^2} \tag{26}$$

Since, the computation of $\frac{1}{1+\lambda_k \times r_c^2}$ is easy than $e^{-\lambda_k \times r_c^2}$, therefore, intensity is calculated as:

$$I_l(r_c) = \frac{I_1}{1 + \lambda_k \times r_c^2} \tag{27}$$

Now, the computed I_l also computes the degree of attractiveness as it is measured by what intensity light is flashed by the fireflies. Therefore, the attractiveness of fireflies is described as:

$$A_f(r_c) = \frac{A_1}{1 + \lambda_k \times r_c^2} \tag{28}$$

where, $A_1 \longrightarrow$ *attractiveness at* $r_c = 0$.

If firefly at exact position $x_i^* = \{x_1^*, x_2^*, \ldots, x_n^*\}$ is brighter in comparison with fireflies at position $x_i = \{x_1, x_2, \ldots, x_n\}$, then, x_i firefly will move towards $x_i{}^*$ having better matched similarity in resource's constraints (fireflies). Thus, the updated position of x_i firefly is given as:

$$x_i = x_i + A_1 \times e^{-\lambda_k \times r_c^2} \times \left(x_i' - x_i\right) + \beta \times \xi \tag{29}$$

where, β is randomized parameter whose value lies in range of [0,1]. ξ consists of vectors with random step-length. β and ξ are the parameters that has the property of the adaptive adjustment and leads to algorithm's convergence. At the beginning of the search larger value of β (random) and smaller ξ is selected, and smaller β, larger ξ's values is used later in the search. In addition, in order to highlight the random movement at the beginning of the algorithm and reduce the random motion in the later stage, random movement step is added. $A_1 \times e^{-\lambda_k \times r_c^2} \times (x_i' - x_i)$ describes the degree of attractiveness of x_i towards x_i'. In real-life, A_1 is considered as 1, *i.e.*, $A_1 = 1$. The parameters of the algorithm are adjusted adaptively according

Algorithm 1 Proposed Resource Discovery Algorithm.

Require: Concepts (C_x), Similarity Computation (SM_c), time (t_x), input query (S_{mat}), query ($Q_{s_{in}}$), membership function (ϑ), decision (D_F), combined decision ($FinalD_F$)

begin
//Discovery process
initialize $n \longleftarrow 1000$, $Q_1 \longleftarrow resources$, $Q_2 \longleftarrow C_{x_{0..n}}$
for $(i = 0..n)$ **do**
    prioritize $SM_c \longleftarrow (C_x, Q_1)$
end for
for $(i = 0..n)$ **do**
    $t_x \longleftarrow calculate\text{-}time(SM_c)$
    if $(t_x > random([0, 1]))$ **then**
        get S_{mat} from C_x
    end if
    output S_{mat}
    print t_x
end for
//Matching process
for $(i = 0..n)$ **do**
    $Q_{s_{in}[i]} \longleftarrow match(S_{mat}, SM_c)$
    if $\left(Q_{s_{in}[i]} == true\right)$ **then**
        switch $Q_{s_{in}[i]}$ **do**
            case $1 \longrightarrow \left(Q_{s_{in}} : S_{mat} == SM_c\right)$:
                $Q_{s_{out[i]}} = 1;$
                break;
            case $2 \longrightarrow \left(Q_{s_{in}} : S_{mat} <= SM_c\right)$
                $Q_{s_{out[i]}} = 1;$
                break;
            case $3 \longrightarrow \left(Q_{s_{in}} : S_{mat} < SM_c\right)$
                $Q_{s_{out[i]}} = 0/1;$
                break;
            case $4 \longrightarrow \left(Q_{s_{in}} : S_{mat} >= SM_c\right)$
                $Q_{s_{out[i]}} = 1/0;$
                break;
            case $5 \longrightarrow \left(Q_{s_{in}} : S_{mat} > SM_c\right)$
                $Q_{s_{out[i]}} = 0/1;$
                break;
            case $6 \longrightarrow \left(Q_{s_{in}} : S_{mat} \neq SM_c\right)$
                $Q_{s_{out[i]}} = 0;$
                break;
    else
        $Q_{s_{out[i]}} \longleftarrow false$
    end if
    return $Q_{s_{out[i]}}$
end for
//Eliminating constraints with respect to time
initialize six membership functions, *i.e.*, ϑ_l, $l = 1, ..., 6$ *//using Eqs. 11-21*
for (i = 1..l) **do**
    compute $D_F \longleftarrow \vartheta_1 \bigcap \vartheta_2 \bigcap .... \bigcap \vartheta_l$
    compute $\vartheta_{D_F} \longleftarrow \sum_{i=1}^{l} \beta_i \times \vartheta_i$
end for
compute $\vartheta_{FinalD_F} \longleftarrow \left\{S_{mat} \mid \vartheta_{D_F} is\ max\right\}$ *// determining ϑ with largest membership value*
compute $FinalD_F \longleftarrow \vartheta_{FinalD_F} \times \sum_{i=1}^{6} \beta_i \times \vartheta_i$
display $FinalD_F$
end

to the lightness variance of the firefly and the random movement step is determined in accordance with the distance of two firefly. *IMPFF* considers the updated locations of brightest fireflies and selects the brighter one by processing them iteratively against threshold value. The firefly that retains for longer period and has maximum similar matches of concepts is selected as optimal solution. For each iteration, the firefly at farther distance from the brightest firefly is discarded.

Now, considering the directional movement of brightest firefly as P, the other fireflies move towards it and generates K vectors, *i.e.*, $p_1, p_2, \ldots, p_k$. Then the movement of brightest firefly is described as:

$$x_i = x_i + P_i \tag{30}$$

Here, if any direction does not exist for firefly then brightest firefly remains at same position only.

Furthermore, if source attractiveness assignment (that is dependent on intensity of firefly and objective) is considered rather than taking $A_{1_i} = 1$ for each firefly i, one possible solution is to assign firefly's intensity ratio. Let say, firefly (k) is located at x_i' position is much brighter than firefly (l), located at x_i, then firefly (l) at x_i will move towards firefly (k), using Eq. 29. At this time, A_1 is given as:

$$A_1 = \frac{I_k}{I_l} \tag{31}$$

where, at $r_c = 0$, I_k and I_l is the intensity for fireflies, k and l, respectively. In real life, A_1 is described as $e^{I_k - I_l}$ in oder to eliminate significant case for $I_1 = 0$. If $A_1 = I_k$ and intensity is less, then firefly l will take longer time to move towards firefly k. Therefore, it is better to adjust A_1 because in either cases A_1 is directly proportional to intensity at source, I_1. The updated brightest firefly is checked for its new intensity iteratively. The firefly having maximum intensity and attractiveness, is selected as the brightest firefly and it provides the optimal solution against query. The summarized working of *IMPFF* is described in Algorithm 2.

Algorithm 2 Proposed IMPFF Algorithm.

Require: Position (x_i, x_i*), Intensity (I_l), Attractiveness (A_f), Euclidean Distance(euc_{dis})

begin
  initialize $k \longleftarrow 0, n \longleftarrow 1000$
  // Call procedure Resource Discovery Algorithm (Algorithm 1)
  for ($k = 1..n$) **do**
    calculate $I_l(r_c) = \frac{I_1}{1+\lambda_k \times r_c^2}$
    calculate $A_f(r_c) = \frac{A_1}{1+\lambda_k \times r_c^2}$
    generate $x_i = x_i + P_i$
  end for
  for ($i = 1..n$) **do** *//fireflies*
    for ($j = 1..n$) **do** *//brightest firefly*
      if ($I_l(x_i) < I_l(x_i*)$) **then**
        $brightest \longleftarrow firefly_j$ *//selecting brightest firefly*
        calculate $euc_{dis}(brightest, firefly_i)$
        move $firefly_i$ to $brightest$
      end if
    end for
  end for
  $rank(brightest - fireflies)$
  for ($i = 1..n$) **do**
    calculate $I_{l_{new}(x_i)} \longleftarrow brightest_i$ *// calculating intensity of brightest fireflies*
    if ($I_l(x_i) < I_{l_{new}}(x_i)$) **then**
      select $B \longleftarrow brightest(firefly_i)$
    end if
  end for
  display(B) //selecting brightest best matched firefly
end

Algorithm 2 begins with calling Algorithm 1 and initializing loop for calculating intensity and degree of attractiveness for the firefly. It generates new position for updated fireflies. Now, the process is executed iteratively to find and select brightest firefly. It compares the intensities of previous with updated fireflies. A random brightest firefly is selected and its distance with other fireflies is checked using euclidean distance. The other firefly will be attracted towards brightest firefly. The brightest fireflies are ranked according to their intensity and another iteration is processed for checking their intensity. The firefly with maximum intensity is selected providing optimal matched resource against query.

Fig. 5. Study Area.

Table 6
Sensors parameters .

Parameters	Values
Frequency Range	25 KHz
Number of Sensor Nodes	13
Max. TX Distance (single hop)	35 m
Max. TX Distance (multi hop)	100 m
Min. CH candidate range	70%
Max. CH candidate range	90%
Packet with error ratio (1/1000)	10
Batteries (Panasonic Eneloop Pro-Ni-MH)	2550 mAh

3.4. Application web terminal

It is the Web Terminal, where clients can register themselves for accessing the services provided by the framework. Here, each client is given a unique *ID* and password at the time of registration, for authentication and authorization. By this terminal, user puts a query and in frequent time, he/she has a response against the required query.

4. Experimental results and discussion

4.1. Study area

The datasets for *ORSF-IoT* is collected from Sutlej river using underwater sensors, at Bassi near Nurpur Bedi, Ludhiana,India (*see* Fig. 5). The Ludhiana city is situated on banks of Sutlej and is a hub of major industries/factories of garments, bicycles, glass making, *etc*. The river acts as a source of irrigation water for southern villages of Punjab. The river's quality is deteriorating due to the discharges of effluents from domestic sewage as well as from various industrial units. Therefore, it needs to be monitored regularly and take timely preventive measures.

The various parameters of sensors that are taken for deployment are shown in Table 6 [30].

The objective of framework on this river is to automate the process of monitoring through accurate deterioration detection by analyzing physical parameters such as Electrical Conductivity, *pH*, Temperature, Chloride, and Dissolved Oxygen, to minimize man-power, on-site sampling processes; and to provide timely alarms to monitoring stations.

4.2. System performance

For real-time measurements, the system's performance is analyzed on the collected data from destined location, *i.e.*, Sutej river. The data value includes Electrical Conductivity, *pH*, Temperature, Chloride, and Dissolved Oxygen of river water and are recorded for three months. The collected parameters are shown in Table 7, measured observation (collected) are shown in Tables 8, 9, 10, 11, 12.

The performance of *IMPFF* is tested on data values of these parameters using benchmark functions as described in Table 13.

These functions are widely used for evaluating optimization of nature-inspired algorithms. The *Sphere* is a multi-modal function having global minimum solution, $f_* = 0$ located at $x^* = (0,0,...,0)$, *Rosenbrock* is single modal function having global minimum $f_* = 0$ located at $x^* = (1,1,...,1)$ and *Ackley* is multi-modal function having global minimum solution, $f_* = 0$ located at origin $x^* = (0,0,...,0)$. These functions are implemented to validate the optimizibility for *IMPFF* algorithm. For the

Table 7
Water Physical Parameters .

Parameter	Nominal Range	Computational Formula	Effects/ Reasons
Electrical Conductivity (mg/l)	350–500	$EC(mg/l) = \frac{TDS\ (ppm)}{0.64}$	The conductivity is a measure of total dissolved solids *(TDS)* in water (in terms of ppm or mg/l). It is related with dissolved salts in water. It's increased level is critical for aquatic animals because they have specific tolerance ranges and is dependent on climatic conditions and pollutants which increases during or after rainfall.
pH	6.6–7.1	$pH = -\log[H_3O^+]$	*pH* factor is used to estimate the acidic or alkaline nature of water. The lesser value of pH results into acidic nature of water and it affects various flora and fauna in water.
Temperature (°C)	25–27.5	$\Delta T = \frac{Q}{mc}$	Temperature plays an important role for growth of flora and fauna within water. The increased level of temperature increases the chemical reaction and hence affects organic beings. Temperature is simply measured as amount of heat released or absorbed by organic beings.
Chloride (mg/l)	80–95	$NaCl + H_2O \longrightarrow Na^+ + Cl^- + H_2O$	The high chloride concentration in fresh water is toxic to aquatic organisms and is a threat to survival, growth and reproduction of the species.
Dissolved Oxygen (mg/l)	7–8	DO = Amount of Titrant Used (mL)	Dissolved oxygen *(DO)* is a measure of total amounts of Titrant in water. It is required by aquatic animals and phytoplankton for respiration and photosynthesis processes, respectively. it is necessary source of life residing underwater.

Table 8
Electrical Conductivity .

Electrical Conductivity (mg/l) (March, April, May 2019)											
Days	March	April	May	Days	March	April	May	Days	March	April	May
1	433	431	452	12	465	388	538	23	375	494	546
2	448	447	465	13	472	396	532	24	357	499	480
3	467	459	470	14	451	408	526	25	341	502	482
4	417	441	481	15	399	424	521	26	327	496	498
5	395	426	494	16	444	430	515	27	353	485	510
6	377	401	499	17	464	442	510	28	380	481	524
7	422	383	506	18	472	461	477	29	404	477	536
8	457	381	512	19	437	476	467	30	422	471	491
9	402	379	520	20	407	483	490	31	349	-	502
10	414	386	529	21	387	491	502				
11	432	387	534	22	379	492	521				

same, different non-linear unconstrained equations are considered as hives for fireflies to reach local minima as global optimal solution. The equations are $e^{(-(x-4)^2-(y-4)^2)}$, $e^{(-(x+4)^2-(y-4)^2)}$, $e^{(-(x+4)^2-(y+4)^2)}$, $e^{(-(x-4)^2-(y+4)^2)}$, $2 \times e^{(-x^2-(y+4)^2)}$, $2 \times e^{(-x^2-y^2)}$, $2 \times e^{(-x^2-(y-4)^2)}$, $2 \times e^{(-(x-4)^2-y^2)}$ and $2 \times e^{(-(x+4)^2-y^2)}$. In Fig. 6, the peaks generated through these equations are shown as *3D-view*.

The data values of physical parameters of river water as fireflies, are considered for evaluating the performance of benchmark functions. Fig. 7 is the top view of Fig. 6 in which hives are shown as round circles and fireflies are shown as blue dots. These fireflies are attracted towards the center of hives on the basis of degree of attractiveness towards brightest one. The firefly having euclidean distance farther from brighter one, move randomly and wait for other brightest fireflies in range. The firefly with lowest threshold value is discarded and the iteration is processed for selecting the brightest firefly (optimal selection), *i.e.*, the erred data value of physical parameters which exceeds the nominal range.

For verifying the performance of framework, the benchmark functions are being tested for *IMPFF* and existing nature-inspired algorithms such as *PSO, ABC* and *CA*. The *PSO* considers real-number randomness and global communication among resources as swarm particles. It optimizes the function by iteratively improving user solution with regard to given measure of quality. *ABC* provides optimal solution by considering the position of nectar amount of food as resources corresponding to quality (fitness) of associated outcome. *CA* is an extension to conventional genetic algorithm. Its knowledge component (belief space) is divided into distinct categories that represent different domains of knowledge with respect to its search space. The belief space is updated iteratively to select the best resources using fitness function among the population. In short, these algorithms are used because they provide highly influential results in terms of optimization.

Table 9
pH Data .

pH(March, April, May 2019)											
Days	March	April	May	Days	March	April	May	Days	March	April	May
1	6.9	7	6.9	12	6.9	7	6.8	23	7	7.2	6.9
2	7.1	7	6.7	13	6.8	6.9	7	24	7.2	7	6.8
3	6.8	6.8	7.1	14	6.7	6.8	7.1	25	7.1	7.1	6.9
4	6.9	6.9	7.2	15	7	7	7	26	6.8	7	6.8
5	7	7	7.1	16	7.2	7.1	6.9	27	6.9	7	6.9
6	6.8	6.9	7.1	17	7.1	6.9	7	28	6.8	6.9	7
7	6.9	7	7.2	18	7	7	6.8	29	6.8	7.1	6.9
8	6.8	7.1	6.9	19	6.9	7	6.9	30	6.7	7	7
9	6.9	7.2	7	20	6.8	7.1	7.1	31	6.6	-	6.8
10	6.9	7.1	6.9	21	6.8	6.9	7.1				
11	7	7.1	7	22	6.9	7.1	7				

Table 10
Temperature Data .

Temperature($^\circ$C)(March, April, May 2019)											
Days	March	April	May	Days	March	April	May	Days	March	April	May
1	23.1	25.3	26.2	12	26.9	25.6	27.2	23	27.2	25.4	27.3
2	23.6	25.4	26.1	13	26.8	25.5	27	24	27.5	25.5	27
3	24.2	25.5	26	14	27.7	25.5	27.1	25	27.6	25.8	27.1
4	24.8	25.5	26.1	15	27.6	25.4	27	26	27.4	26.1	26.9
5	25.6	25.4	26.3	16	27.7	25.6	27.3	27	27.5	26.3	27
6	27.6	25.3	26	17	26.5	25.4	27.3	28	27.6	26.1	26.9
7	27	25.4	26.3	18	26.7	25.7	27.4	29	27.7	26.2	27.1
8	27.3	25.5	26.4	19	26.9	25.6	27.2	30	27.7	26.4	27.2
9	26.7	25.3	26.8	20	26.9	25.7	27.1	31	27.9	-	27.6
10	26.6	25.6	27.1	21	27	25.8	26.9				
11	26.7	25.3	27	22	27.2	25.6	27.1				

Table 11
Chloride Data .

Chloride(mg/l)(March, April, May 2019)											
Days	March	April	May	Days	March	April	May	Days	March	April	May
1	94.4	85.3	86.7	12	76.9	81.3	80.8	23	77.9	92.1	86.1
2	94.9	80.3	85.1	13	75.5	82	80.5	24	78.3	93.3	84.7
3	80.4	77.3	84.7	14	73.9	83.2	78.8	25	79	94.4	84
4	76.4	75.4	84.2	15	73.5	84.2	79.7	26	80.2	94.6	83.4
5	80.6	76	83.6	16	72.8	85.4	82.1	27	81.1	92.9	82.6
6	83.2	72.9	82.6	17	75	86.4	82.5	28	83.4	92.2	82.1
7	82.1	72.4	82.5	18	72.5	87.1	83.6	29	84.2	91	81.8
8	79	74.2	81.9	19	71.4	87.4	84	30	85.4	90.2	81
9	77.9	75.3	81.4	20	70.3	88.7	83.9	31	86.2	-	82.4
10	81	77.8	81.1	21	72.4	90.2	85.1				
11	80.2	78.9	81.2	22	76.5	90.6	85.6				

Table 12
Dissolved Oxygen .

Dissolved oxygen(mg/l)(March, April, May 2019)											
Days	March	April	May	Days	March	April	May	Days	March	April	May
1	7.9	7.7	7.9	12	7.7	7.4	7.7	23	7.6	7.6	7.7
2	7.8	7.6	7.9	13	7.7	7.5	7.6	24	7.3	7.5	7.8
3	7.8	7.5	7.8	14	7.6	7.3	7.6	25	7.5	7.5	7.7
4	7.6	7.5	7.9	15	7.7	7.5	7.8	26	7.4	7.3	7.6
5	7.9	7.6	7.6	16	7.6	7.3	7.6	27	7.3	7.2	7.7
6	7.6	7.6	8	17	7.5	7.3	7.7	28	7.3	7.2	7.6
7	7.7	7.6	7.9	18	7.4	7.4	7.7	29	7.4	7.3	7.6
8	7.6	7.5	7.7	19	7.8	7.3	7.6	30	7.2	7.2	7.7
9	7.7	7.4	7.7	20	7.8	7.4	7.5	31	7.3	-	7.8
10	7.8	7.5	7.6	21	7.6	7.1	7.6				
11	7.7	7.3	7.7	22	7.4	7.2	7.8				

Table 13
Benchmark Functions .

S.No	Function Name	Function
1	Sphere	$f_1(x) = \sum_{i=1}^{n} x_i^2,\ \text{for } 0 \leq x_i \leq n$
2	Rosenbrock	$\sum_{i=1}^{n} \left[(x_i - 1)^2 + 100 \times \left(x_{i+1} - x_i^2 \right)^2 \right],\ \text{for } 0 \leq x_i \leq n$
3	Ackley	$-20 \times \left(1 - e^{\left(-0.2 \times \sqrt{\sum_{i=0}^{n} \frac{x_i^2}{n}} \right)} \right) - e^{\left(\sum_{i=0}^{n} \frac{cos(2 \times \pi x_i)}{n} \right)} + 20 + e^1,\ \text{for } 0 \leq x_i \leq n$

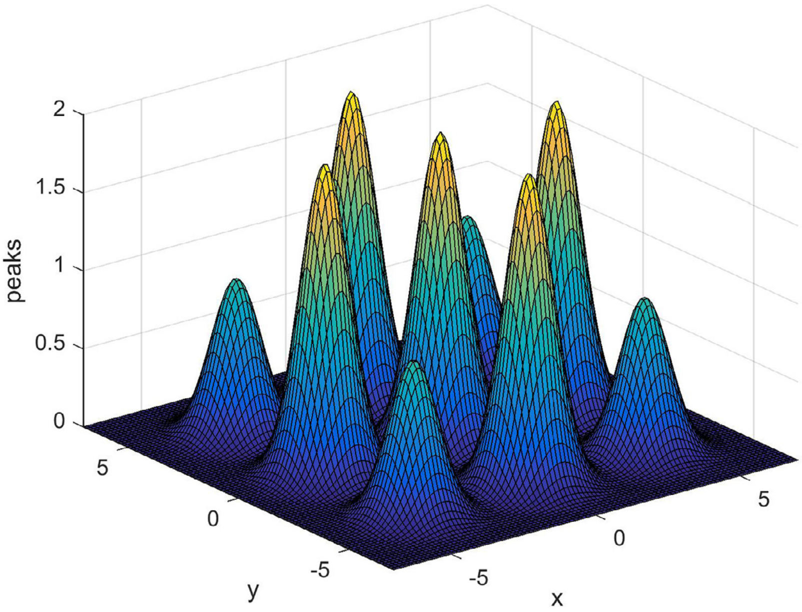

Fig. 6. Working of *IMPFF* - Range of Firefly.

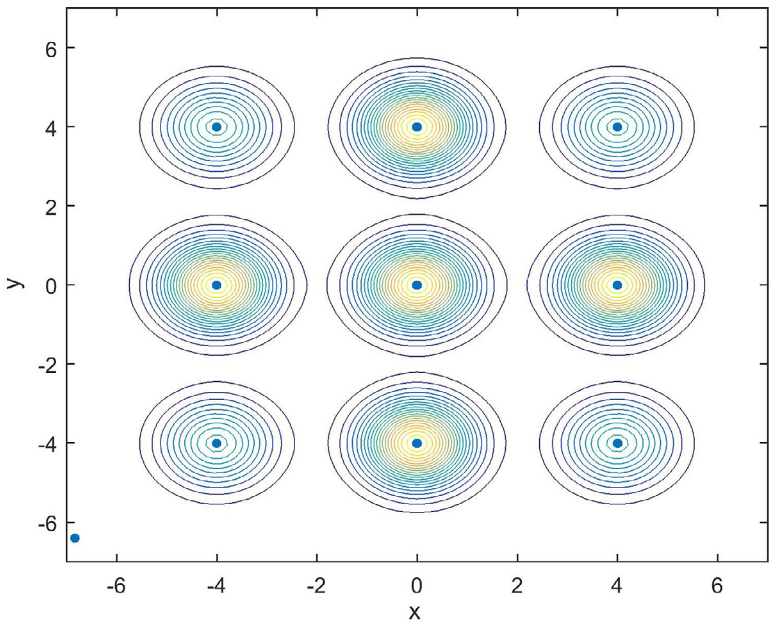

Fig. 7. Working of *IMPFF* - Attracted Firefly towards Brightest.

Table 14
Comparison among Approaches .

Functions	Approaches	Metrics				
		Mean	Standard Deviation	Standard Error of Mean *(SEM)*	Processing Time (seconds)	Execution Time (seconds)
Sphere	*PSO*	8.94E+04	35.6324	2.24E+03	0.8745	1.9559
	ABC	7.87E+04	27.7562	1.97E+03	0.8126	2.0124
	CA	7.36E+04	24.7486	1.85E+03	0.5468	1.0656
	IMPFF	6.75E+04	14.3554	1.69E+03	0.1046	1.0018
Rosenbrock	*PSO*	8.72E+04	34.6502	2.19E+03	0.8621	2.4257
	ABC	7.61E+04	26.774	1.91E+03	0.7453	1.5987
	CA	7.11E+04	23.7664	1.78E+03	0.4526	1.0425
	IMPFF	6.51E+04	13.3732	1.63E+03	0.2412	0.0955
Ackley	*PSO*	8.25E+04	38.7041	2.07E+03	0.7421	1.7864
	ABC	7.21E+04	30.8279	1.81E+03	0.6354	1.3289
	CA	7.65E+04	27.8203	1.92E+03	0.4452	1.2267
	IMPFF	6.95E+04	17.4271	1.74E+03	0.3173	1.0148

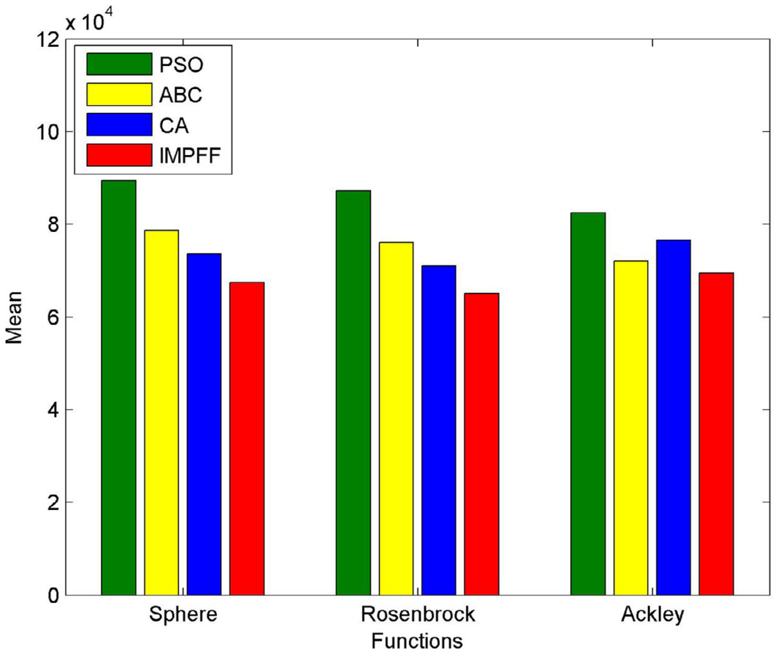

Fig. 8. Mean Metric using PSO, ABC, CA and IMPFF.

The proposed *IMPFF* is compared with these algorithms in terms of metrics *namely,Mean, Standard Deviation, Standard Error of Mean (SEM), Processing Time, Execution Time*; in order to evaluate its efficacy. The low *Mean* denotes the minimal utilization of computing resources in finding the matched sets and increasing the speed of computation. Lower *Standard Deviation* denotes the maximum resources that converges to optimal selection against query. It tends to denote maximum matched sets near to mean. The low *SEM* provides the stability in finding the rightful resource with computation of maximum resources. It denotes best matched solution with higher accuracy. The algorithm's speed is evaluated in terms of *Processing* and *Execution Time*. The complex algorithm consumes more processing time in search computation and has increase in execution time to process the query. Less processing and execution time denotes the efficiency of algorithm in computing the query and to find the optimal solution in minimal time. For testing purposes, the data values of physical parameters as fireflies are selected. The process is run for *30* times with *1000* iterations to find the erred value that exceeds the limit of nominal ranges (given in Table 7) and the system will provide timely alarms/warning to the monitoring stations to take preventive measures. The comparison among various benchmark functions, is described in Table 14.

The graphical notation of comparison of evaluation metrics for functions on the basis of Table 14 is shown in Figs. 8–12. These figures are obtained with the average *Mean, Standard Deviation, SEM, Processing Time, Execution Time* of *30* times test statistics. Fig. 8 shows the comparison of *Mean* for the functions. It is found that *PSO, ABC* and *CA* provides larger mean

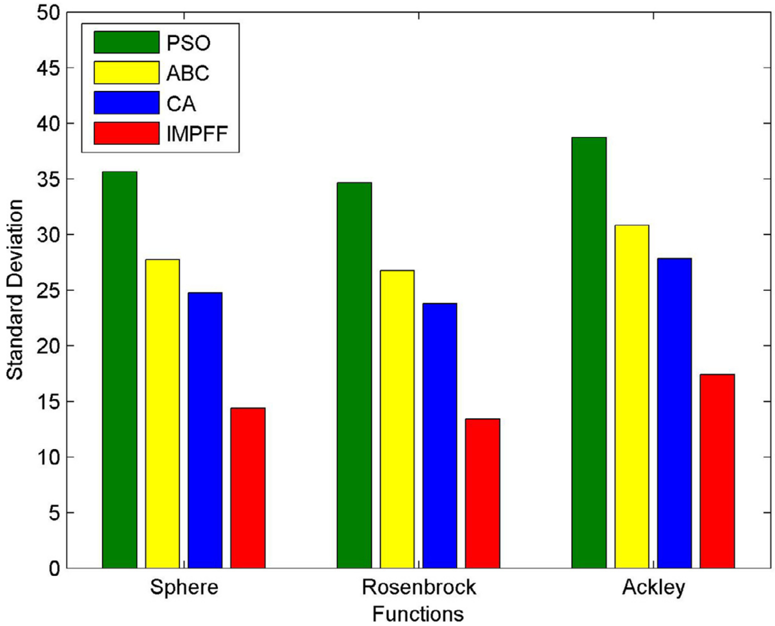

Fig. 9. Standard Deviation Metric using PSO, ABC, CA and IMPFF.

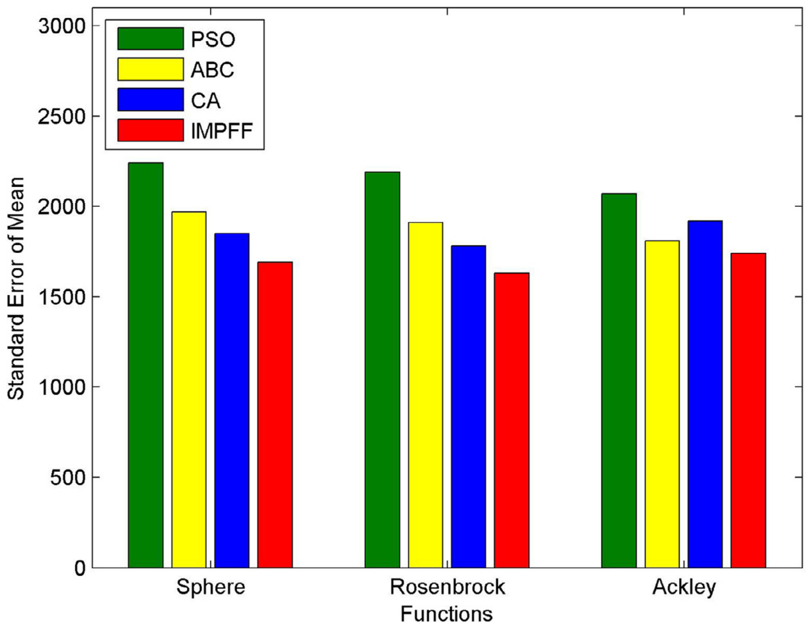

Fig. 10. Standard Error of Mean Metric using PSO, ABC, CA and IMPFF.

as compared with *IMPFF*. This suggest that *IMPFF* is more stable in providing accurate erred parametric values than other algorithms.

Fig. 9 shows the *Standard Deviation* for the functions. It is observed that *IMPFF* provide minimum standard deviation as compared with *PSO, ABC* and *CA*. Thus, it implies that *IMPFF* converges to select optimal solution. Fig. 10 shows the *SEM* for the functions. It is observed that *IMPFF* provide minimum *SEM* as compared with *PSO, ABC* and *CA*. Thus, it implies that *IMPFF* helps in finding the rightful resource with computation of maximum resources. Fig. 11 shows the comparison of *Processing Time* of *IMPFF* and existing nature-inspired algorithms. It is found that *IMPFF* works smoothly and take *0.1-0.3* (seconds) providing efficient search with respect to other algorithms. Fig. 12 shows the comparison of *Execution Time* for functions processed using *IMPFF* and existing nature inspired algorithms. It is postulated that *IMPFF* takes minimum time

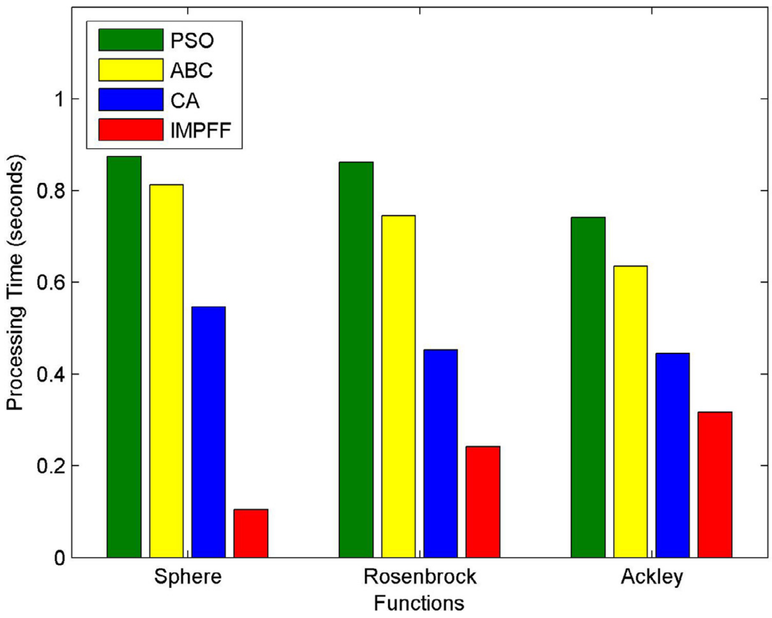

Fig. 11. Processing Time Metric using PSO, ABC, CA and IMPFF.

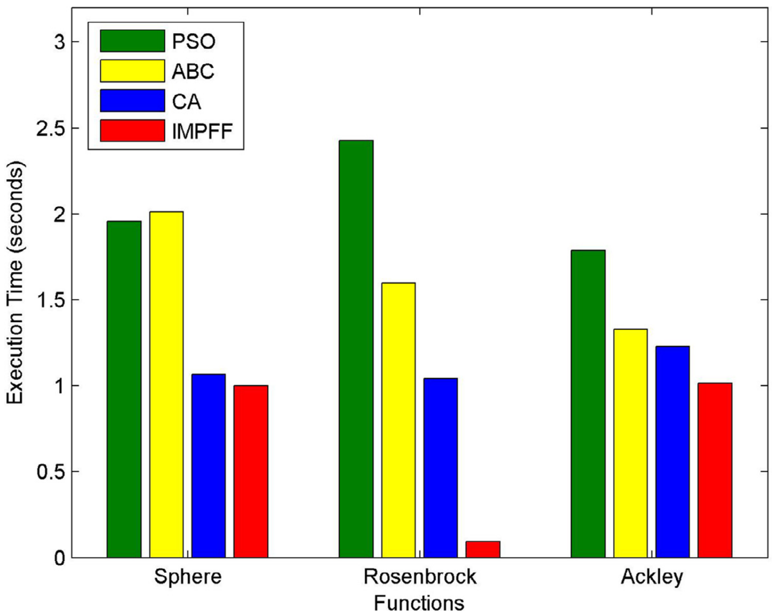

Fig. 12. Execution Time Metric using PSO, ABC, CA and IMPFF.

for executing a process against query, in comparison with *PSO, ABC* and *CA*. Thus, *IMPFF* improves the performance of system by taking less execution time.

Further, to support its efficacy, the algorithms are observed for its iterations that provides the best cost assumption for the benchmark functions as shown in Figs. 13 - 15. It presents the comparison of *PSO, ABC, CA* and *IMPFF* for evaluating cost to check framework's performance with respect to *Processing* and *Execution Time*; processed for functions, *namely, Sphere, Rosenbrock* and *Ackley*. It runs *30* times for *1000* iterations independently for these functions. Figs. 13 - 15 are the illustrations for the best outcomes of the iterations.

It is observed from Fig. 13 that *IMPFF* converges slow providing the lowest cost as compared with other approaches. Fig. 14 describes that *IMPFF* goes downward to negative whereas maximum cost for other methods have their estimated cost between *0–10* and is constant throughout. Fig. 15 shows the lowest cost for *IMPFF* due to slow convergence as compared with *PSO, ABC* and *CA*. It is depicted that *IMPFF* provides the best cost as compared with other algorithms.

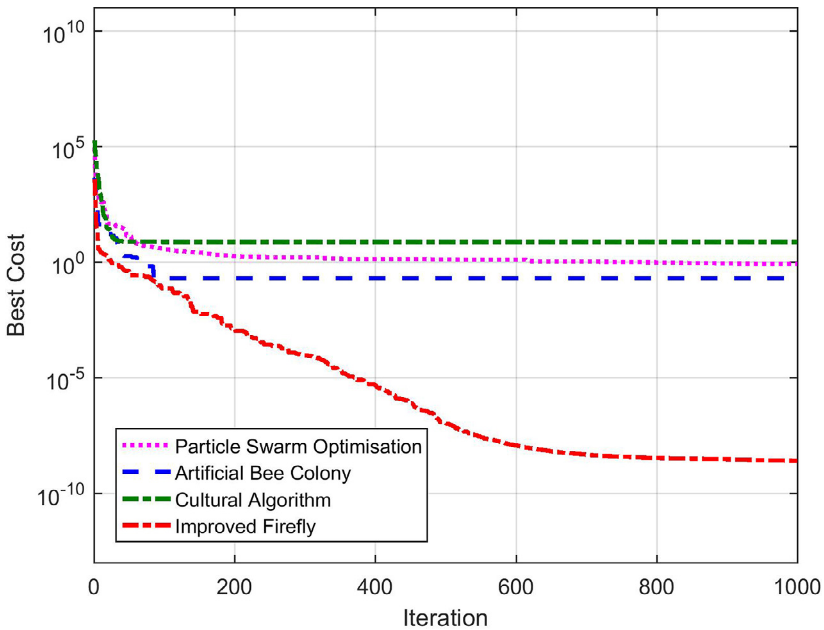

Fig. 13. Comparison among Sphere based Cost Function solved using PSO, ABC, CA and IMPFF.

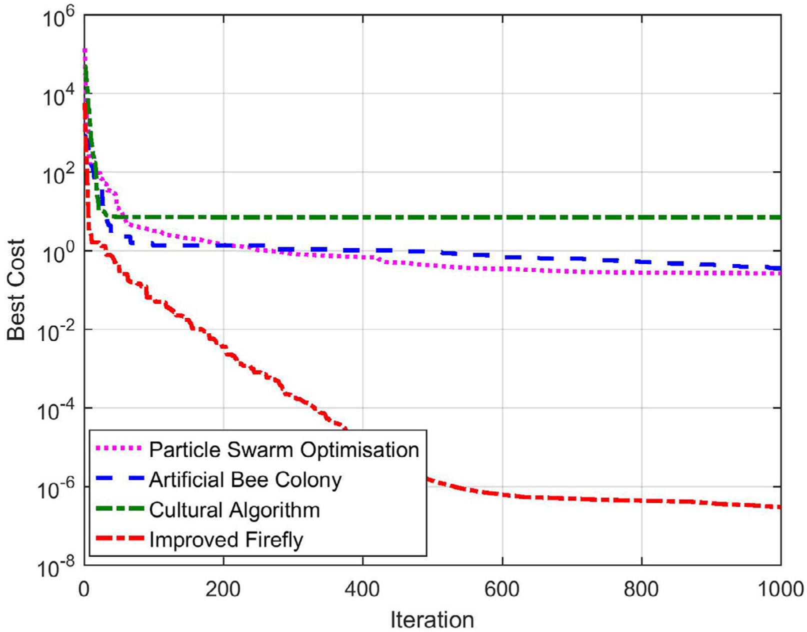

Fig. 14. Comparison among Rosenbrock based Cost Function solved using PSO, ABC, CA and IMPFF.

Thus, it is concluded that *IMPFF* significantly provide optimal solution for selecting the parametric value having exceeded limit as the brightest firefly in less time and providing timely alarms to the system for corrective measurements. The results justifies the statement providing minimum mean and standard deviation with less time consumption; as compared with existing nature-inspired algorithms.

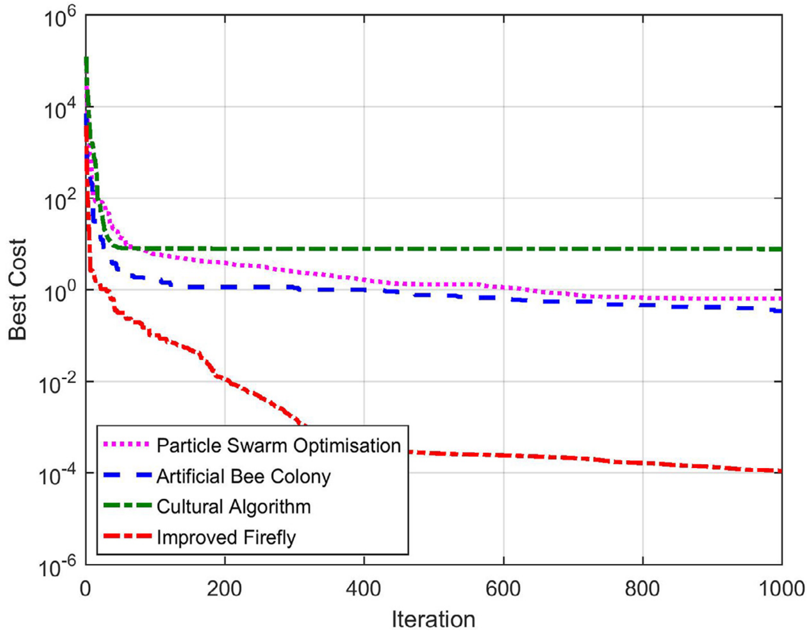

Fig. 15. Comparison among Ackley based Cost Function solved using PSO, ABC, CA and IMPFF.

5. Conclusion

ORSF-IoT intends to represent knowledge using semantic based Shared Virtual Composite Ontology. The ontology has prescribed rules for relations and concepts and eliminates the interoperability issue to a great extent. The discovery of resources is achieved using Fuzzy control rules that interprets the formed knowledge. Further, the discovered resources are investigated for the selection of optimal resource using *IMPFF* algorithm, the performance of which is analyzed with respect to various evaluation metrics, *namely, mean, standard deviation, execution* and *processing time*.

The results obtained confirm the effectiveness of the *IMPFF* with respect to these evaluation metrics. It is found that it provides the optimal selection of resource as brightest firefly that has maximum contextual information against input query. Moreover, it provides faster convergence that results in selecting the best resource when compared with its counterparts, *namely, PSO, ABC* and Cultural. Thus, the framework provides an efficient method for optimal resource discovery and selection.

Declaration of Competing Interest

All authors have participated in (a) conception and design, or analysis and interpretation of the data; (b) drafting the article or revising it critically for important intellectual content; and (c) approval of the final version.

This manuscript has not been submitted to, nor is under review at, another journal or other publishing venue.

The authors have no affiliation with any organization with a direct or indirect financial interest in the subject matter discussed in the manuscript

References

[1] Datta SK, Bonnet C. Describing things in the internet of things: From core link format to semantic based descriptions. In: Proceedings of the IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW). IEEE; 2016. p. 1–2.
[2] Delicato FC, Pires PF, Batista T. The Activities of Resource Discovery and Resource Estimation. Cham: Springer International Publishing; 2017. p. 33–44. doi:10.1007/978-3-319-54247-8_4. 978-3-319-54247-8
[3] Perera C, Vasilakos AV. A knowledge-based resource discovery for Internet of Things. Knowl Based Syst 2016;109:122–36. doi:10.1016/j.knosys.2016.06.030.
[4] Perera C, Zaslavsky AB, Christen P, Compton M, Georgakopoulos D. Context-aware sensor search, selection and ranking model for internet of things middleware. CoRR 2013;abs/1303.2447.
[5] Datta SK, Bonnet C. Search engine based resource discovery framework for Internet of Things. In: Proceedings of the 4th global conference on consumer electronics (GCCE). IEEE; 2015. p. 83–5.
[6] Chandrasekher K, Lee K, Kairouz P, Pedarsani R, Ramchandran K. Asynchronous and noncoherent neighbor discovery for the iot using sparse-graph codes. In: Proceedings of the IEEE international conference on communications (ICC). IEEE; 2017. p. 1–6.
[7] Anas M, Khatab S, Badr A. Heuristiciot: a framework for augmenting heuristic search algorithms by internet-of-Things data. Int J Comput Inf Sci 2016;12(1):133–41.
[8] Shemshadi A., Sheng Q.Z., Zhang W.E., Sun A., Qin Y., Yao L.. Searching for the Internet of Things on the web: where it is and what it looks like. arXiv:1607068842016.

[9] Thakur A, Sathiyanarayanan R, Hota C. Steep: speed and time-based energy efficient neighbor discovery in opportunistic networks. Wirel Netw 2018. doi:10.1007/s11276-018-1721-4.
[10] Onasanya A, Elshakankiri M. Smart integrated iot healthcare system for cancer care. Wirel Netw 2019. doi:10.1007/s11276-018-01932-1.
[11] Naik K, Patel S. An open source smart home management system based on iot. Wirel Netw 2018:1–7.
[12] Datta SK, Gyrard A, Bonnet C, Boudaoud K. A Semantic Engine for Internet of Things: Cloud, Mobile Devices and Gateways. In: Proceedings of the 4th International workshop on extending seamlessly to Internet of Things, in conjunction with the 9th international conference on innovative mobile and internet services in ubiquitous computing (IMIS 2015). Blumenau, Brazil; 2015. https://hal.archives-ouvertes.fr/hal-01206822
[13] Chun S, Seo S, Oh B, Lee K-H. Semantic description, discovery and integration for the Internet of Things. In: Proceedings of the IEEE International Conference on Semantic Computing (ICSC). IEEE; 2015. p. 272–5.
[14] Li Z, Chen R, Liu L, Min G. Dynamic resource discovery based on preference and movement pattern similarity for large-scale social Internet of Things. IEEE Internet Things J 2016;3(4):581–9.
[15] Hussein D, Han SN, Lee GM, Crespi N, Bertin E. Towards a dynamic discovery of smart services in the social Internet of Things. Comput Electr Eng 2017;58:429–43.
[16] Ueda K, Yamashita N. Erratum to: a regularized newton method without line search for unconstrained optimization. Comput Optim Appl 2017;66(1):219–22.
[17] Daniello G, Loia V, Orciuoli F. A multi-agent fuzzy consensus model in a situation awareness framework. Appl Soft Comput 2015;30:430–40.
[18] Rapti E, Karageorgos A, Houstis C, Houstis E. Decentralized service discovery and selection in Internet of Things applications based on artificial potential fields. Serv Orient Comput Appl 2017;11(1):75–86.
[19] Bröring A, Datta SK, Bonnet C. A categorization of discovery technologies for the Internet of Things. In: Proceedings of the 6th international conference on the Internet of Things. ACM; 2016. p. 131–9.
[20] Zhou Y, De S, Wang W, Moessner K. Search techniques for the web of things: a taxonomy and survey. Sensors 2016;16(5):600.
[21] Zhang M, Zheng R, Wu Q, Wei W. A novel multi-x cooperative decision-making mechanism for cognitive Internet of Things. J Netw 2012;7(12):2104–11.
[22] Li S, Oikonomou G, Tryfonas T, Chen TM, Da Xu L. A distributed consensus algorithm for decision making in service-oriented Internet of Things. IEEE Trans Ind Inf 2014;10(2):1461–8.
[23] Bui K-HN, Jung JE, Camacho D. Game theoretic approach on real-time decision making for iot-based traffic light control. Concur Comput: Pract Exp 2017;29(11):e4077.
[24] Garcia-de Prado A, Ortiz G, Boubeta-Puig J. Collect: collaborative context-aware service oriented architecture for intelligent decision-making in the Internet of Things. Expert Syst Appl 2017;85:231–48.
[25] Wang S, Fan C, Hsu C-H, Sun Q, Yang F. A vertical handoff method via self-selection decision tree for internet of vehicles. IEEE Syst J 2016;10(3):1183–92.
[26] Halder S, Ghosal A, Conti M. Limca: an optimal clustering algorithm for lifetime maximization of Internet of Things. Wireless Networks 2019:1–19.
[27] Bharti M, Saxena S, Kumar R. Intelligent resource inquisition framework on Internet-of-Things. Comput Electr Eng 2017;58:265–81.
[28] Shelby Z.. Constrained restful environments (core) link format2012;.
[29] Pedersen T, Patwardhan S, Michelizzi J. Wordnet::similarity: Measuring the relatedness of concepts. In: Demonstration Papers at HLT-NAACL 2004. Stroudsburg, PA, USA: Association for Computational Linguistics; 2004. p. 38–41.
[30] Jindal H, Saxena S, Kasana SS. A sustainable multi-parametric sensors network topology for river water quality monitoring. Wirel Netw 2018;24(8):3241–65.

Monika Bharti is currently working as an Assistant Professor in Computer Science Engineering Department, Jaypee University of Information Technology, Waknaghat, Solan. She has also worked as an Assistant professor in Chandigarh University for 2.5 years. She did her Ph.D. and M.E. in Computer Science from Thapar Institute of Engineering and Technology, India. She was awarded with both JRF and SRF from DeitY, Govt. of India. She has guided 4 M.E. thesis. She has 23 international publications in reputed journals and conferences. Her area of research includes cloud computing, networking, and Internet-of-Things.

Rajesh Kumar is Professor, CSED, Thapar Institute of Engineering and Technology, India. He obtained his M.Sc., M.Phil. and PhD from IIT Roorkee. He has more than 18 years of teaching and research experience. He has more than 78 publications in Journals and conferences. He is member of professional societies of repute. His area of research includes Mathematical Modeling, Wireless Networks, Software Engineering, Pattern Recognition.

Sharad Saxena is Associate Professor, CSED, Thapar Institute of Engineering and Technology, Patiala. He did his Ph.D. (CSE) from Uttarakhand Technical University, India and M. Tech. (CSE) from Shobhit University, India. He has published 45 International Papers, guided 15 M. Tech. and 4 Ph.Ds. He is reviewer of reputed journal and member of IEEE, ACM, CSI societies. His research interest includes WSN and Ad-hoc networks.

Himanshu Jindal is currently working as an Assistant Professor in Computer Science Engineering Department, Jaypee University of Information Technology, Waknaghat, Solan.He has worked as Teaching Associate and Lecturer in Thapar Institute of Engineering and Technology and Rayat Bahra Group of Institutes, for 3 and 2 years, respectively. He obtained his Ph.D. (CSE), M. Tech. (CSA) from TIET, Patiala after completing B.Tech from SUSCET, Tangori. He has 21 international publications in reputed journals and conferences. His research interest includes Wireless Sensor Network, Image Processing, Machine Learning and Internet-of-Things.