



Service Discovery Based on Social Profiles of Objects in a Social IoT Network

Iury Araújo^(✉), Mikaelly F. Pedrosa, Jessica Castro, Eudisley G. dos Anjos, and Fernando Matos

Informatics Center - Federal University of Paraíba (UFPB),
João Pessoa, PB 58051900, Brazil

{iuryrogerio, jessicamaciel}@ppgi.ci.ufpb.br,
mikaelly.felicio@cc.ci.ufpb.br, {eudisley, fernando}@ci.ufpb.br

Abstract. The service interaction provided by objects in IoT networks enables the creation of advanced services to answer application requests. However, the growing number of objects into the IoT network, besides its ad hoc structure, are disturbing some functionalities, such as service discovery. Therefore, when searching for services, the navigability is impaired because the system needs to sweep a great quantity of objects without a previous organization. Social Internet of Things (SIoT) emerged as an alternative to solve several problems faced by IoT through the concept of social networks. In SIoT each object has its own social profile, which contains its characteristics and information, and are organized by relationships. Thus, this research propose a solution for service discovery in a SIoT network. This solution uses the relationships between objects to improve the discovery scalability and considers their social profiles to meet the requisitions in a more satisfactorily way. Simulated results demonstrates the solution performance to answer service requisitions in an urban SIoT network.

Keywords: Social Internet of Things · Social network · Service discovery

1 Introduction

Urban spaces are each day more crowded, causing a variety of problems related to the quality of life, such as increased pollution, resource shortage, limited access of public services, urban mobility and security [14]. Thereby, intelligent solutions for these problems are always in great demand, which, using technology, are most likely to solve or mitigate problems caused by urban expansion. A large amount of these solutions are developed using the concept of smart environments, defined as a “small world” where different types of smart objects work together to improve the life of its inhabitants [17]. This concept has fomented many research areas, for instance, smart cities, smart healthcare, smart grid, that despite their differences, they have in common the performance of several smart

objects working together to accomplish many tasks. Increasingly, these objects acquire the capacity to connect to the internet, whether to perform actions, gather, send or process data. This capability facilitates collaboration between objects, thus providing greater functionality for applications [16].

The smart environment scenario is possible through the implementation of the Internet of Things (IoT) paradigm. IoT is described as the pervasive presence around us of “things” or objects, which over unique address schemes, are allowed to interact between them and collaborate to achieve a specific goal [3]. These objects are the technological devices of our daily life, such as computers, tablets, smartphones, sensor nodes and so forth. However, the growing number of connected objects is a concern for IoT. According to [9], the number of connected objects is already larger than the world population and will reach fifty billions until 2020. This increase directly affects the network management regarding its scalability, navigability and the heterogeneity of the objects, making it difficult to perform important paradigm tasks [12].

Service management is also hampered by the increase in the number of objects. Because an object can provide one or more services, the total number of services in a network may increase greatly. Such services are abstractions of functionalities offered by objects, without worrying about the technologies or protocols. IoT architecture components that handle services, such as discovery and service composition, suffer directly from these problems. As a solution, a new paradigm was proposed, which uses the concept of social networks to organize the objects and their services, called Social Internet of Things (SIoT) [5].

SIoT introduces the use of a social network for the organization of objects, in which each object has the potential to form “friendships” with others objects through relationships [4]. From the social network generated it is possible to manage the scalability and navigability of the network, the heterogeneity of objects and the transmission of data, as is already done by human social networks [2]. Although SIoT inherits IoT features, most of them need to be adapted to handle with this new paradigm, for instance the service discovery, which is a key feature to attend applications through the discovery of services that can be later can be compound to perform complex activities. Another challenge is to use the social profile of objects as useful information, as a way to improve SIoT network features.

In this paper we propose a service discovery in a SIoT network based on social profiles (Social Profile Search - SPS). In this discovery we use the social network’s relationships to improve its navigability and scalability, considering also the information and characteristics contained into social profiles of objects to improve the service requisition satisfaction level. To verify the efficiency of the solution, tests were performed with data from an urban SIoT network, comparing the results between SPS and conventional literature solutions.

This paper is organized as follows. Section 2 exposes a overview of researches in SIoT field, including their approaches to service discovery and the social profile. Section 3 discuss concepts of SIoT and its service discovery. In Sect. 4 is presented this paper proposal for a service discovery based on social profiles.

Section 5 discourse about the proposal implementation, describes the tests carried out and presents and discusses the results achieved. At last, Sect. 6 concludes this paper and discuss future researches.

2 Related Work

Because it is a relatively recent area, most SIoT works focus on conceptual issues of the paradigm, such as, SIoT definitions, applicability, necessary infrastructures and proposed architectures. A small number of researches present paradigm implementations. Nevertheless, some researches focus on the social capacity of objects, using their characteristics, information and the potential of the interactions to provide solutions for the management of objects and services. The works that will be presented, also are concerned with the SIoT network functionalities regarding, explicitly or implicitly, the service discovery.

The SIoTcampus proposed by [1] exemplifies the use of SIoT for the dissemination of academic information among users of a university campus. this work proposes an exclusive smart objects network in which context information, such as location, calendar, user, etc., are used as criteria for establishing relationships between objects. As the social network has dynamic relationships, which can be modified through the acquisition of information from the owner of the object, it is possible to propagate information, events and news through the network to a greater number of users interested on that context.

A possibility proposed by the SIoT is to adopt the objects' social network as a tool to ascertain the trustworthiness of an object. The research performed in [7] attempts to construct a network within a medical environment so that objects can transmit and receive hospital system data securely, identifying the reliability level of objects and blocking attacks. Trustworthiness identified through the creation of a trustworthiness management protocol, which uses the relationships and information of the objects to determine their trustworthiness level. In [18] also is proposed a model that verifies the trustworthiness of objects in SIoT, being based on the collective work of objects that provide computational and storage operations. In this model, tasks are sent to objects and serving them, a reward is given to objects, thus helping to create relationships and establish trustworthiness between objects that cooperate frequently with each other.

The research developed in [15] presents a solution for the implementation of a SIoT architecture service discovery. Using the relationships built by the social network's objects it is possible to create a discovery method that with a small number of hops in a network implemented as graph, find objects which provide the required services. This is accomplished using two properties: the first is intrinsic to the social network, based on object friendships; the second is external to the social network, based on the similarity between the object and the request.

In [8] is proposed the creation of a method for service composition in SIoT using the relationships formed by the objects in the social network. The method encapsulates heterogeneous objects as web. Using the RESTful style, applications may require homogeneous access to objects, allowing collaboration between

them to compose more robust processes services. Composition is possible through the modeling of relationships between services over the social network, classifying relationships in three dimensions: location, type and correlation. When requesting a complex process, which needs to be composed of several services, the component performs a search throughout the social network using its three relationships to find the candidate services. These candidates are then evaluated and the best ones are compound into the required process. The work also proposes a service discovery algorithm used to find object services using the breadth-first search for graphs.

Through the works exposed in this section it is possible to notice that there are researches concerned with addressing the social capacity of objects, using its characteristics, information and the potential of their interactions to provide solutions for SIoT applications, especially within the context of smart environment. Likewise, there is a growing concern about creating SIoT functionalities that fit the use of social network, being the service discovery one of them. Therefore, this work presents the proposal of a SIoT service discovery method based on the information and characteristics, from social profiles of objects, to improve its efficiency. Differing from other solutions by restricting the need to sweep the entire network to find the appropriate services required.

3 SIoT Service Discovery

Social Internet of Things (SIoT) is a paradigm derived from the IoT that proposes the use of social networks in conjunction with the well-established IoT infrastructure. The use of the concept of social networks in IoT can lead to great advantages, such as: (i) SIoT structure can be shaped to ensure the navigability of the network, allowing efficient discovery of services and objects; (ii) the trustworthiness of objects can be determined by analyzing the interaction with their “friend” objects and models used to study social networks can be easily applied to address problems related to IoT [5]. The formalized concept of SIoT determines it as a social network where each node is a smart object capable of establishing relationships, autonomously, with other objects, following rules established by its owners [10].

A relationship in SIoT is the representation of a bond that objects can have in the real world, being this representation used for the construction of the social network through the formation of friend objects that are linked by relationships. The researches in SIoT establish five types of relationships that can exist between the objects [4, 5, 10]: (i) *Parental*: established between objects belonging to the same production batch, are usually homogeneous objects, originated in the same period and by the same manufacturer; (ii) *Co-location*: established between objects that are always used in the same place, in the case of sensors, actuators and objects of augmented reality are considered objects in the same environment as a house or smart city; (iii) *Co-work*: established when objects cooperate to provide certain functionality; (iv) *Ownership*: established between heterogeneous or homogeneous objects that belong to the same user; and (v) *Social*:

established when objects come in contact, continuous or sporadic, because their owners came in contact during some period of their lives.

SIoT objects are organized through a social network of objects, used in several processes of the paradigm, such as service composition and discovery. The social network is composed of representations of the physical existence of objects in the network, called social profiles. Each profile, contains attributes related to the object it represents, such as its identification, the friends formed by its relationships with other objects, the services it offers, its characteristics and information [11]. Characteristics of an object are the data inherent to its constitution and are generally immutable, for example its manufacturer, production batch, number of processing cores and primary and secondary memory, among others. While, the information of an object is data on the object that are constantly changing, such as storage capacity, battery status, object trustworthiness, security settings, among others [13].

The service discovery generally used in the literature, such as Chen et al. [8], uses only the relationships to search the network for objects that have services that satisfy the requests. Due to the peculiarity of such relationships, the search results may be different for the same objects. Figure 1 presents an example of how different relationships affect the discovery in SIoT networks. Each social profile is represented by a node and each edge identifies a relation of friendship between objects. The relationships *Ownership* and *Co-work* (Figs. 1(a) and (b), respectively) are used as examples. When a request is sent to the discovery component, it identifies the object that requested the service (requester) and makes it the initial search object. This is possible because any requester is an object that is part of the social network. In both figures, the requester is the object *D*. From the requester, the discovery method seeks to know if the services it offers meet the request. Each color represents the set of objects accessed at each search level. In the Fig. 1(a), from *D*, your friends, which are the objects *B*, *C*, *D* and *G*, are searched. Search continues on friends of previous objects (*A*, *F*, *H*, *I*, *L* and *O*), thus continuing until the discovery has covered all the objects in the network. Same process occur in Fig. 1(b), where the discovery is performed using a different relationship. In this case, it is possible to visualize that the result will be different, since there is no *Co-work* relationship between objects (*N*, *M*, *I*) and the rest of the network.

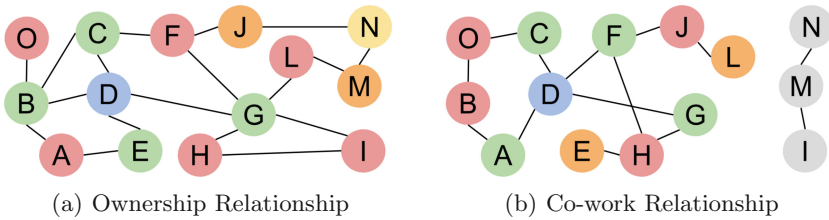


Fig. 1. Graphs representing SIoT network relationships. (Color figure online)

In both cases, the service discovery is performed by the breadth-first search for graphs, which checks all objects in the social network to find the services that best serve the request. This approach returns all the services that can fulfill the request, including several services that perform the same activities or do not meet the requirements of other SIoT network functionalities. Therefore, an additional step to filter services is required. Such step may compromise the performance of the application performing the request.

4 Social Profile Search - SPS

The SPS discovery method proposed in this work uses the characteristics and information contained in the social profiles of the objects as additional comparison criteria to verify if the services of an object, and itself, meet the request requirements. The SPS also introduces a new stopping condition for the breadth-first search, once all requested services are found, regardless of the number of objects visited, the search is interrupted. In addition, as in the breadth-first search, the SPS also ends its execution when all objects in the network have already been searched. Thus, the method ensures that a satisfying response to the request can be found in less time, balancing the quality of the services found and the search time, however leaving open the possibility of the worst case scenario, that happens when it is not possible to find in the network all the services to fulfill such request.

Figure 2 presents the request message that is received by the SPS, which is formed by three attributes: (i) Requester object (RO), containing the identification of the object that requested the search; (ii) Service List (SL), containing the required services that will be searched by SPS; and (iii) Requirements List (RL), containing requirements that will be verified and must be met by the characteristics and information of the objects, for instance, type of transmission technology, amount of memory, trustworthiness of the object, among others.



Fig. 2. Service discovery request and their attributes.

In Fig. 3 the SPS process is illustrated in a network with seven objects that receives a service discovery request. As any object that requires a service discovery is also part of the social network of objects, the SPS uses the requester social profile in the network as the initial object of the search. With the other attributes of the request, the SPS begins to verify if the friends of object *A* have the requested services (Fig. 3(b)). The initial object *A* is marked as not selected, because it either does not have the services sought or does not meet all imposed

requirements, even if it has the required service. When checking the objects *B* and *D*, SPS realizes that the two have required services and that they meet all the requirements, so their identifications are added to a list with the possible result. As all required services were not found the search is expanded to the objects *F* and *C* (Fig. 3(c)), that having the missing services and meeting the requirements, complete the execution of the search.

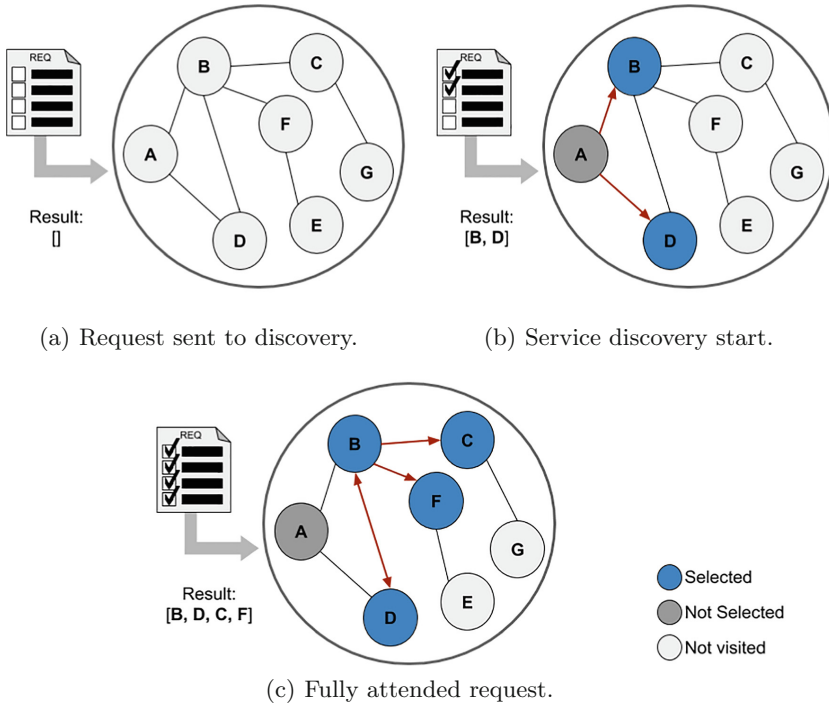


Fig. 3. Operation of the SPS method.

As an example we can consider an object that belongs to a SIoT network. In this network, the objects have as one of its characteristic, the types of wireless connections they can perform, such as bluetooth, 3G, GSM and Wi-Fi, being allowed to have any combination of these four connections. This object has only the ability to connect over Wi-Fi, while the discovery request calls for object services that can communicate over Wi-Fi and 3G, even if this object has a service that meets the request, it will not be considered since it does not meet the minimum discovery requirements. This type of approach facilitates the manipulation of the discovery results for other SIoT network functionalities, like the composition of services, since it guarantees that the services found do not have to undergo a new requirement check evaluation.

Algorithm 1 describes the SPS service discovery method considering the profiles of objects. The request is treated as three separate blocks, (i) required service

list (*required_service_list*) which represents the list containing the services must be found; (ii) requirement list (*requirement_list*) which contains all the minimum requirements that objects must attend; and requester (iii) (*requester*) which will be used as the starting point of the SIoT network discovery. In addition to the request, the algorithm also receives as input the social network (*social_network*), containing all social profiles of objects in the network.

Algorithm 1. Service Discovery Based on Social Profiles (SPS)

Input: *required_service_list*, *requirement_list*, *requester*, *social_network*

Output: *service_list*

Require: *requester* must be an object belonging to the social network

```

1: search_list.add(social_network.retrieve(requester))
2: while search_list is not empty do
3:   head ← search_list.removeHead()
4:   if required_service_list was met then
5:     break
6:   else
7:     if head.visualized = true then
8:       continue
9:     else if head met required_service_list and head met requirement_list then
10:      service_list.add(head.services())
11:    end if
12:  end if
13:  head.visualized ← true
14:  search_list.add(social_network.retrieveFriends(head))
15: end while
16: return service_list

```

Initially, the SPS adds the social profile of the requesting object to *search_list* and then initiates the loop by searching services over the entire network until the *required_service_list* has been completely met or *search_list* is empty. For each object removed from the list *search_list*, and added to *head*, is verified that the social profile has already been searched (line 7). If so, the next object is checked. If not, it is checked whether one or more of *head* correspond to the services required in *required_service_list* and if that object meets the minimum requirements established in *requirement_list* (line 9). Services that meet both conditions are added to *service_list* (line 10). The SPS then marks the social profile contained in *head* as already seen and add its friends to *search_list* (line 14), starting the process again. When the algorithm finishes its execution by one of the two stopping conditions, the found services are delivered as request response. The end result of the SPS allows three different states, (i) find no service that meets the request, (ii) find only some services, thus partially meeting the request, or (iii) find all services in order to meet the request.

5 Test Environment and Evaluation

The SPS method was implemented in Python and tested using the dataset of an urban SIoT network provided by [6]. which was created in the context of the SmartSantander project and executed in the cities of Belgrade, Guildford, Lübeck and Santander, however, the only available dataset is the one from the city of Santander.

The dataset consists of 16216 devices (objects of the social network) of various technologies. It provides information about social network relationships and the services that each object offers. The services are represented by numbers that indicate some service that the object can provide, and each object can have up to eight different types of services out of a total of sixteen. Because the dataset does not work with the services being unique entities, with their own identification, the SPS was implemented with a minor change, that instead of returning the services encountered, the objects that provide them are returned. The dataset presents object relationships through matrices with Boolean values, that represent the existence, or otherwise, of a friendship between objects. The ownership relationship was used to carry out the tests of this work. Figure 4 illustrates the organization of a relationship matrix where the number one identifies a friendship between two objects. For example, objects (A, C) and (E, F) are friends because the intersection of their columns and rows have one as value.

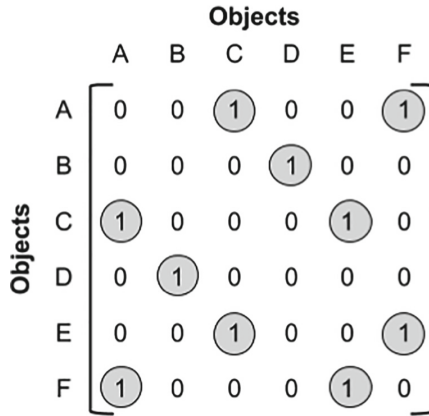


Fig. 4. Dataset relationship matrix.

To verify the efficiency of the SPS, two metrics were used: search time and solution scalability. The search time is the time that the algorithm takes to respond to the request, being calculated from the moment that the algorithm begins to consult the social network to find services until the moment the list of found objects is returned. Scalability evaluates the behavior of the solution as the number of objects in the network increases. The tests were carried out on a

machine with a i7-8750H hexa-core of 2666 MHz processor and 16 GB of DDR4 RAM memory.

The SPS was compared with the algorithm proposed by Chen et al. [8], that uses the breadth-first search method. To test the search time, a file was created containing all the requests that were made during the tests. The use of the file guarantees that the requests realized in the method SPS were identical to those realized in Chen. Each line of the file represents a request, containing the following information: the requesting object, the services to be found and the requirements that will be used by the search. This latter information is only used by the SPS method.

Each attribute of the request was generated randomly obeying certain minimum and maximum values. The requesting object is necessarily an object belonging to the network. The desired services are generated from a list of sixteen possible dataset services. Of those, four are chosen to form the desired services of each request, in way that an already chosen service can not be chosen again. To generate the requirements, it was necessary to first create the social profiles for the dataset objects, defining what characteristics and information would be associated with the profile and would become its attributes, these, can be observed in the Table 1. The next step was to fill in the attributes contained in the social profiles, following the allowed values in the Table 1. Finally, the requirements were generated and added to the request, which are based on the social profiles attributes. Random initial values were used for each requirement, respecting certain limits imposed in Table 1. In total, 2380 requisitions were created for the tests.

Table 1. Social profiles attributes and requisition requirements.

Attributes	Minimum value	Maximum value
Primary memory	256 MB	4 GB
Secondary memory	0 MB	512000 MB
Processing cores	1	8
Connectivity	1	5
Storage capacity	15%	75%
Battery status	10%	100%
Trustworthiness	45	95

The scalability test should check how the solution behaves by increasing the number of network objects. To accomplish this, it was necessary to divide the network into several samplings representing networks of varying sizes, consisting of five hundred, a thousand, two thousand, four thousand, twelve thousand and sixteen thousand objects. These samplings must obey the probability distribution of the object types present in the initial dataset. Table 2 displays the percentages for each type of device.

Table 2. Distribution of device types in dataset.

Device type	Percentage
Smartphone	22.45%
Personal computer	20.72%
Car	13.57%
Printer	13.07%
Tablet	9.87%
Smart fitness	5.43%
Parking	4.17%
Home sensors	3.70%
Street light	3.12%
Smartwatch	1.23%
Transportation	0.88%
Environment and weather sensors	0.86%
Point of interest	0.58%
Alarms	0.23%
Indicator	0.06%
Garbage truck	0.04%

Once it was guaranteed that the samplings had the same distribution as the original dataset, it was necessary to deal with another problem caused by the samplings. The randomness choice of objects makes it difficult for objects having relationships to be chosen together, especially for small-sized samplings. This situation impairs the navigability and makes it difficult to perform the search, since it is possible that there are several objects or groups of them isolated in the network. As a solution, it was necessary to manually add new relationships, so that all social profiles have at least one relationship with another object of the same sampling, thus guaranteeing the navigability of the network.

5.1 Results

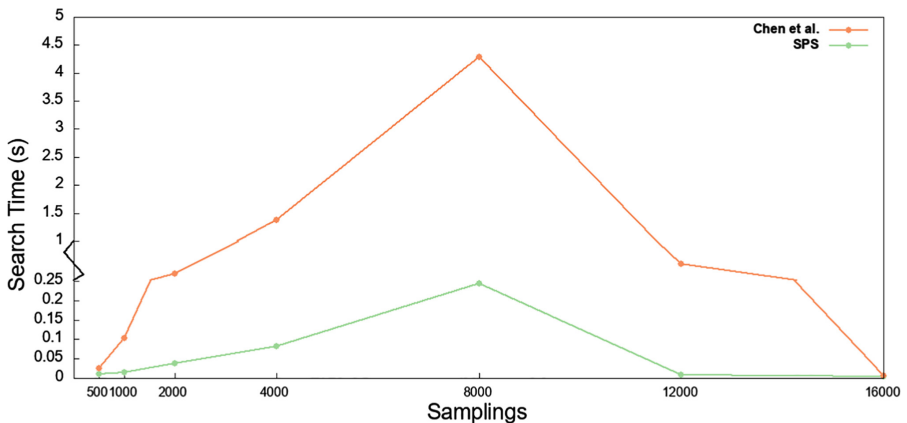
Ten executions were performed for each network sampling using the SPS and Chen algorithms. For each test the median search time of the 2380 requisitions was calculated, the median was used as a way to prevent outliers from impacting the results. Then, the mean of all the test medians of a sampling were calculated. The Table 3 displays the median of the search time, in seconds, of each sampling. Analyzing the values, it is possible to notice that the SPS algorithm can respond to the requests of all the samplings at a time significantly lower than the algorithm of Chen. This behavior can be seen mainly in sampling 4000 and 8000, where the SPS search time is less 93.98% and 94.32% than Chen, respectively.

Some surprises were observed during the comparison of the search times between the algorithms, being this comparison shown in Fig. 5. Among the 500

Table 3. Search time median, in seconds, for each request in the test samplings.

Samplings (n° of objects)	Chen (s)	SPS (s)
500	0,02594487	0,01072197
1000	0,10345836	0,01405903
2000	0,38549267	0,03829563
4000	1,37108796	0,08257948
8000	4,28259313	0,2434388
12000	0,56687746	0,00898465
16000	0,00587981	0,00399698

to 8000 samplings it is possible to notice that there is a growth in the average time needed for a request to be answered, and this behavior is already expected with the increase of objects in the social network. However, in the 12000 and 16000 samplings we have a totally reverse effect with an abrupt decrease in search time. For example, in the 12000 sampling, the Chen algorithm has a time reduction at the rate of 86.76%, while the SPS has a time reduction of 96.30%.

**Fig. 5.** Comparative graph of search time behavior in algorithms Chen et al. and SPS.

One of the possibilities for this behavior is that adding new objects has created a much more connected network, making it easier for the algorithms to reach objects that previously had few relationships and were more isolated. Another possibility is the randomness used to create relationships and to choose objects from the network have made its operation more complex. A great indication would be the fact that most of the changes made in the network were made in the smaller samplings, from 500 to 8000, while few changes were made at 12000 and even less at 16000. This being the case, these results can be a clear

testimony to the efficiency of SIoT solutions, because the smaller the change in the original network, provided by [6], the better its efficiency.

When comparing the algorithms' execution through the analysis of Fig. 5 it is easy to see how the SPS can keep the search time much lower than the algorithm Chen et al., being generally 94.03% faster and taking no more than 0.25 s, on average, to respond to a request. Another important aspect to be observed and analyzed is the algorithms' behavior regarding the network scalability. Obviously 16000 objects are still a very small number to test the scalability of these algorithms in a network that today already bears billions of objects. However, indications that the SPS algorithm is scalable can be observed through the controlled search time increase in relation to the expansion of the number of objects in the network and also of the expressive fall of it in larger samplings.

In contrast to SPS, Chen et al. has a search time increase well accented until the sampling 8000, where it then decrease to values very close to those of the SPS. Although there is a suggestion that Chen et al. is not scalable, the observed results show that it is necessary to verify this algorithm's behavior with an even larger number of objects per sampling to have a final verdict. However, it is possible to affirm that the solution proposed in this work through the SPS algorithm was more scalable than the method proposed by Chen et al. [8]. Other key information indicated by the tests presented in Fig. 5 is the need to verify how the organization of social profiles and their relationships affect service discovery algorithms in a SIoT network.

The Table 4 displays the average of objects returned in the search in each sampling. It is possible to observe that the number of objects returned by Chen is much larger than that of SPS (on average 99.09% more objects). Although Chen's method can always find the best services for each request, an additional step is still required to choose which objects, among those found, will be used to satisfy the requisition, which can impact the performance of the application that is performing such request. In SPS, this additional step is not required, since the method returns, as soon as it finds, the necessary objects to completely satisfy the request. Thus, in addition to a faster search time, the SPS shows that it can

Table 4. Average number of objects returned per request.

Sampling (n° of objects)	Chen	SPS
500	344	3
1000	679	5
2000	1340	8
4000	2459	17
8000	3502	37
12000	3789	42
16000	4102	39

more satisfactorily fulfill service discovery requests by taking into account the network social profiles of objects.

6 Conclusion

This work proposed a solution for a more efficient service discovery in a SIoT network, using the characteristics and information contained in the object social profile to attend the discovery requests more satisfactorily. For this purpose a service search method, called SPS, was proposed from the modification of a breadth-first search for graphs. The SPS adds a new stopping condition to improve discovery efficiency and inserts the use of the characteristics and information of the object social profile to verify the quality of the found services.

Simulated tests using the dataset of an urban SIoT network verified that the SPS algorithm performs a service search, 94.03% on average, faster than the method proposed in the literature. Similarly, the results of these tests also indicate a good adaptability of the SPS to deal not only with the increase in the number of objects in the network, but also with the complexity of its organization. The tests also indicated that there is a need to study how the relationships between objects are being formed in the social network, being one of the points that can lead to improvements in the efficiency and effectiveness of service discovery algorithms in SIoT.

As future works, new tests will be performed to determine the behavior of the SPS with more varied requests, containing a greater number of services required and also a greater variation of the characteristics presented. Another test that should be essentially performed is the verification of the behavior of service discovery algorithms when dealing with the different types of possible relationships in a SIoT network, identifying if there is a specific search order among these relationships that benefits the service discovery. In addition to the tests it is possible to focus efforts on the construction of other functionalities of the SIoT network that depend directly on the service discovery, as methods of social network management and service composition. Other future work can be done by identifying other service discovery strategies used in the Internet of Things and verifying the possibility of adapting them to the SIoT discovery, including methods based on artificial intelligence.

References

1. Alves, T.M., da Costa, C.A., da Rosa Righi, R., Barbosa, J.L.V.: Exploring the social Internet of Things concept in a university campus using NFC. In: 2015 Latin American Computing Conference (CLEI), pp. 1–12. IEEE, October 2015
2. Atzori, L., Carboni, D., Iera, A.: Smart things in the social loop: paradigms, technologies, and potentials. *Ad Hoc Netw.* **18**, 121–132 (2014)
3. Atzori, L., Iera, A., Morabito, G.: The Internet of Things: a survey. *Comput. Netw.* **54**(15), 2787–2805 (2010)
4. Atzori, L., Iera, A., Morabito, G.: SIoT: giving a social structure to the Internet of Things. *IEEE Commun. Lett.* **15**(11), 1193–1195 (2011)

5. Atzori, L., Iera, A., Morabito, G., Nitti, M.: The Social Internet of Things (SIoT)-when social networks meet the Internet of Things: concept, architecture and network characterization. *Computer Netw.* **56**(16), 3594–3608 (2012)
6. Atzori, L., Nitti, M., Marche, C.: Social Internet of Things (2016). <http://www.social-iot.org/index.php?p=downloads>. Accessed 28 Mar 2019
7. Brittes, M.P., Schneider Jr., B., Wille, E.C.: Trustworthiness management through social relationships in Internet of medical Things. *J. Commun. Inf. Syst.* **32**(1) (2017)
8. Chen, G., Huang, J., Cheng, B., Chen, J.: A social network based approach for IoT device management and service composition. In: 2015 IEEE World Congress on Services, pp. 1–8. IEEE, June 2015
9. Evans, D.: The Internet of Things: how the next evolution of the internet is changing everything. CISCO White Pap. **1**(2011), 1–11 (2011)
10. Girau, R., Martis, S., Atzori, L.: Lysis: a platform for IoT distributed applications over socially connected objects. *IEEE Internet Things J.* **4**(1), 40–51 (2017)
11. Jadhav, B., Patil, S.C.: Wireless home monitoring using social Internet of Things (SIoT). In: 2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT), pp. 925–929. IEEE, September 2016
12. Jarwar, M., Kibria, M., Ali, S., Chong, I.: Microservices in web objects enabled iot environment for enhancing reusability. *Sensors* **18**(2), 352 (2018)
13. Kim, J.E., Fan, X., Mosse, D.: Empowering end users for social Internet of Things. In: Proceedings of the Second International Conference on Internet-of-Things Design and Implementation, pp. 71–82. ACM, April 2017
14. Neves, A.R.D.M., Sarmanho, K.U., Nascimento Jr., F.C., Meiguins, B.S.: Iniciativa Smart Campus: um estudo de caso em progresso na Universidade Federal do Pará. In: Anais do I Workshop de Computação Urbana (COURB 2017), vol. 1, no. 1/2017. SBC, May 2017
15. Nitti, M., Pilloni, V., Giusto, D.D.: Searching the social Internet of Things by exploiting object similarity. In: 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), pp. 371–376. IEEE, December 2016
16. Santos, B.P., et al.: Internet das coisas: da teoria à prática. Minicursos SBRC-Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (2016)
17. Vodanovich, S., Sundaram, D., Myers, M.: Research commentary—digital natives and ubiquitous information systems. *Inf. Syst. Res.* **21**(4), 711–723 (2010)
18. Wang, K., Qi, X., Shu, L., Deng, D.J., Rodrigues, J.J.: Toward trustworthy crowdsourcing in the social Internet of Things. *IEEE Wirel. Commun.* **23**(5), 30–36 (2016)