ARTICLE IN PRESS
JID: CAEE [m3Gsc;October 23, 2017;12:54]

Computers and Electrical Engineering 000 (2017) 1–12

Contents lists available at ScienceDirect

Computers and Electrical Engineering

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

Mining Internet of Things for intelligent objects using genetic algorithm☆

Wail Mardini*, Yaser Khamayseh, Muneer Bani Yassein, Montaha Hani Khatatbeh

Computer Science Department, Jordan University of Science and Technology, Irbid, Jordan

A R T I C L E I N F O

Article history:
Received 31 March 2017
Revised 15 October 2017
Accepted 16 October 2017
Available online xxx

Keywords:
IoT
Social IoT (SIoT)
Friendship selection
Link selection
Searching IoT
Genetic algorithm

A B S T R A C T

The Internet of Things (IoT) is overpopulated by a large number of objects and millions of services and interactions. Therefore, the ability to search for the right object to provide a specific service is important. The merger of the IoT and social networking, the Social Internet of Things (SIoT), has made this possible. The main idea in the SIoT is that every object in the IoT can use its friends' or friends-of-friends' relationships to search for a specific service. However, this is usually a slow process because each node (object) is required to manage a large number of friends. This paper addresses the issue of link selection of friends and analyzes five strategies in the literature. Then it proposes a link selection strategy using the Genetic Algorithm (GA) to find the near optimal solution. The results show an improvement over the examined strategies in terms of several parameters.

1. Introduction

The Internet of Things (IoT) is considered the next evolution of the current global Internet [1]. The main idea is to increase its ability to gather, analyze, and distribute data and transform them into information, knowledge, and wisdom. However, it is not about connecting people. It is about connecting things, hence its name. It covers many possible application areas, and it enables objects to connect anytime, anywhere, and to anything.

In the IoT, a thing could be anything and everything, from a mobile device or a dishwasher to a controlling system of a car or a plane. It can be absolutely anything that moves or does not move. If it has an IP address, it is possible to connect it or track it. Thus, these things are not just smart phones and tablets; they are everything [2].

The IoT includes a vast number of objects that generate information about the physical world. This information can be obtained through standard Web browsers. In addition, the IoT can provide new services to end-users. However, in [3], the authors explained that the search of each service in the IoT is huge because the number of objects that connect to the network is continuously and rapidly increasing.

In addition, the traditional interaction model is based on the idea that humans are looking for information (human-object interaction). However, in the IoT, this model must change to object-object interaction, which means that an object will look for a service from other objects. In the literature, several models were proposed for real-time search [1,4]. However,

☆ Reviews processed and recommended for publication to the Editor-in-Chief by Guest Editor Dr. S. A. Aljawarneh.
* Corresponding author.
E-mail addresses: mardini@just.edu.jo (W. Mardini), yaser@just.edu.jo (Y. Khamayseh), masadeh@just.edu.jo (M.B. Yassein), kmmontaha14@cit.just.edu.jo (M.H. Khatatbeh).

https://doi.org/10.1016/j.compeleceng.2017.10.010
0045-7906/© 2017 Elsevier Ltd. All rights reserved.

these traditional models employ centralized systems for their engines; hence, they do not scale properly with the number of devices and queries. In order to overcome this shortage, a new approach based on the Social Internet of Things (SIoT) was proposed [4].

The SIoT can be used as an analog term for "social network of intelligent objects" [5]. Therefore, the SIoT can be thought of as the ability to have integration between the IoT and social networks in an intelligent way [4,6]. In the SIoT, objects will have the ability to search for a desired service using its friends' objects through available connections between them (i.e., friendship connections). As a result, each node will eventually have a large set of nodes (friendships) to manage, which will negatively affect the search time. Therefore, it is advisable to limit the number of friendships for each node. Moreover, choosing which friendships to keep will affect the search efficiency [7].

In the SIoT, every node is an object that can establish social relationships with other things in a predefined way, according to the rules that where set by the owner [6]. Many types of relationships exist [8]:

1. Parent-object relationship (POR).
2. Co-location object relationship (CLOR).
3. Co-work object relationship (CWOR).
4. Owner-object relationship (OOR).
5. Social-object relationship (SOR).

This paper addresses the issue of link selection of friends and analyzes five strategies in the literature for this purpose. It then proposes and implements a link selection strategy using the Genetic Algorithm (GA) to find the near optimal solution (near optimal link selection).

The rest of the paper is organized as follows. Section 2 discusses some works that are related to this topic. Section 3 evaluates the performance of some strategies that are proposed in the literature. Section 4 includes the authors' proposed GA for link selection, and Section 5 discusses sample performance results. Finally, Section 6 provides some conclusion notes.

2. Related work

A social approach for the IoT is expected to change the way nodes (i.e., objects) discover or search for other objects to gather information about the physical world [9,10,11].

The works in [1,2,6,12] are examples of existing approaches for service search in the IoT. In the above-mentioned approaches, the authors used a hierarchal structure of mediators to cope with the large number of objects. While these approaches work well with pseudo-static metadata, they are not scalable in the case of frequent network changes. The search for data from real-world entities and sensors is a major service in the IoT. However, two specific limitations exist. The first one is the frequent changes in the data of objects and the second is the large number of objects.

In [13], the authors proposed a probabilistic centralized system. In this approach, the contact to the objects is based on a prediction model that calculates the probability of matching the query. However, for good scalability with the number of objects, the search engine does not need to contact all the sensors. Some requests may not be required and will only increase traffic on the network.

In [14], the authors presented a method for a friend recommendation system in social networks using the GA. The proposed technique was mainly developed to analyze the importance of two major issues in friendship formation. The first issue is how links are formed in social networks and the second is why links are formed in social networks. The authors employed the idea of the GA to develop a friend recommendation system that aimed to optimize relationship preferences. Their approach resulted in a higher quality friends list that was relevant and appropriate for further future links (friendships).

In [15], the authors presented a design of policy language expression improvement (Ponder) for the SIoT. This improvement simplified the complexity of the general policy languages and was designed to provide easy training for users. Moreover, it was developed to be a policy editor in the SIoT environment. This research deployed a scene for the SIoT named Magic-Home. This scene has two management scenarios: access control and resources interactive. The Magic-Home framework consists of household appliances and three types of sensors (luminance, temperature, and humidity). In the scenario of access control, the authorization is given to the user before sending the control message to the IoT device. In the resources interactive scenario, the policymakers must define a policy for all possible actions that the platform processes when the sensor data are received. The performance measurement of the new IoT policy editor was conducted for three performance indicators: Simplicity, to make the training easy for users; Enforceability, which means the tools must be open source to easily allow for improvements; and Expressiveness, to process the policy requirement of the system. After the analysis of five major policy languages to develop a policy editor, the research selected Ponder. The performance of the new IoT policy editor was evaluated against the performance of a well-known language (PonderTalk). The obtained evaluation results show that the new editor achieved higher accuracy as well as higher spell and syntax error detection.

In [16], Jadhav and Patil designed a system for home monitoring in the SIoT that used appliances with the integration of social networks. The generated information in the system can be monitored using Facebook. In addition, sensors can receive instructions from Facebook comments. The presented system was developed using Arduino Uno board, which is connected to a set of sensors, such as smoke detection, intensity of light, detection of obstacle, and temperature sensor. These sensors are connected to the Raspberry Pi B2 unit, which is used as a gateway to connect the server with the Internet. The Raspberry

Pi connection to the server is used to exchange and sort data within the Facebook interface. The server receives interchange data from the sensor and posts it on Facebook. The user can then send commands by posting comments that the server, through keywords, converts to commands.

In [17], the authors proposed an agents-based architecture that aims to achieve better device-to-device and human-to-device communication. The idea is based on a set of rules to support the needs and the available services in the network. The authors used the TaskAgent in the SIoT architecture to represent *Applications* and to refer to some type of intelligence in this context. They demonstrated the architecture using the *Cultural Heritage* protection and preservation test that aims to detect hazardous conditions. They presented a simulator tool called *Agent Simulator for Social Smart Things* (ASSIST) and used it in their test case. The agents in this use case take advantage of the relationships resulting from friends of friends to discover services. The researchers used a threshold binding the maximum number of friends and demonstrated that ten friends were a good choice. Lower numbers could result in some loss of service discovery, but higher numbers slow down the discovery process.

A new Trust and Reputation model (T&R) was proposed in [18] to tackle security and trust issues in the SIoT communications against internal attacks. The authors defined *trust* as the interaction of personal experience and defined *reputation* as the interaction of social circles. They described malicious activities in the IoT by two main characteristics: malicious service provision and malicious recommendations. They assigned each characteristic to a 1-bit digit. For example, a benevolent behavior is represented (000). Then, the authors identified seven possible behaviors for each node. Finally, they constructed a mapping table to map behaviors to 3-bit characteristics codes. For example, the “hard to detect with Quality of Service (QoS) as a metric” behavior is mapped to 010 code. The trust and reputation values for each node is marked positive or negative based on the feedback from the other nodes. The mathematical model uses three variables: Satisfaction (s), which indicates the QoS; Weight (w), which identifies the importance of service for well-being; Fading factor (f), which diminishes the significance of older interactions. Moreover, the researchers proposed two models to calculate the reputation value: the Platform method and the Neighborhood method. In the Platform method, the reputation is calculated based on the society views, and it is used when the node cannot offer feedback from its social circle. In the Neighborhood method, the reputation value is calculated from the social circle of the node. The proposed model in [18] was evaluated using the TRMSim-WSN simulator. The results show that this model outperformed three well-known models (PeerTrust, Eigentrust, and PowerTrust) in several scenarios and achieved similar performance in the other scenarios.

A new paradigm, which is used to solve a number of issues for the development process of the SIoT, was proposed in [5]. The model in [5] tackles many issues, such as data transmission and sensing other objects. Many organizations and initiatives (ITU, EPCGlobal, and CASAGRAS) proposed solutions for the SIoT. However, it is important to propose protocols and solutions that ease and exploit things-related services. The Pachube platform is an example of a platform that resembles the behavior of a social network of objects. Pachube allows developers to transmit sensors' data to applications on the Web. The work in [5] focuses on both establishing and exploiting social relationships among things, not among their owners. The authors derived many relations from objects in the IoT by analyzing various relationships in social network services. Examples of possible relationships are co-location object relationship, co-work objects relationship, and social objects relationships.

In order to successfully implement the SIoT, the following components are required. First, an ID management component, which provides a universal identifier for all objects and includes IPV6, URL, and other protocols. Second, object profiling, which contains information about objects based on their features. Third, owner control rules and access control policies, which govern object operations; relationship management component; service discovery, which establishes and exploits social relationships among things, not among their owners; and a trustworthiness management component.

The authors proposed a three-layer server-side architecture with the following layers: the base layer, which encompasses the database for storage and management of data with relevant descriptors, ontologies database, semantic engines, and communications; the component layer, which hosts tools for basic and satellite component implementation; and the application layer, which includes interfaces to objects, humans, and third-party services. On the other hand, the authors also proposed a three-layer object architecture with the following layers: the object layer where the physical objects are located and are reached through their specific communication interfaces. The abstraction layer, which harmonizes communication between the different devices through common languages and procedures; and the social agent, which is devoted to communication among objects and with the SIoT servers to update the profile and friendships and to discover/request services from the social network. The service management is the interface of humans to control the object behavior.

The authors in [19] proposed an efficient automated demand-side management system for factories. The main goal of the proposed system was to minimize the customers' service costs. This encompassed a policy to minimize energy consumption. The proposed system interacts directly with the customer and the service company. In their model, the authors suggest the usage of rechargeable batteries to store power from renewable energy sources and whenever the energy prices are low. The role of the SIoT in this model is to simulate humans and their relations and to tie machines together. The authors use two types of relations: co-location relation and similarity relation. These relations are then applied to manage a group of machines in different areas (co-location relation) and a group of machines of the same type (similarity relation) autonomously. The energy consumption scheduling problem is formulated as an optimization problem with the objective function of minimizing energy costs to all machines at PAR. The Convex programming technique is used to solve the optimization problem. The outcome of this model is a minimized and cost-effective solution based on both the interaction between customers and the service company, as well as the interaction between the machines themselves.

Please cite this article as: W. Mardini et al., Mining Internet of Things for intelligent objects using genetic algorithm, Computers and Electrical Engineering (2017), https://doi.org/10.1016/j.compeleceng.2017.10.010

Traditional centralized models were also investigated for the search problem [20,21]. The authors in [20] utilized the Named Data Networking (NDN) architecture to propose a context-aware service discovery mechanism. The work in [20] combines Information-Centric Networking (ICN) concepts with the IoT by employing semantic capabilities for understanding and processing context information. In [21], the authors adopted a new mechanism to deal with the diverse nature of information sources by using well-defined basic requirements for context storage systems and analyses of context organizations models. Finally, they proposed a new context storage solution that improves scalability and semantic extraction and minimizes semantic ambiguity.

The approach that is investigated in this paper follows a different path to overcome the challenges associated with an increased number of information sources in the IoT. Indeed, the proposed model focuses on the decentralized aspects of the underlying architecture rather than on the centralized aspects.

3. Evaluation of previous techniques

This paper offers a twofold contribution. On one hand, it evaluates the performance of five different link selection strategies proposed in [6] using simulation. On the other hand, the paper proposes a solution for link selection in the SIoT using the GA. The performance of these five different strategies is evaluated using simulation in terms of the following parameters:

1. Average degree of connections: It measures the average number of direct friendships of each node.
2. Average path length: It measures the average connection length between any two nodes in the network.
3. Local cluster coefficients: It measures the spread of nodes and connections among the network.

The goal is to help the nodes in the social network in the process of selecting the best set of friends in order to improve the overall network performance. Each node will follow the same procedure until a certain condition is reached.

The initial steps in all strategies are the same. At the beginning, a node accepts all friendship requests. This is done until the node reaches the maximum number of friends allowed (N_{max}). Then, the node uses one of the following five strategies to manage any further requests [6,8].

First strategy: Simply reject all new requests after reaching N_{max} friends.
Second strategy: After receiving any new connection request, each node sorts its friends list in decreasing order of degree and accepts the first N_{max} friends.
Third strategy: It is similar to the second strategy, but the sort is in increasing order of degree. Then, the node accepts the first N_{max} friends.
Fourth strategy: After receiving any new connection request, each node sorts its friends list in decreasing order in terms of their common friends and accepts the first N_{max} friends.
Fifth strategy: It is similar to the fourth strategy, but the sort is in increasing order of common friends. Then, the node accepts on the first N_{max} friends.

The first strategy aims to simplify the selection process. The nodes will not perform any calculation and will not be biased toward any nodes with high or low friendship degrees. In the second and the third strategies, the nodes will choose, based on the nodes friendship degree, to enhance the average degree of connections (i.e., friendships). In the fourth and fifth strategies, the idea is to have better paths between nodes in terms of reliable and trustful information.

In this paper, the authors adopt a similar methodology as in [6,8] to study the effect of each strategy on the performance of the network. However, they need information regarding friendship requests, such as friend's profile, settings, and movements. This information is needed for all objects. Unfortunately, these data are not available to date; no real applications have been deployed to capture these pieces of information. Therefore, the authors adopted an alternative that is similar to the solution that was adopted in [8]:

1. Use a human social network and analyze it.
2. Build a social network of objects by extracting information from such network.
3. Use characteristics from the network in order to run a model that can generate synthetic networks with similar properties.
4. Finally, apply the strategies and analyze the results.

In order to implement Step 1, the Network Brightkite dataset, from the Stanford Large Network Dataset collection [22], has been used. It includes about 58k nodes and around 200k edges. The authors used a part of this network that has 2k nodes and 2k edges [6] in order to simplify the analysis.

The authors in [6] used Gephi (The open graph Viz platform) in order to visualize and analyze the social network. Gephi is a well-known visualization and exploration software for all kinds of graphs and networks. In addition, Gephi is an open-source software package written in Java on the NetBeans platform. For the visualization process, the format of the dataset obtained from Brightkite does not match the format used by Gephi. The information of the first 2000 nodes from the dataset was imported from Brightkite and then normalized to the standard format of Gephi. The selected social network is shown in Fig. 1.

The above-mentioned strategies were implemented under this Gephi environment. The code was written in Java. The dataset of the graph was imported to NetBeans. Then, the authors used the Filtering plugin to apply the strategies and

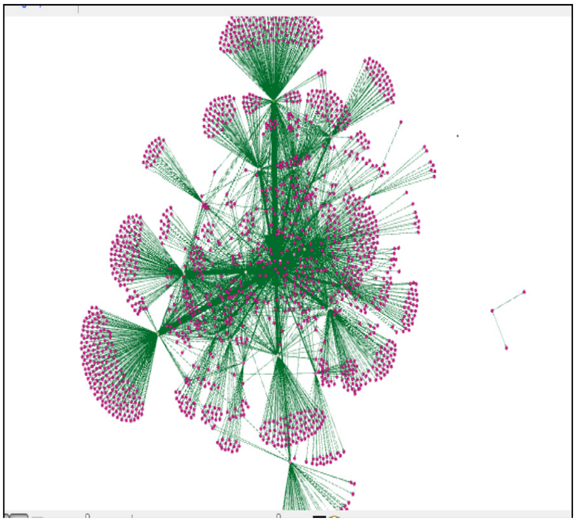

Fig. 1. The main graph.

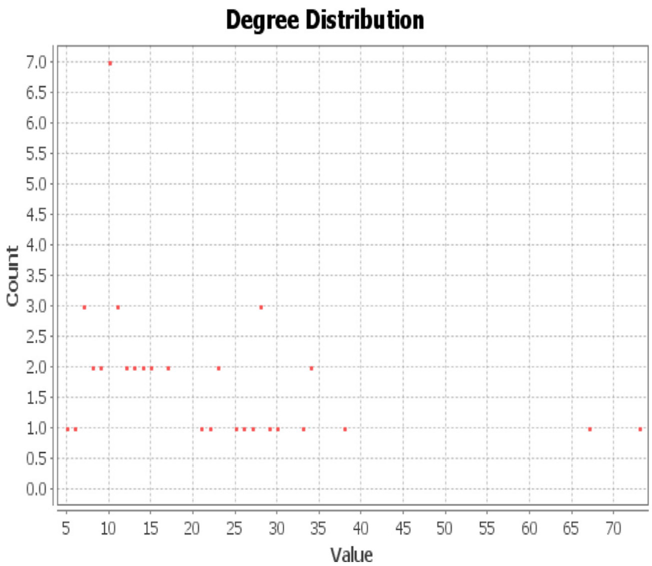

Fig. 2. Degree distribution - Strategy 2.

calculated the performance measures (i.e., average degree for connections for nodes, average connection lengths, and local clustering coefficient).

Figs. 2–5 show sample results for the implementation of Strategies 1–5. More in detail, Fig. 2 shows sample results for degree distribution for Strategy 2 as a single run. It illustrates that the node degrees distribution ranges between 5 and 40 after applying the second strategy.

Figs. 3–5 show average results for all strategies, repeated 10 times for the three performance measures. Fig. 3 highlights that the average degree for Strategy 1, where first N_{max} connections achieves the highest average degree, and the maximum neighborhood degree achieves the lowest values. This could be due to the fact that nodes kept the first few maximum degree nodes neighbors, which helped them in later requests. Indeed, it will not be necessary to have many nodes with lower degrees to satisfy the new requests.

Fig. 4 illustrates the average path length that was calculated based on the local neighboring information. The strategy, which is based on minimum local clustering and designed to have more information about the neighboring nodes, achieves the lowest values. The minimum neighbor degree achieves the worst values. Noticeably, when no limit is imposed on the connections, the lowest value is 8.

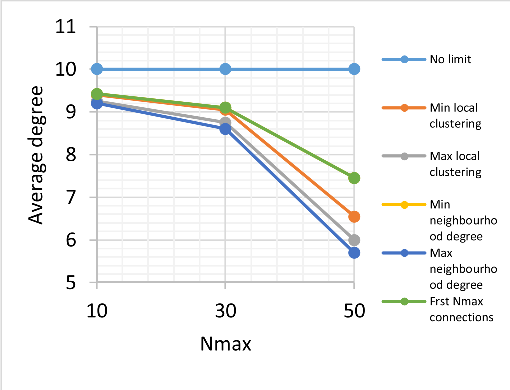

Fig. 3. Average degree - Strategies 1–5.

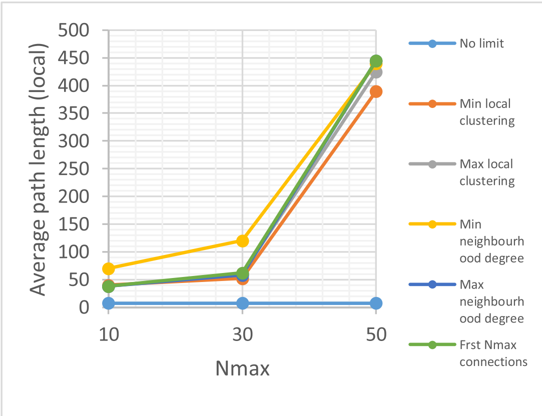

Fig. 4. Average path length (local information) - Strategies 1–5.

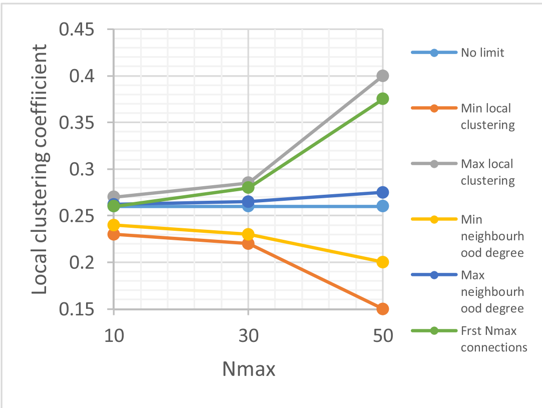

Fig. 5. Local clustering coefficient - Strategies 1–5.

Fig. 5 shows the local clustering coefficient for all strategies. As Strategy 4 was designed to have fewer common friends, it achieves the lowest values because fewer common friends are kept and used for future request. On the other hand, Strategy 5 achieves the highest values. The value of 0.27 is obtained when no limit on the number of connections is imposed.

4. Proposed genetic algorithm for link selection

The authors used the GA to find a near optimal solution of the link selection problem in the SIoT. The GA is one of the most robust search algorithms for complex and ill-behaved optimization problems. It is widely and efficiently used to

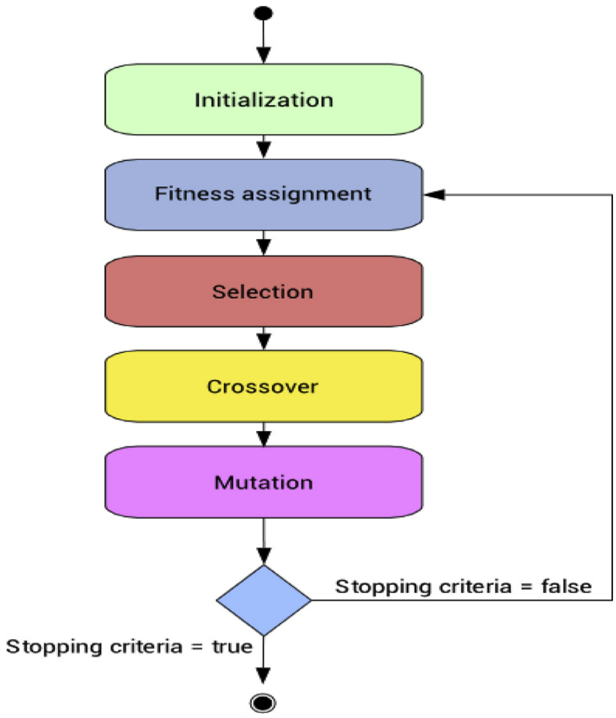

Fig. 6. The obtained graph when applying the GA [23].

solve many problems in the networking area. The GA was used in developing different algorithms in similar areas [23,25]. The main idea of the GA is to deal with available solutions (i.e., candidates) and improve them through a set of steps (i.e., iterations). The basic steps include selection, crossover, and mutation of the solutions.

The main steps for the GA are illustrated in Fig. 6. There are a number of parameters and common operations that should be set and defined before starting any GA. For example, it is necessary to define the initial population, fitness function, gene, chromosome, crossover, mutation, and repetition count.

Thus, the parameters and operations for the GA are defined next.

1. The size of the population is 1000 nodes.
2. Fitness Function: The following formula was used in order to get the maximum number of friends and friends of friends for each graph:

$$F(x) = \frac{Number\ of\ Common\ Friends}{Number\ of\ Friends_S\ X\ Number\ of\ Friends_R} \quad (1)$$

   where Number of Friends_S is the number of friends for the sending node and Number of Friends_R is the number of friends for the receiving node.
3. Gene: It represents the digit of node label.
4. Chromosome: It represents the node itself.
5. Crossover operation: Selecting pair of chromosome based on the value of their fitness function; selecting the chromosomes with the maximum fitness value.
6. Mutation operation: Making change in the gene by selecting a gene randomly in a chromosome.
7. G: The count of generation (termination criteria). The process is repeated 10 times.

The application of these steps to the proposed link selection algorithm is explained below:

1. Generate the initial population from the graph, selecting about 1k (1000) nodes.
2. Calculate the fitness function for each node in the initial population using Eq. (1). The main characteristic of the fitness function is that it calculates the number of friends and number of common friends for each node (i.e., max degree and max cluster coefficient). When the maximum value of this fitness function is obtained, it is possible to collect a number of nodes that have a high number of friends and common friends. This leads to a link selection that contains the node with the required service (i.e., the node asks its friends or common friends for a service instead of searching for a service in the whole graph).
3. Perform the crossover operation on two randomly selected nodes (with a maximum value of fitness function)

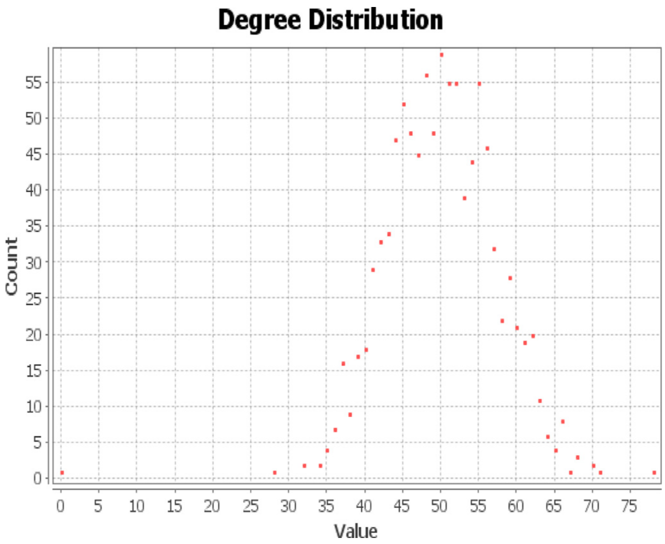

Fig. 7. Degree distribution for the GA.

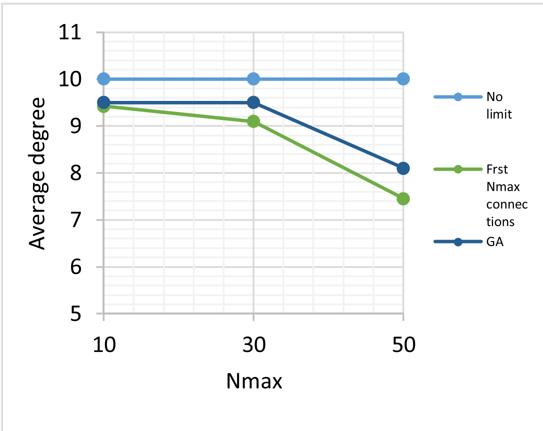

Fig. 8. Average degree distribution with the GA.

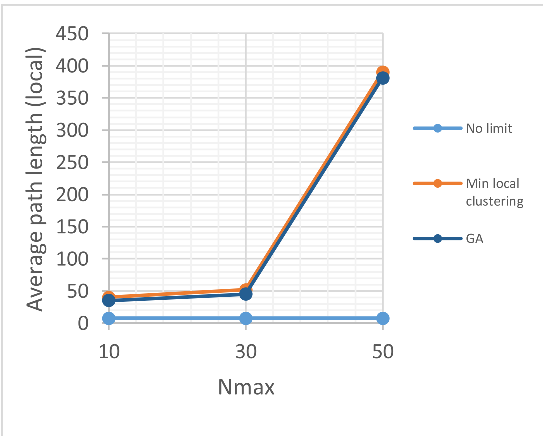

Fig. 9. Average path length with the GA.

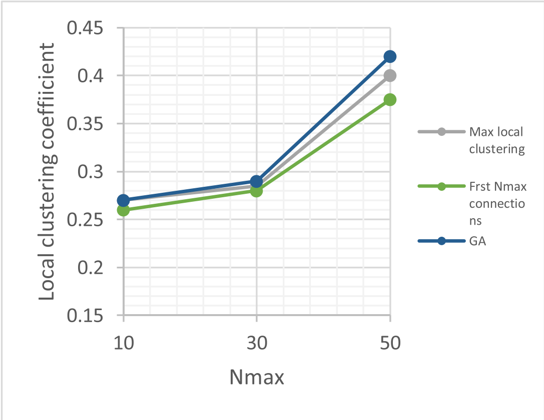

Fig. 10. Local clustering coefficient with the GA.

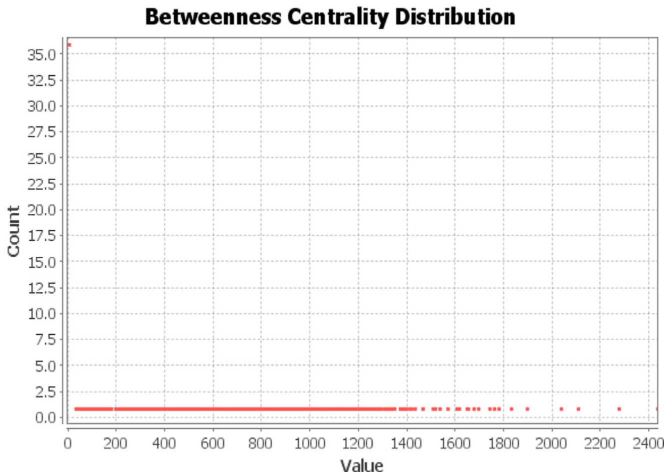

Fig. 11. Betweenness distribution when applying GA.

4. Perform the mutation operation by changing one digit (i.e., gene) in the node (i.e., chromosome). Place the new nodes in the new population.
5. Repeat this process a number of times in order to get the optimal solution.

5. Simulation results and discussion

The authors used the same environment that is illustrated in Section 3 to implement the GA of Section 4. The results are discussed below.

Fig. 7 outlines the results for degree distribution for a single experiment using GA. The figure shows that most of the nodes got degree of values between 35 and 65, which will aid in better service discovery.

Figs. 8–10 compare the two best results from Strategies 1–5, which were shown previously in Figs. 3–5 with the GA results. In Fig. 8, the average degree for the GA approach gives better results than all the strategies presented before. Please note the max value with no limit is 10.

Fig. 9 shows the average path length for the GA compared with the best (lowest) strategy before, which is the minimum clustering coefficient strategy. Both results are approximately the same, with very little improvement for the authors' GA approach. However, in Fig. 10 the GA approach gives much better results in terms of the clustering coefficient value, which is the maximum clustering approach. In order to evaluate the results of the authors' approach and measure the quality of these paths created by the GA, the betweenness measure is discussed below.

Betweenness gives an indication about the quality of the shortest paths [24]. It has been used in many studies as a measure for the connectivity factor between the nodes. For example, in [24] the authors used betweenness to find energy-efficient routes using fuzzy logic and ant colony optimization. In [20], the authors relied on the fact that the users in an Online Social Network (OSN) usually use their direct friends in their main communication behavior to propose an energy-efficient cross-layer design for wireless mesh network with content sharing in OSN. Thus, Fig. 11 highlights betweenness

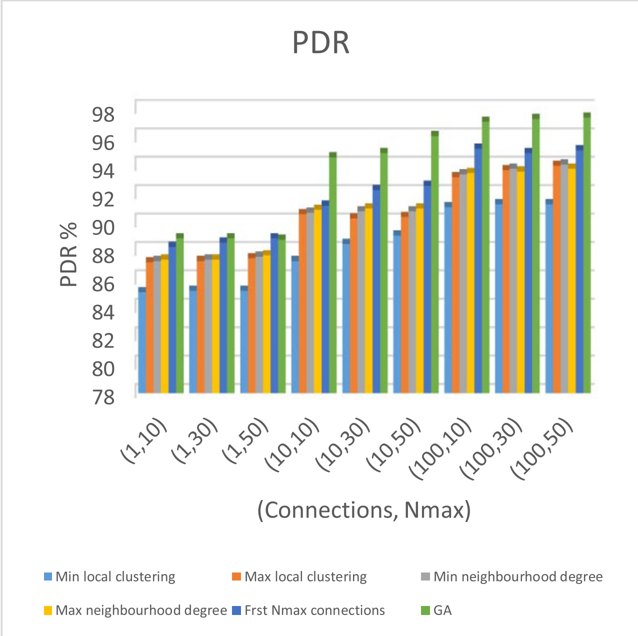

Fig. 12. PDR results.

results for the GA. Fig. 11 shows that most nodes have a betweenness value of 1 (other nodes' betweennesses are 0), which means that all nodes are equally important on the shortest paths among all pairs of nodes in the networks.

In order to measure the performance of the proposed scheme in delivering the requested services successfully, the authors conducted a set of simulation experiments and measured the Packet Delivery Ratio (PDR) for different schemes. The average of 10 runs was computed. Fig. 12 depicts the obtained PDR ratio for six different schemes. The experiments were conducted for three Nmax values (10, 30, and 50) for different request loads. Requests were assumed to have fixed size, and the number of load per node was assumed to follow exponential distribution with different average (1, 10, and 100). The results show the superiority of the GA algorithm in successfully delivering the requested services by the users.

6. Conclusions

Object search in IoT is considered an important issue due to its large and complicated search space. This complication rises from the fact that every object in the IoT can use its friends' or friends-of-friends' relationships to search for a specific service. The proposed Genetic algorithm based technique was introduced to overcome the limitations of some of the art of the state algorithms. The paper first discussed five heuristic search functions introduced in the literature. A new genetic algorithm based search algorithm is then introduced to find the near optimal solution (near optimal link selection).

The proposed strategy for the link selection in the SIoT achieves better results in terms of average degree and average cluster coefficient. However, the authors' strategy gives a slight enhancement in terms of the average shortest path length.

For future work, the authors recommend designing and implementing a hybrid GA fitness function to overcome the shortcoming of the traditional function. In the suggested function, the optimal solution would be the chromosomes with the shortest path and the maximum cluster coefficient.

References

[1] Yap K-K, Vikram S, Mehul M. MAX: human-centric search of the physical world. In: Proceedings of the 3rd international conference on embedded networked sensor systems. ACM; 2005. p. 166–79.
[2] Zhang D, Laurence TY, Hongyu H. Searching in internet of things: vision and challenges. In: Proceedings of the 2011 IEEE 9th international symposium on parallel and distributed processing with applications (ISPA). IEEE; 2011. p. 201–6.
[3] Geetha S. Social internet of things. World Sci News 2016;41:76.
[4] Nitti M, Luigi A, Irena Pletikosa C. Friendship selection in the social internet of things: challenges and possible strategies. IEEE Internet Things J 2015;2(3):240–7.
[5] Atzori L, Antonio I, Giacomo M. Siot: Giving a social structure to the internet of things. IEEE Commun Lett 2011;15(11):1193–5.
[6] Nitti M, Luigi A, Irena Pletikosa C. Network navigability in the social internet of things. In: Proceedings of the 2014 IEEE world forum on internet of things (WF-IoT). IEEE; 2014. p. 405–10.
[7] Ortiz AM, Dina H, Soochang P, Son NH, Noel C. The cluster between internet of things and social networks: review and research challenges. IEEE Internet Things J 2014;1(3):206–15.
[8] Atzori L, Antonio I, Giacomo M. The internet of things: a survey. Comput Netw 2010;54(15):2787–805.
[9] Mendes P. Social-driven internet of connected objects. In: Proceedings of the interconn. smart objects with the internet workshop; 2011.

[10] Weber RH, Romana W. Internet of things, Vol. 12. New York, NY, USA: Springer; 2010.
[11] Nitti M, Luigi A. What the SIoT needs: a new caching system or new friendship selection mechanism?. In: Proceedings of the 2015 IEEE 2nd world forum on internet of things (WF-IoT). IEEE; 2015. p. 424–9.
[12] Wang H, Chiu CT, Qun L. Snoogle: A search engine for pervasive environments.. IEEE Trans Parallel Distrib Syst 2010;21(8):1188–202.
[13] Ostermaier B, Kay R, Friedemann M, Michael F, Wolfgang K. A real-time search engine for the web of things. In: In internet of things (IOT). IEEE; 2010. p. 1–8. 2010.
[14] Naruchitparames J, Mehmet Hadi G, Sushil JL. Friend recommendations in social networks using genetic algorithms and network topology. In: Proceedings of the 2011 IEEE congress on evolutionary computation (CEC). IEEE; 2011. p. 2207–14.
[15] He L, Xiaofeng Q, Yi W, Teng G. Design of policy language expression in SIoT. In: Proceedings of the 2013 22nd wireless and optical communication conference (WOCC). IEEE; 2013. p. 321–6.
[16] Jadhav B, Patil SC. Wireless Home monitoring using social internet of things (SIoT). In: Proceedings of the international conference on automatic control and dynamic optimization techniques (ICACDOT). IEEE; 2016. p. 925–9.
[17] Kasnesis P, Lazaros T, Dimitris K, Charalampos ZP, Iakovos SV. ASSIST: An agent-based SIoT simulator. In: Proceedings of the 2016 IEEE 3rd world forum on internet of things (WF-IoT). IEEE; 2016. p. 353–8.
[18] Kokoris-Kogias E, Orfefs V, Theodora V. TRM-SIoT: a scalable hybrid trust & reputation model for the social internet of things. In: Proceedings of the 2016 IEEE 21st international conference on emerging technologies and factory automation (ETFA). IEEE; 2016. p. 1–9.
[19] Eom J, Laihyuk P, Woongsoo N, Nhu-Ngoc D, Seon Min J, Yong HK, et al. Using social internet of things (SIoT) demand side management on the plant. In: Proceedings of the 2016 eighth international conference on ubiquitous and future networks (ICUFN). IEEE; 2016. p. 685–7.
[20] Hu J, Lie-Liang Y, Lajos H. Energy-efficient cross-layer design of wireless mesh networks for content sharing in online social networks. IEEE Trans Veh Technol 2017.
[21] José Q, Mário A, Daniel C, Diogo G, Rui LA. On the application of contextual IoT service discovery in information centric networks. Comput Commun 2016;89:117–27 ISSN 0140-3664.
[22] Leskovec, J, and Andrej K. {SNAP Datasets}:{Stanford} large network dataset collection. (2015).
[23] Lai C-C, Ting C-K, Ko R-S. An effective genetic algorithm to improve wireless sensor network lifetime for large-scale surveillance applications. In: Proceedings of the 2007 IEEE congress on evolutionary computation (CEC). IEEE; 2007. p. 3531–8.
[24] Jain A. Betweenness centrality based connectivity aware routing algorithm for prolonging network lifetime in wireless sensor networks. Wireless Netw 2016;22(5):1605–24.
[25] Kim M, Ko I-Y. An efficient resource allocation approach based on a genetic algorithm for composite services in IoT environments. In: Proceedings of the 2015 IEEE international conference on web services (ICWS). IEEE; 2015. p. 543–50.

Wail Mardini is an Associate professor of Computer Science at Jordan University of Science and Technology (JUST) in Jordan since 2006. He received his Master and Ph.D. degree in Computer Science from University of New Brunswick/Canada at 2001 and University of Ottawa/Canada at 2006, respectively. Wail is currently working on Wireless Mesh Networks, Wireless Sensor Networks, and IoT.

Yaser Khamayseh is an Associate professor of Computer Science at Jordan University of Science and Technology (JUST) since 2007. He received his master's and Ph.D. degree in computer science from University of New Brunswick/Canada in 2001 and from University of Alberta/ Canada in 2007, respectively. His research interests include simulation and modeling, wireless networks, performance evaluation, security, and next generation Internet.

Muneer Masadeh Bani Yassein received his M. Sc. And Ph.D. degree in Computer Science from Al-bayt University/Jordan in 2001 and from the University of Glasgow/U.K. in 2007, respectivly. He is an Associate professor in the Department of Computer science at Jordan University of Science and Technology (JUST) since 2007. Muneer is currently conducting research in Mobile Ad hoc Networks, Wireless Sensors Networks, and Cloud Computing.

Montaha Hani Khatatbeh received her bachelor in Computer Engineering from Jordan University of Science and Technology (JUST) in 2014. She also received her master degree in Computer Engineering from JUST in 2017. Her research interests includes Wireless Sensor Networks and Internet of Things.

Please cite this article as: W. Mardini et al., Mining Internet of Things for intelligent objects using genetic algorithm, Computers and Electrical Engineering (2017), https://doi.org/10.1016/j.compeleceng.2017.10.010