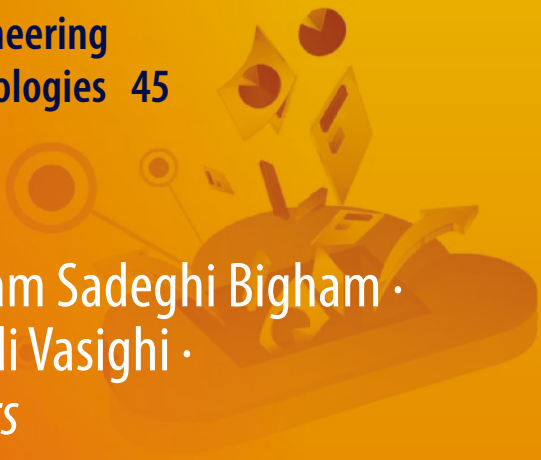
Lecture Notes on Data Engineering
and Communications Technologies 45



Mahdi Bohlouli · Bahram Sadeghi Bigham ·
Zahra Narimani · Mahdi Vasighi ·
Ebrahim Ansari *Editors*

Data Science: From Research to Application

Springer

Lecture Notes on Data Engineering and Communications Technologies

Volume 45

The aim of the book series is to present cutting edge engineering approaches to data technologies and communications. It will publish latest advances on the engineering task of building and deploying distributed, scalable and reliable data infrastructures and communication systems.

The series will have a prominent applied focus on data technologies and communications with aim to promote the bridging from fundamental research on data science and networking to data engineering and communications that lead to industry products, business knowledge and standardisation.

**** Indexing: The books of this series are submitted to SCOPUS, ISI Proceedings, MetaPress, Springerlink and DBLP ****

More information about this series at http://www.springer.com/series/15362

Mahdi Bohlouli · Bahram Sadeghi Bigham ·
Zahra Narimani · Mahdi Vasighi ·
Ebrahim Ansari
Editors

Data Science: From Research to Application

Springer

Editors
Mahdi Bohlouli
Institute for Advanced Studies
in Basic Science
Zanjan, Iran

Bahram Sadeghi Bigham
Institute for Advanced Studies
in Basic Science
Zanjan, Iran

Zahra Narimani
Institute for Advanced Studies
in Basic Science
Zanjan, Iran

Mahdi Vasighi
Institute for Advanced Studies
in Basic Science
Zanjan, Iran

Ebrahim Ansari
Institute for Advanced Studies
in Basic Science
Zanjan, Iran

ISSN 2367-4512 ISSN 2367-4520 (electronic)
Lecture Notes on Data Engineering and Communications Technologies
ISBN 978-3-030-37308-5 ISBN 978-3-030-37309-2 (eBook)
https://doi.org/10.1007/978-3-030-37309-2

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Data science is a rapidly growing field and as a profession incorporates a wide variety of areas from statistics, mathematics and machine learning to applied big data analytics. This is not limited to computer science, but also physics, astronomy, medicine, labour market analysis, marketing and many more. For instance, the Large Synoptic Survey Telescope (LSST) is planned to deliver 100 petabytes of data in the next decade12. This demands awareness of emerging technologies and calls for technologies and services of tomorrow, especially big data and data science expertise. According to the Forbes, "Data Science" is listed as LinkedIn's fastest growing job in 2017. The need to have professional data scientists is not limited to just education, but also proper development and preparation of environments and tools to brainstorm on recent research and scientific achievements of the area.

Such a scientific need for having an event that scientists can share their first-class data science achievements was known for us since years ago, and we tried in this regard to put special interest and focus on data science in our past executions of CICIS conference. This year, in its fifth run, we decided to dedicate the conference to data science area and accordingly to even keep it as professional data science event in the future. The 5th international conference on Contemporary issues in Data Science (CiDaS) has provided a sort of a real workshop (not listen-shop) to scientists and scholars to share ideas, initiative future collaborations and brainstorm challenges as well as industries to catch emerging solutions from the science to their real data science problems. In this regard, we tried hard in the frame of CiDaS 2019 to support all scientists and involve them in this move towards the successful future. In addition, we received special general interest from most of academics and industries in Iran and abroad by submitting significant number of manuscripts.

In particular, we received submissions from ten different countries and have tried to deliver at least two constructive reviews per submission. The acceptance rate of full paper submissions in CiDaS 2019 was about 30%. Furthermore, CiDaS 2019 has had significant number of national and international sponsors. Current chapters of this book are accepted papers of the conference. In addition, we also had scholarly well-known keynote speakers, who covered wide range of data science topics from academic and industrial points of views. By having over 55 experts as

scientific committee members from over 15 countries, we provided multinational and context-aware reviews to our audience, which also improved the quality of accepted papers as well.

We believe that we will be able to support all data scientists from various areas in our future events and involve them in this move towards the successful future and welcome your support. We hope that you will enjoy our future iterations of CiDaS. If you find our work interesting for you and your field, we always welcome collaborations and supports in this scientific event.

Sincerely Yours,

Mahdi Bohlouli
Bahram Sadeghi Bigham
Zahra Narimani
Mahdi Vasighi
Ebrahim Ansari
CiDaS 2019 Steering Committee

Acknowledgement

We would like to appreciate all scientific supports of following scholars through their reviews and constructive feedback:

- Hassan Abolhassani, Software Engineer, Google, USA
- Mohsen Afsharchi, University of Zanjan, Iran
- Sadegh Aliakbary, Shahid Beheshti University, Iran
- Ali Amiri, University of Zanjan, Iran
- Morteza AnaLoui, Iran University of Science and Technology, Iran
- Amin Anjomshoa, Massachusetts Institute of Technology, USA
- Lefteris Angelis, Aristotle University of Thessaloniki, Greece
- Nikos Askitas, Research Data Center, Institute of Labour Economics, Germany
- Zeinab Bahmani, Uni-Select Inc., Canada
- Davide Ballabio, University of Milano-Bicocca, Italy
- Markus Bick, ESCP Europe Business School, Germany
- Elnaz Bigdeli, University of Ottawa, Canada
- Mansoor Davoodi Monfared, Institute of Advanced Studies in Basic Sciences, Iran
- Mohammad Reza Faraji, Institute of Advanced Studies in Basic Sciences, Iran
- Agata Filipowska, Poznan University of Economics and Business, Poland
- Holger Fröhlich, University of Bonn, Germany
- George Kakarontzas, Technical Educational Institute of Thessaly, Greece
- Alireza Khastan, Institute of Advanced Studies in Basic Sciences, Iran
- Antonio Liotta, University of Derby, UK
- Rahim Mahmoudvand, Bu-Ali Sina University, Iran
- Samaneh Mazaheri, University of Ontario Institute of Technology, Canada
- Federico Marini, University of Rome "La Sapienza", Italy
- Maryam Mehri Dehnavi, University of Toronto, Canada
- Nima Mirbakhsh, Arcane Inc., Canada
- Ali Movaghar, Sharif University of Technology, Iran
- Ehsan Nedaaee Oskoee, Institute of Advanced Studies in Basic Sciences, Iran
- Peyman Pahlevani, Institute of Advanced Studies in Basic Sciences, Iran

- Paurush Praveen, Machine Learning Research, CluePoints, Belgium
- Edy Portmann, University of Fribourg, Switzerland
- Masoud Rahgozar, University of Tehran, Iran
- Shahram Rahimi, Southern Illinois University, USA
- Reinhard Rapp, Hochschule Magdeburg, Germany
- Mohammad Saraee, University of Salford, Manchester, UK
- Frank Schulz, SAP AG, Germany
- Mehdi Sheikhalishahi, InnoTec21 GmbH, Germany
- Ioannis Stamelos, Aristotle University of Thessaloniki, Greece
- Athena Vakali, Aristotle University of Thessaloniki, Greece

Contents

Efficient Cluster Head Selection Using the Non-linear Programming Method for Wireless Sensor Networks

Maryam Afshoon[1], Amin Keshavazi[1(✉)], Tajedin Darikvand[2], and Mahdi Bohlouli[3]

[1] Department of Computer Engineering, Marvdasht branch, Islamic Azad University, Marvdasht, Iran
Keshavarzi@miau.ac.ir
[2] Department of Mathematic, Marvdasht branch, Islamic Azad University, Marvdasht, Iran
[3] Department of Computer Science, Institute for Advanced Studies in Basic Sciences, Zanjan, Iran

Abstract. Wireless sensor networks consist of thousands of sensor nodes that are battery-based and have a limited lifetime. Accordingly, performance and energy efficiency are big challenges in wireless sensor networks. In this regard, numerous techniques were studied and developed to reduce energy consumption. In this paper, a mathematical-based method has been proposed for the optimal selecting of the cluster head in wireless sensor networks. In the proposed algorithm, a node was selected as a cluster head that has the maximum energy, weight, and density, as well as the lowest total distance from the other nodes. In this respect, the problem was converted into a math function, which was solved by non-linear programming. The experiment results show that the presented algorithm is efficient, as compared with the other approaches that have, hitherto, been used to solve this problem.

Keywords: Wireless sensor networks · Mathematical modeling · Non-linear programming · Clustering · Cluster head selection

1 Introduction

Wireless sensor networks are a combination of hundreds or thousands of battery-based sensor nodes and are a subset of the distributed systems. The battery of these nodes is limited and non-chargeable. One of the traditional methods for the energy efficiency of data transfer in these nodes is clustering [1]. The clustering process divides a geographical area into smaller parts and allocates a node as a head cluster to each part. Selecting a head cluster, which changes in each round, plays a critical role in data transfer energy efficiency [2]. The number of head clusters and the number of member nodes in each cluster can be constant or variable in the network. Also, nodes can directly send their data to the base station [3]. Wireless sensor networks have big challenges, including routing [3] and topology control [4]. The most important challenge of wireless sensor networks is energy efficiency [9]. Clustering is one of the most

M. Bohlouli et al. (Eds.): CiDaS 2019, LNDECT 45, pp. 1–12, 2020.
https://doi.org/10.1007/978-3-030-37309-2_1

important acts to handle this issue. Sensor nodes collect information from the surroundings and send it to the base station. Now, should all of the nodes send data simultaneously, problems like congestion, band width loss and error increase will occur, leading to an energy loss and therefore decreasing the network lifetime. To prevent these problems, for each cluster, a node must be selected as a head cluster node whose sensor nodes can separately send the collected information from the surroundings to their head cluster node and the head cluster node can send this information to the base station after collection and compression. The existence of the head cluster node is to decrease the number of connections in the network, leading to a reduction in the energy consumption and an increase in the network lifetime. The energy or battery lifetime, the sum of distances and the density are the key factors in head cluster selection. The algorithm chosen for clustering and head cluster selection is so effective in network energy consumption. In this paper, we have attempted to present a new method that works for clustering and head cluster selection based on mathematical optimization. The results of the simulation at the end of the paper show that this method has better outputs than the former methods in the same field; in the other words, by using this method, the networks lifetime will increase incredibly in comparison with the other methods.

Section 2 presents a literature review. Section 3 illustrates the non-linear programming method for cluster head selection in wireless sensor networks. Section 4 provides the simulation configurations. Section 5 depicts the evaluation results, and Section 6 concludes the paper.

2 Related Works

As it was said in the previous section, various algorithms are introduced for cluster head selection. The most famous algorithms in this field are:

The HEED [5] is a distributed protocol that selects the clusters independently of how the nodes are distributed based on the main parameter of the remaining value. In this protocol, the second parameter consisting of the node degree or proximity neighbors selection is also used. The HEED protocol selects the cluster heads according to the hybrid of node residual energy and a secondary parameter such as node proximity to its neighbors or the node degree. Moreover, HEED can asymptotically guarantee the connectivity of clustered networks [5].

The PEGASIS[1] [6] is a near-optimal chain-based protocol which is an improvement on the LEACH method. In PEGASIS [6], each node communicates only with a close neighbors and turns to the base station. Therefore, it reduces the amount of energy spent per round.

In [3], a new hybrid Genetic Algorithm (GA) and a K-means clustering namely EAR[2] has been proposed to maximize the network lifetime efficiency. This method

[1] Power-efficient gathering in sensor information.

[2] Energy Aware Routing.

uses the improved GA and the dynamic clustering network environment that is generated by k-means algorithm [3].

The new hybrid method of GA and fuzzy logic has been applied to balance the energy consumption among the CHs. In this method, the fitness function is calculated based on the difference of the current energy and the previous one. The BS selects the chromosome that has the minimum difference.

The fitness function is:

$$F = \left| E^{k}_{network} - E^{k-1}_{network} \right|$$

In the above formula, $E^{k}_{Network}$ represents the Energy in the round k (Energy flow in the network) and $Ek^{-1}_{Network}$ depicts the Energy of round k-1.

The algorithm contains a number of steps, which have been summarized here:

The first case is initialize the network (specifying the number of sensors). In the second phase, each node sends its position to its neighbors. Then, to calculate the "probability", fuzzy parameters such as energy, density, and centrality have been measured.

The nodes with a higher probability of fuzzy parameters will be selected as a candidate for cluster head. After that, GA is applied to select the cluster head. The cluster heads are presented to all nodes. Each sensor node joins to the nearest or adjacent cluster head and sends the information to the cluster head. The data aggregation is performed in each cluster head, and then the cluster head sends the received information of the package [3].

LEACH[3] [11], which is a self-organized clustering protocol, distributes the energy load to the network sensors. In this algorithm, the nodes organize themselves in the local clusters, therefore a node can act as a cluster in the cluster. High-energy nodes are randomly rotated to avoid the clogging the energy of the entire cluster network. Additionally, the data is locally aggregated to reduce the power consumption and increase the network life [11].

In this method [11], the nodes select themselves as a cluster head with a certain probability. These cluster heads inform the rest of nodes about their status. Each node chooses a cluster based on the minimum communication energy and becomes a member of selected cluster. When all nodes are organized into the clusters, each cluster head creates a scheduler for its nodes. Based on this scheduler, to saves the energy, the non-Cluster head nodes only turn on their radio when it comes to sending them, and in the rest of the time they are silent.

When the cluster node collects all members' data, it aggregates and compress the data and sends it to the base station. In this method, the nodes decide on their remaining energy. Each node decides independently of the other nodes. Therefore, additional negotiations are needed to diagnose the cluster head. LEACH is a cluster-based routing protocol in wireless sensor networks which is introduced in 2000 by Heinzelman et al. [11]. The purpose of this protocol is to reduce the energy consumption of nodes and improve the lifespan of the wireless sensor network.

[3] Low Energy Adaptive Clustering Hierarchy.

BCEE[4], which has been studied in [17], is a routing protocol that try to reduce energy consumption by balanced clustering of network nodes. In addition, more methods are designed for this purpose and are used in some cases. Some of them focus solely on head cluster selection like the evolutionary algorithms, data mining and fuzzy system.

In [7–9] the genetic algorithm and in [10] the ants colony algorithm and decision tree have been used for cluster head selection. The genetic algorithm is one of the best methods for determining the optimal points. In terms of input parameters and application of a set of functions and operators, one can propose a variety of methods based on the genetic algorithm for a single problem. Therefore, different researchers have presented various methods in this regard. Also, the genetic algorithm is one of the most famous and widely used evolutionary algorithms. It begins its work algorithm with a population of candidate answers (called chromosomes). During the implementation of this algorithm, the generation of chromosomes will gradually be improved and the subsequent generations will be generated in order to eventually satisfy the termination condition of the algorithm. In [12] the author has suggested a combined routing algorithm to develop the lifetime of network (Table 1).

Table 1. A review of different clustering algorithms in wireless sensor networks

Algorithm	Ref	Distribution	Selection Method	Stability	Energy efficiency
LEACH	[11]	Non-uniform	Random	Low	Low
TEEN	[16]	Non-uniform	Probable	Mid	Mid
Bayes	[14]	Non-uniform	Probable	High	High
PSO	[13]	Non-uniform	Probable	High	High
HSA	[13]	Non-uniform	Random	High	Mid
BCEE	[17]	Uniform	Random	Mid	Low
HSA-PSO	[13]	Non-uniform	Probable	High	High
Non-linear programming	This paper	Non-uniform	Probable	High	Very high

3 Proposed Algorithm

The method used in this paper is optimization with the non-linear modeling so as to choose an appropriate cluster head. The algorithm methodology has been depicted in Fig. 1.

Optimization in its own concept can be used to solve every engineering problem. The mathematical designing of a module is the main part in the mathematical optimization process. To obtain good relation results in achieving a proper optimized answer, a decision-making factor in a module should be introduced as a math function. The foregoing factor is called “target function”. There are various factors that affect the

[4] Balanced-clustering Energy Efficient.

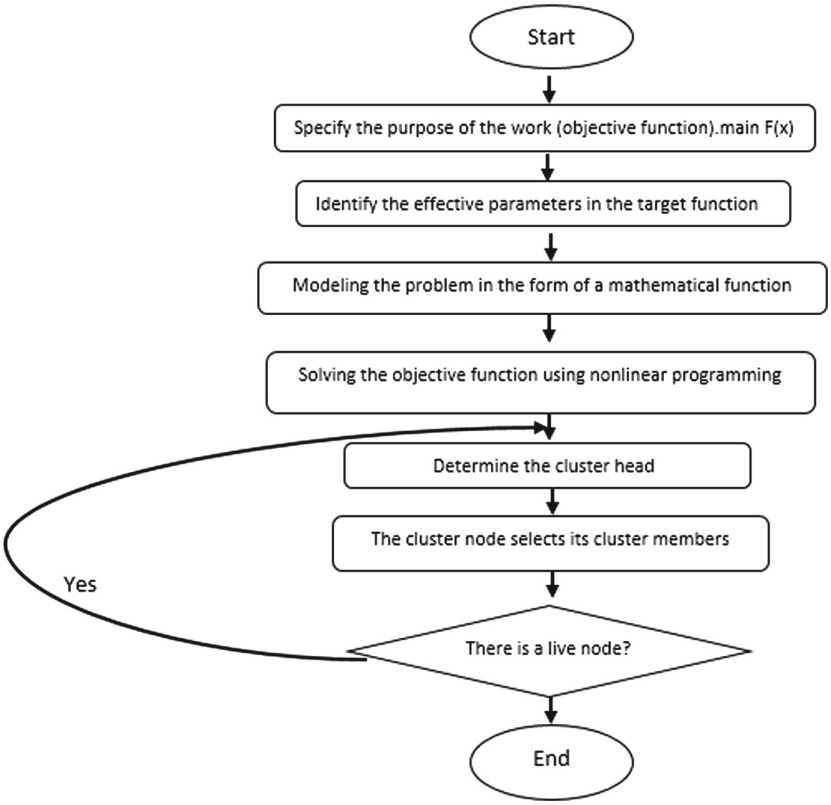


Fig. 1. Flowchart of proposed method

target function of a module and change its amount. These factors are introduced as parameters in the math pattern and are called "design parameters". In fact, the target function is written based on the foregoing parameters. The design parameters and target function are the two non-removable elements of each optimization problem. In the math designing of an optimization problem, the limitations are written as equality or inequality relations in accordance with the design parameters. It is noteworthy that some of the optimization problems have no limitations. In optimization problems, and among all of the accepted modules, the module that minimizes or maximizes the target function is called "optimized module" (according to the point that the problem will be minimized or maximized).

After recognizing all of the properties and parameters of a problem, we will write an appropriate math relation for optimization. In this mathematical pattern, the target function is a criterion for making decisions. The decision-making criterion with a combination of existing limitations will create a module. Writing the mathematical pattern of a problem is the most important part of optimization. The mathematical pattern can be written identically in all of the science and fields. This general module that consists of the target function, equality limitations and inequality limitations is as follows:

$$\begin{cases} Min\, F(x) & \{x\} \epsilon R^n \\ G_i(x) \leq 0 & i = 1 \ldots m \\ H_i(x) = 0 & j = 1 \ldots p \end{cases} \tag{1}$$

The function F(x) shows the target function of the problem that has to be minimized. Numbers n, m and p are the number of design parameters, inequality limitations and equality limitations, respectively [13].

After recognizing the necessities, limits and required criteria, a suitable math pattern is suggested and then solved.

Our point is to increase the lifetime of wireless sensor networks by selecting the proper head cluster node. The following factors are those which affect our point which increases the network lifetime; in the meantime, they are the parameters of the problem:

- Sum of distance
- Residual energy
- Density of nodes
- Weight of nodes
- Amount of initial energy of nodes

The target function of this module is considered as follows:

$$\text{Max(ED)} = (\text{Weight} * \text{density} * \text{Energy})/(\text{meandistance}) \tag{2}$$

And the limitation or the problem condition is:

$$\rho(i) = \sum_{i=0}^{n} \left(sumd(i) < \frac{d_0}{8} \right) \tag{3}$$

The method used in this paper utilizes a mathematic method based on non-linear modeling for the purpose of selecting the right head cluster node. In this method, at first we calculate the target function for each node and the head cluster node is the one with the maximum value of the target function. Then head cluster nodes begin to select their own members of the cluster (according to the density parameter or compression around the node) and in fact, they choose their own domain.

Distance between the nodes is calculated by the Euclidian relation (Eq. 4):

$$d = \sqrt{(x1 - x2)^2 + (y2 - y1)^2} \tag{4}$$

And in each round, the weight is changed based on a criterion or a condition. The foregoing parameter is renewed in each round of amounting according to the following relation that has two cases:

1. If the number of nodes that are selected as the cluster head node is less than 3, this parameter will be valued by the following relation:

$$Weight = weight - 0.5 \tag{5}$$

2. If the number of nodes that are selected as the cluster head node is more than 3, this parameter will be valued by the following relation:

$$Weight = weight - 0.25 \tag{6}$$

As it was said before, a module generally has several practical answers. Among all of the available answers, the best one is chosen, which is called "optimized answer". It is noteworthy that "optimization" is, in fact, a way to obtain the best answer.

In optimization problems, all of the relations, including the target function and limitations are introduced as the first power of the parameters. This is called "linear programming". Should at least one of the parameters enter the problem limitations or the target function with power more than one or in a form of non-linear functions like trigonometric functions or exponential functions, a non-linear optimization problem will emerge.

After making the relations of optimization problems, we have to solve them. There is not a unique method for the efficient solution to optimization problems. Because of this, various methods of optimization have been developed to solve different types of optimization problems. According to [13], the methods of non-linear optimization problem-solving are various, depending on the problem as to whether it is bound or unbound and whether it is linear or non-linear. The search method, repetitive method, binary gradient method, Newton method, modified Newton method, semi-Newton method, gradient picture method and multi-target optimization are some of the solving methods.

This paper has used a combined solving method, which is a combination of the repetitive method and convertor method (a change to the parameter). There is a loop in the repetitive part in each round, which calculates the target function for each node, and in the parameter changing part, we have used a weight parameter changing and renewing the parameter.

4 Simulation

We used Matlab software for simulation. We compared the input of our suggested method with the output of two other papers, and the result of this comparison is as follows:

In the beginning of the network, the nodes are randomly scattered in the environment and the environment dimensions have been shown in Fig. 2 (200 * 200). Also, the initial energy of the nodes at the beginning is considered as 2 J. Nodes with the shortest distance and the highest energy and density are chosen as a head cluster.

The consumed energy for the data transmission of the head cluster node must be calculated in each round, and based on that, the level of residual energy needs to be updated. According to (7), the amount of consumed energy for each member is calculated by the following relation:

The amount of lost or consumed energy for the head cluster node is calculated as follows:

$$E_{dis} = E_{Rx(l)}^{elec} + E_{Tx(l)}^{elec} + E_{DA} \tag{7}$$

The consumed energy to receive the cluster head node is calculated by the following relation:

$$E_{RX(l)} = l * E_{Rx}^{elec} \tag{8}$$

The amount of residual energy for each cluster head node at the end of each round:

$$E_{res} = E - E_{dis} \tag{9}$$

And for member nodes:

$$E_{res} = E - E_r \tag{10}$$

This procedure will be continued until the end of the network lifetime (until the death of the last node).

4.1 The Default Values and Assumptions

As it was said in the previous part, we have compared our input with the [14] article. In this comparison, we have used data from this paper. The constant data (the simulation parameters) of the [14] is as follows (Table 2):

Table 2. Default values

Parameter	Value
Number of wireless nodes	100
Operation environment	(200 * 200)
Location of the sync node	(100, 100)
Number of rounds	8000
Cluster head	30%, 5
E_{Tx}^{elec}	50 nj
E_{Rx}^{elec}	50 nj
ε_{fs}	10 nj
ε_{mp}	0.0013 pj
L	4000 bits
E_{DA}	5 nj

The amount of d_0 is calculated according to the following expression:

$$d_0 = \sqrt{(\varepsilon_{fs}/\varepsilon_{mp})} \tag{11}$$

and the initial energy of the nodes is joules.

5 Evaluation and Data Analysis

After running the proposed algorithm in Matlab software, and based on the default values from the previous part, the results of outputs have been shown in Figs. 2, 3, 4 and 5. Figures exhibit the number of existing living nodes in each round and the amount of the residual energy. It is completely evident that the suggested method has better results than the compared algorithms.

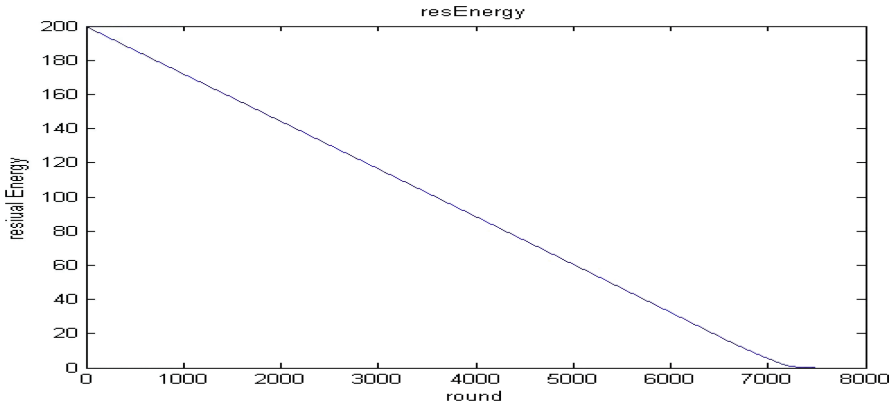


Fig. 2. The residual energy in each round

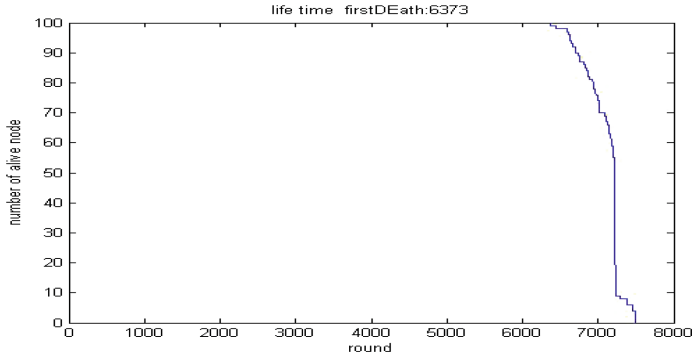


Fig. 3. The number of live nodes in each round

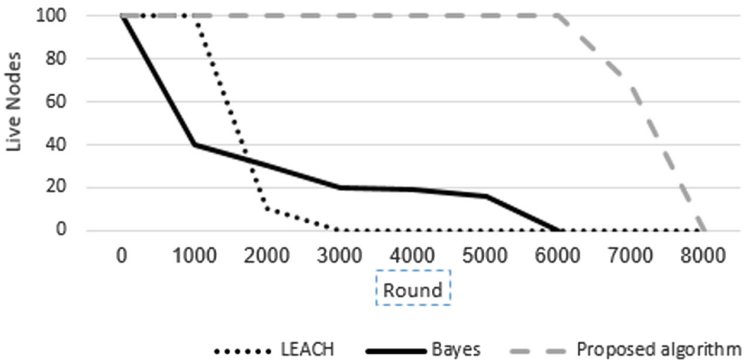


Fig. 4. The number of live nodes in each round, as compared with the other methods

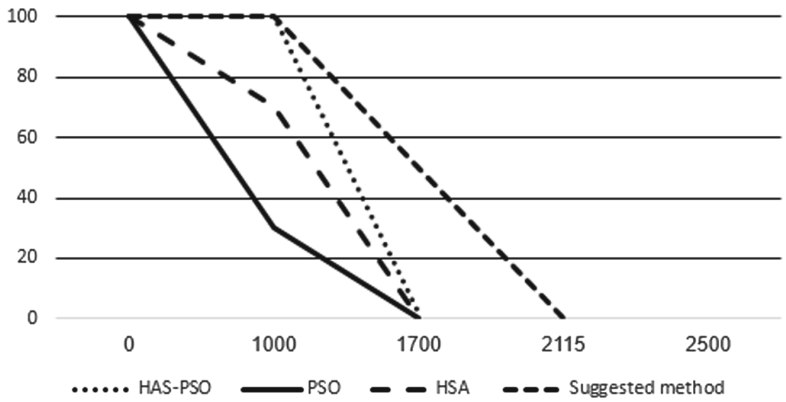


Fig. 5. The residual energy in each round compared with the HSA, PSO method and combined HSA-PSO method

Figures 4 and 5 are the results of the comparison of the proposed algorithm with the LEACH [11], Bayes algorithms [14], each of the HSA and PSO methods and their combination [13]. The comparison of the proposed algorithm with the other studied algorithms in this paper has been shown in Table 3. It is obvious that our proposed algorithm has the best performance in the optimization of the cluster head selection in wireless sensor networks.

Table 3. The results of the comparison of the proposed method with the related works

Algorithm	First dead node	Last dead node
Loss function according to Bayes [14]	95	6800
Leach algorithm [11]	1400	3500
PSO [13]	11	1600
HAS [13]	8	1680
Hybrid algorithm of HSA-PSO [13]	1304	1744
Suggested method	1181	2115

6 Conclusion and Future Works

As it was said before, mathematical optimization methods in choosing the head cluster in wireless sensor networks have rarely been used. Whereas using methods with a math basis have numerous advantages, including the algorithm flexibility. According to the results of the comparison of our proposed methods with Leach algorithm and loss function based on Bayes, we can realize that in our proposed method, further nodes could survive longer and the network lifetime is longer. This method will distribute the energy in the network in a completely equal and balanced manner and all nodes will survive until the last round, and in the final rounds all nodes start to lose their energy simultaneously, which is the best advantage of this method. It is evident that the proposed method has the best performance in the optimization of wireless sensor networks.

As it was said in the previous part, the suggested algorithm has high flexibility. Should someone be interested in working in this field, he can easily work and carry out research in this major by adding or removing parameters or by changing the available parameters. In addition, there are still many other methods to solve this problem. Interested researchers can use the following methods to select a proper head cluster like the honeybee method, intercross method and firefly (glow worm) method. Besides, they can utilize the linear or non-linear methods or a combination of these methods to reach better results.

References

1. Deosarkar, B.P., Yadav, N.S., Yadav, R.P.: Clusterhead selection in clustering algorithms for wireless sensor networks: a survey. In: 2008 International Conference on Computing, Communication and Networking, pp. 1–8 (2008)
2. Blum, C., Roli, A.: Metaheuristics in combinatorial optimization: overview and conceptual comparison. ACM Comput. Surv. CSUR **35**(3), 268–308 (2003)
3. Amgoth, T., Jana, P.K.: Energy-aware routing algorithm for wireless sensor networks. Comput. Electr. Eng. **41**, 357–367 (2015)
4. Li, M., Yang, B.: A survey on topology issues in wireless sensor network. In: ICWN, p. 503 (2006)
5. Younis, O., Fahmy, S.: HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. IEEE Trans. Mob. Comput. **4**, 366–379 (2004)
6. Lindesy, S., Raghavendra, C.: PEGASIS: power-efficient gathering in sensor information system. In: Proceedings of 2002 IEEE Aerospace Conference, pp. 1–6 (2002)
7. Barekatain, B., Dehghani, S., Pourzaferani, M.: An energy-aware routing protocol for wireless sensor networks based on new combination of genetic algorithm & k-means. Proc. Comput. Sci. **72**, 552–560 (2015)
8. Pal, V., Singh, G., Yadav, R.P.: Cluster head selection optimization based on genetic algorithm to prolong lifetime of wireless sensor networks. Proc. Comput. Sci. **57**, 1417–1423 (2015)
9. Hamidouche, R., Aliouat, Z., Gueroui, A.: Low energy-efficient clustering and routing based on genetic algorithm in WSNs. In: International Conference on Mobile, Secure, and Programmable Networking, pp. 143–156 (2018)

10. Kaur, S., Mahajan, R.: ACCGP: enhanced ant colony optimization, clustering and compressive sensing based energy efficient protocol (2017)
11. Cui, X.: Research and improvement of LEACH protocol in wireless sensor networks. In: 2007 International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications, pp. 251–254 (2007)
12. Rao, S.S.: Engineering Optimization: Theory and Practice. Wiley (2009)
13. Shankar, T., Shanmugavel, S., Rajesh, A.: Hybrid HSA and PSO algorithm for energy efficient cluster head selection in wireless sensor networks. Swarm Evol. Comput. **30**, 1–10 (2016)
14. Jafarizadeh, V., Keshavarzi, A., Derikvand, T.: Efficient cluster head selection using Naïve Bayes classifier for wireless sensor networks. Wirel. Netw. **23**(3), 779–785 (2017)
15. Lloret, J., Shu, L., Gilaberte, R.L., Chen, M.: User-oriented and service-oriented spontaneous ad hoc and sensor wireless networks. Ad Hoc Sens. Wirel. Netw. **14**(1–2), 1–8 (2012)
16. Manjeshwar, A., Agrawal, D.P.: TEEN: a routing protocol for enhanced efficiency in wireless sensor networks. In: Null, p. 30189a (2001)
17. Cui, X., Liu, Z.: BCEE: a balanced-clustering, energy-efficient hierarchical routing protocol in wireless sensor networks. In: 2009 IEEE International Conference on Network Infrastructure and Digital Content, pp. 26–30 (2009)

A Survey on Measurement Metrics for Shape Matching Based on Similarity, Scaling and Spatial Distance

Bahram Sadeghi Bigham[1(✉)] and Samaneh Mazaheri[2]

[1] Department of Computer Science and Information Technology, Institute for Advanced Studies in Basic Sciences, Zanjan, Iran
sadeghi@iasbs.ac.ir

[2] Faculty of Business and Information Technology, Ontario Tech University, Ontario, Canada
Samaneh.Mazaheri@uoit.ca

Abstract. Measuring difference or similarity between data is one of the most fundamental steps in data science. This topic is of the utmost importance in many artificial intelligent systems, machine learning and any data mining and knowledge extraction. There are several applications in image processing, map analysis, self-driving cars, GIS, etc., when data are in the shape of polygons or chains. In the present study, principal metrics for comparing geometric data are studied. In each metric, one, two or three features out of similarity, scaling, and spatial distance is considered. Evaluation for metrics based on three perspectives is discussed and results are provided in a detailed table. Additionally, for each case, one practical application is presented.

Keywords: Shape matching · Similarity · Measurement metric · Clustering · Data science

1 Introduction

Data science is one of the hot topics nowadays which is the knowledge of managing the existing data and extracting useful information to utilize in different situations. Some other topics such as data mining, big data, and data extraction also have the same objective. Comparison between data is a significant part of these topics. For instance, clustering and classification are not feasible without comparing data and computing the respected difference. In addition, there is a need for comparing data in all database queries.

To compare each type of data, there are special metrics. In this study, geometric data are on focused. Data are in the shape of polygons, path, tree, parts of a map, or simple shapes. There are three parameters in consideration, when comparing shapes; similarity (called first feature in this paper), scaling (called second feature), and spatial distance (third feature). At the time of evaluating similarity, scale of two shapes is not important; so the scaling changes in a way that two shapes have the most similarity.

M. Bohlouli et al. (Eds.): CiDaS 2019, LNDECT 45, pp. 13–23, 2020.
https://doi.org/10.1007/978-3-030-37309-2_2

Additionally, it is assumed that two shapes are overlapped and spatial distance will not make them different.

In fact, scaling means magnitude or measure of two shapes, which can be expressed in different ways, such as perimeter or area or combination of those. Third feature, i.e. spatial distance indicates the distance between two shapes. For example, between introduced metrics in this study, turning function is examined first features, i.e. similarity; while Fréchet distance assessing all three features at the same time.

When two shapes are compared, based on the application, one, two or all three features can be considered.

In next section, important metrics for measuring similarity between geometric shapes will be introduced, and in Sect. 3, a few applications of geometric data comparison that require different features will be discussed. In Sect. 4, a table including a comparison for different metrics has been presented in terms of considering each feature, which will help researchers to find out the most suitable metric based on their applications, and objectives, by considering the metric's capabilities. Section 5 will conclude the paper, and suggest future works.

2 Some Common Metrics

For comparing any two geometric objects, an appropriate metric is required. Several metrics have been proposed for this specific problem. However, in this study, only those methods which ignores definition and color are considered. Furthermore, learning techniques, and utilizing neural networks also excluded in this paper.

For simplicity, assume two polygons, two chains, or two cuts from a map are being compered together. In some applications, all of these three cases can occur. For instance, trajectories are the most common objects that mentioned metrics are applied on. Trajectories can be two simple chains, two simple polygons, or a piece of urban map. In trajectory topic, time is another dimension of the data, which is ignored in this study. In the following, some of the recognized metrics which have been used for this problem, will be discussed.

Essentially, trajectory is allocated into cohesive groups according to their mutual similarities. An appropriate metric is necessary [1–3].

Euclidean Distance [4]: Euclidean distance requires that lengths of trajectories should be unified and the distances between the corresponding trajectories points should be summed up,

$$D(X.Y) = \frac{1}{N}\sum_{i=1}^{N}\left((x_{1i} - y_{1i})^2 + (x_{2i} - y_{2i})^2\right)^{1/2}$$

where x_{ji} and y_{ji} indicate the i^{th} point of trajectories X and Y in Cartesian coordinate. N is the total number of points. In [4], Euclidean distance is used to measure the contemporary instantiations of trajectories.

Hausdorff Distance [5]: Hausdorff distance measures the similarities by considering how close every point of one trajectory to some points of the other one, and it measures trajectories X and Y without unifying the lengths in [6, 7],

$$D(X.Y) = \max\{d(X.Y).d(Y.X)\}, \text{in which}$$
$$\begin{cases} d(X.Y) = max_{x\in X} min_{y\in Y}||x - y|| \\ d(Y.X) = max_{y\in Y} min_{x\in X}||y - x|| \end{cases}$$

Bhattacharyya Distance [8]: Consider two data sets where both are divided into N sets. According to the distributions, each set would have its own frequency that the probability of occurrence of all data in each set, sums up to 1.

Bhattacharyya coefficient formula (e.g. ρ) is:

$$\rho(P.P\prime) = \sum_{i=1}^{N} \sqrt{P(i)P\prime(i)}$$

Maximum value for Bhattacharyya coefficient happens when each and every rectangle (probably the sets) are same with value equal to 1 and at most difference, this value is 0 or converges to 0.

Now Bhattacharyya metric is defined based on Bhattacharyya coefficient as follows:

$$d(p.p\prime) = \sqrt{1 - \rho(p.p\prime)}$$

This formula which treats in contrast of Bhattacharyya coefficient; whenever there is more similarity, Bhattacharyya coefficient yields to 1 and Bhattacharyya distance yields to 0.

This formula is defined for Bhattacharyya metric in range of [0, 1], to overcome this issue, this metric can also be defined as $d = -ln(\rho)$. In this case, with more similarity, the Bhattacharyya coefficient equals to 1 and $ln(1) = 0$, so Bhattacharyya distance becomes 0; and if there is less similarity, Bhattacharyya coefficient equals to 0 and $ln(0)$ is equal to infinite negative, thus the Bhattacharyya distance is reported as positive. To concise, $0 \leq d < \infty$ is the Bhattacharyya distance between two datasets where they are distributed in unity ranges.

Fréchet Distance [9]: Fréchet distance measures similarity between two curves by taking into account location and time ordering. After obtaining the curve approximations of trajectories X and Y, their curves map unit interval into metric space S, and a re-parameterization is added to make sure t cannot be backtracked. Fréchet distance is defined as

$$D(X.Y) = inf_{\alpha.\beta}\, max_{t\in[0.1]}\{d(X(\alpha(t)).Y(\beta(t)))\},$$

Where d is distance function of S, α, β are continuous and non-decreasing re-parameterization.

Dynamic Time Warping (DTW) Distance [10, 11]: DTW is a sequence alignment method to find an optimal matching between two trajectories and measure the similarity without considering lengths and time ordering [10, 11].

$$W(X.Y) = min_f \frac{1}{n} \sum_{i=1}^{n} \left| x_i - y_{f(i)} \right|$$

where X has n points and Y has m points, all mappings $f : [1.n] \rightarrow [1.m]$ should satisfy the requirements that $f(1) = 1, f(n) = m$ and $f(i) \leq f(j)$, for all $1 \leq i \leq j \leq n$.

Turning Function [12]: Another common method to compare, is to use Turning Function. In this method, shape analogy is done in the same scale, thus the difference in sizes would not matter and this metric merely discusses the similarity in shapes' structures.

To this point, first a plot with x axis representing the length and y axis representing angle in radian is considered. Then starting from one side of the shape, its angle to horizon is measured and inserted on the plot. In the next step, the next side and its angle is inserted on the plot. This process continues until it reaches the starting point. Similar process is applied for the other shape.

To compute the differences of two shapes using the Turning Function, it is enough to find the area between these two plots. But it is important to consider that by changing the starting point in shapes, various results are obtained. To overcome this issue, separate plots should be illustrated for each starting point, the acceptable answer is the one showing the least difference.

Longest Common Subsequence (LCSS) Distance [1]: $LCSS_{\varepsilon.\delta}(X.Y)$ aims at finding the longest common subsequence in all sequences, and the length of the longest subsequence could be the similarity between two arbitrary chains with different lengths.

Some more other distance types are proposed to consider more properties such as angle distance [13], center distance and parallel distance, which are defined as:

$$d_{angle}(L_i.L_j) = \begin{cases} |L_j| \times \sin(\theta); & 0 \leq \theta \leq \frac{\pi}{2} \\ |L_j|; & \frac{\pi}{2} < \theta \leq \pi \end{cases}$$

where θ is the smaller intersecting angle between L_i and L_j.

$$d_{centre}(L_i.L_j) = ||centre(L_i) - centre(L_j)||,$$

where $centre(L_i)$ and $centre(L_j)$ are the centre points of lines L_i and L_j [14].

$$d_{parallel}(L_i.L_j) = \min\{l_1 - l_2\},$$

where l_1 is the Euclidean distances of p_s to s_i and l_2 is that of p_e to e_i. p_s and p_e are the projection points of s_j and e_j onto L_i respectively [15].

3 Some Applications

As discussed earlier, in any application, some of the features, similarity, scaling and spatial distance are considered. The main objective of this study is to categorize these features, and introducing appropriate metrics for each specific application. Three applications of shape comparison which have different requirements for the mentioned features are discussed in the following.

3.1 Fingerprint Matching [16]

As it can be seen in Fig. 1, to compare two fingerprints, computing similarity is essential. Also, it is required to consider the scale for these two fingerprints. However, it is not necessary to consider the spatial distance.

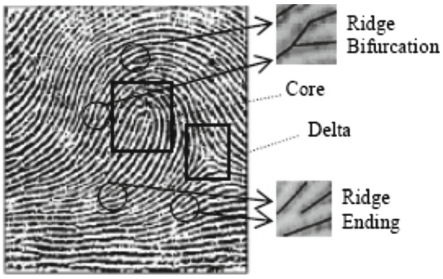


Fig. 1. Singular Points (Core & Delta) and Minutiae (ridge ending & ridge bifurcation) [16]

There are three different level of information contained in a fingerprint, namely, pattern (Level 1), minutiae points (Level 2), and pores and ridge contours (Level 3) [16].

There are a lot of contemporary fingerprint authentication systems which are automated and (Level 2 features) minutiae based. Minutiae-based systems rely on finding correspondences between the minutiae points present in "query" and "reference" fingerprint images generally. These types of systems normally perform well with high-quality images of fingerprints and enough surface area of fingerprint. However, these conditions are not always attainable [16].

In many situations, only a small portion of the "query" fingerprint can be compared with the "reference" fingerprint. This will lead to reduction of number of minutiae correspondences. In these cases, the matching algorithm would not be able to make a decision with high accuracy [16].

When the system is dealing with intrinsically poor quality fingerprints, only a subset of the minutiae can be extracted and used with enough reliability. Minutiae do not always constitute the best trade-off between accuracy and robustness, they may carry most of the fingerprint's discriminatory information. This fact has led the designers of fingerprint recognition techniques to look for other distinctive fingerprint features, other than minutiae which may be utilized in conjunction with minutiae

(consider that it is not as an alternative) to increase the system robustness and accuracy. It is worth mentioning that the presence of level three features provides detail for matching as well as potential for increased accuracy [16].

In this case, computing similarity and scale play an important role in fingerprint authentication systems, so the matching algorithm would be able to make a decision with high certainty [16].

3.2 Robot Pose Estimation

Autonomous navigation of a mobile robot is generally defined as control of robot motion to arrive at a given position in its environment without human intervention. This navigation task can be decomposed into the various aspects such as robot pose estimation (localization) and path planning.

To generate a path from an initial position to a target position, the localization is vital which frequently provides and updates a reliable estimate of the position and orientation of the robot given a global coordinate system in a structured or semi structured environment. Maintaining an estimate of the subsequent location of a robot using odometer, while the position and orientation of the robot is known at any time, is a common solution. But this method is sensitive to errors and the outcome is not reliable because of the integration of errors over time. Thus, the need arises for developing an algorithm to overcome the challenge.

One possible strategy to tackle the problem of pose estimation is to use scan matching, in which the geometric representation of the current scan is frequently compared to a reference scan until an optimal geometric overlap with the reference scan is obtained.

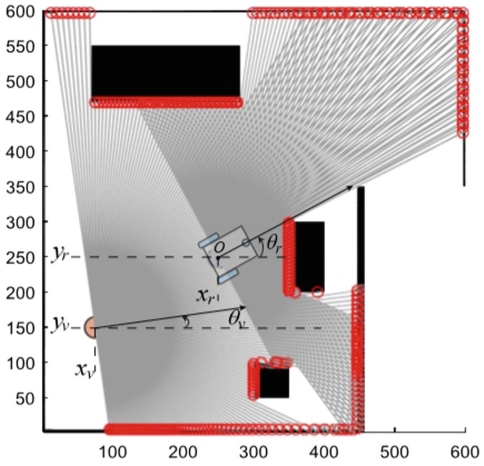


Fig. 2. RSD (real spatial description), and VSD (virtual spatial description) of a robot [18]