A SURVEY ON CONSENSUS PROTOCOLS IN MULTI-AGENT SYSTEMS

Tran Thi Minh Dung

The University of Danang, University of Science and Technology; greenfield01@gmail.com

Abstract - In the past few year, the research community has paid much attention to consensus problems in multi-agent systems, especially, for wireless sensor networks, i.e. control systems that are physically distributed and cooperate by exchanging information through a communication network. This paper gives a brief survey on consensus problems for multi-agent systems based on the current literature. In particular, the general view of the consensus protocols as well as its applications in various fields are presented. Furthermore, we also summarize the studies on designing the consensus matrix according to its convergence analysis. Finally, we give some open problems that can be investigated in the future.

Key words - Consensus Protocols; Multi-agent systems; Graph Theory; Formation control; Coordination control.

1. Introduction

Multi-agent systems (MASs) have received a growing interest in the last decades. They are developed for the demand of flexibility, robustness, and re-configuration features that appear in various application domains including manufacturing, logistics, smart power grid, building automation, disaster relief operation, intelligent transportation systems, surveillance, environmental monitoring and exploration, infrastructure security and protection, etc. A MAS is a system composed of multiple interacting intelligent agents (sensors, plants, vehicles, robots, etc.) and their environment in Figure 1.

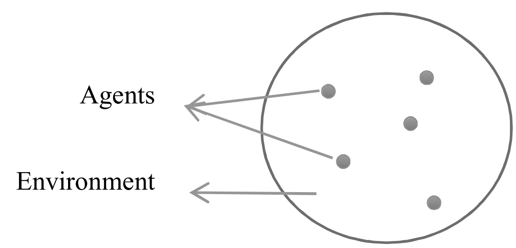
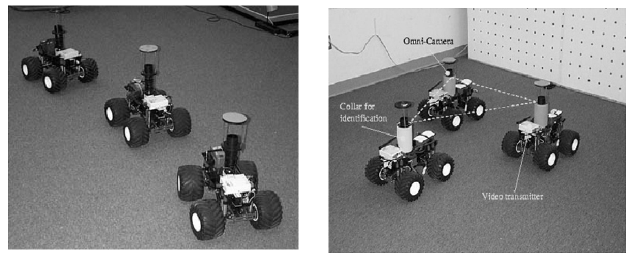


Figure 1: A Multi-agent system with their agents and the environment.

To summarize, a MAS is a group of nodes (agents) representing vehicles, sensors, plants, etc., which are able to exchange information in order to reach a common goal. Schematically, MAS can be represented by a network of nodes interconnected via a communication topology. Interconnections between agents in a MAS are usually modeled by directed or undirected graphs.

One thing to note here is that a MAS can deal with tasks that are difficult or even impossible to be accomplished by an individual agent. During recent decades, MASs gain a widespread interest in many disciplines such as mathematics, physics, biology, computer science and social science. An increasing range of research topics in MASs includes cooperation and coordination, distributed computation, automatic control, wireless communication networks, etc.

In automatic control, the interests of MASs is particularly relevant when one has to face with systems consisting of multiple vehicles (which are considered to be the agents) with several sensors and actuators that are intended to perform a coordinated task. In recent years, these cooperative control capabilities including formation control, rendez-vous, attitude alignment, flocking, congestion control in communication networks, task and role assignment, air traffic control have been analysed.



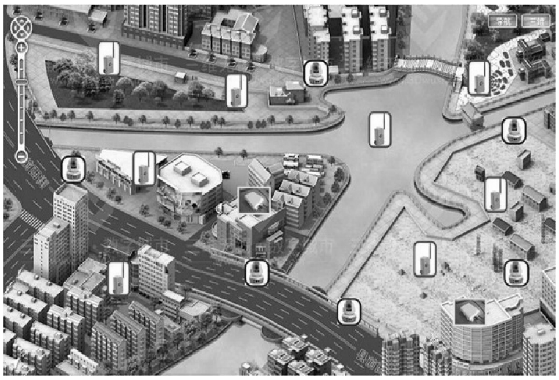
(a) (b)

Figure 2: Examples with cooperative control and formation control: (a) cooperative localization of robots, (b) a simple formation control. Pictures from the website: (http://www.cis.upenn.edu/~cjtaylor/RESEARCH/projects/Multibots/Multibots.html).

In cooperative control strategies to be successful, numerous issues must be addressed, including the definition and management of shared information among a group of agents to facilitate the coordination of these agents. Furthermore, the shared information may take the form of common objectives, common control algorithms, relative position information, or a world map. Information necessary for cooperation may be shared in a variety of ways. For instance, relative position sensors may enable vehicles to construct state information for other vehicles, knowledge may be communicated between vehicles using a wireless network, or joint knowledge might be preprogrammed into the vehicles before a mission begins. Therefore, cooperation requires that the group of agents reach consensus on the coordination data.

In a typical centralized structure, a fusion center (FC) collects all measurements from the agents and then makes the final computations. However, due to the high information flow to FC, congestion can occur. Such a structure is vulnerable to FC failure. Also, the hardware requirements to build wireless communications can be one of reasons for an increase in the cost of the devices and thus, a higher overall cost of the network. For these reasons, a centralized structure can be inefficient. Hence, the research trend of MASs has shifted to decentralize MASs where the interaction between agents is implemented locally without global knowledge. A good example is wireless sensor networks (WSNs), which find broad application domains such as military applications (battlefield surveillance, monitoring friendly forces, equipment and ammunition, etc.), environment applications (forest fire detection, food detection, etc.), health applications (tele-monitoring of human physiological data, etc.), home automation, formation control, etc.. Figure 3 depicts a WSN that collects data for the air quality, light intensity, sound volume, heat, precipitation and wind.

Figure 3: A network of wireless sensors on the light poles all over the city



Therefore, based on local information and interactions between agents, how can all agents reach an agreement (consensus)? This problem is called consensus problem, which is to design a network protocol based on the local information obtained by each agent so that all agents finally will reach an agreement on certain quantities of interest.

Consensus problems of MASs have received tremendous attention from various research communities due to its broad applications in many areas including multi-sensors data fusion [1], flocking behavior of swarms [2], [3], multi-vehicle formation control [4], distributed computation [5], rendez-vous problem [6] and so on. More specifically, average consensus protocols (i.e. the agreement corresponds to the average of the initial values) are commonly used as building block for several distributed control, estimation or inference algorithms.

In the recent literature, one can find average consensus protocols embedded in the distributed Kalman filter [7], distributed Least Squares algorithm [8], distributed Alternating Least Square for tensors factorization [9], distributed Principal Component Analysis [10], or distributed joint input and state estimation [11] to cite few. However, the asymptotic convergence of the consensus protocols is not suitable for these kinds of sophisticated distributed algorithms. A low asymptotic convergence cannot ensure the efficiency and the accuracy of the algorithms, which can lead to other unexpected effects. For example, regarding to the WSNs, a reduction in the total number of iterations until convergence can lead to a reduction in the total amount of energy consumption of the network, which is essential to guarantee a longer lifetime for the entire network. On the other hand, the protocols that guarantee a minimal execution time are much more appealing than those ensuring asymptotic convergence. For this purpose, several contributions dedicated to finite-time consensus have been recently published in the literature [12]. In other words, the consensus is obtained in a finite number of steps.

There are several approaches used by a number of researchers to reach the finite-time consensus in recent years: linear iteration, leader-follower type architecture, and so on. In literature, most authors use a linear iteration, where each node repeatedly updates its value as a weighted linear consensus scheme so that, at each time-step, each node will have only to transmit a single value of its neighbors.

In this paper, we make a brief survey on consensus problem in multi-agent systems in Section 1 and cite a few of its applications in this Section. After introducing the mathematic background on graph theory in Section 2, we present the consensus problem and the consensus matrix design in Section 3. Finally, we conclude this paper and point out some open problems for future research in Section 4.

2. Graph Theory

As we state in the Section I, the interconnection between agents in a network can be modeled by a graph as shown in Figure 4 with graph$G\ (V, E)$.

Where $V = \{1, 2, \ldots, N\}$ is the set of vertices (nodes or agents), and $E \subset V \times V$ is the set of edges (links between agents).

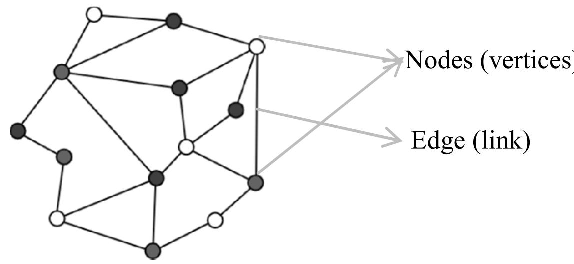


Figure 4: Graph $G\ (V, E)$

According to the communication policy, the graph $G\ (V, E)$ can be distinguished as undirected graph and directed graph. If there is no direction assigned to the edges, then both edges (i, j) and (j, i) are included in the set of the edges E. The graph is called undirected graph. It is said that if agent i and j are connected, then link between i and j is included in E, $(i, j) \in E$. And then, i and j are both called neighbors. The set of neighbors of agent i is denoted by N_i and its degree is presented by $d_i = |N|$, where |.| stands for the cardinality. Conversely, if a direction is assigned to the edges, the relations are asymmetric and the graph is called a directed graph. For a directed edge(i, j), i is called the head and j is called the tail. A vertex i is connected to j by a directed edge, or that j is a neighbor of i if $(i, j) \in E$.

In an undirected graph G, two vertices i and j are connected if there is a path from i to j. And an undirected graph G is connected if for any two vertices in G there is a path between them. Conversely, two vertices i and j in G are disconnected if there is no path from i to j. A directed graph is strongly connected if between every pair of distinct vertices (i, j) in G, there is a directed path that begins at i and ends at j. It is called weakly connected if replacing all of its directed edges with undirected edges produces a connected undirected graph. A graph is said to be complete if every pair of vertices has an edge connecting them, which means that the number of neighbors for each vertex is equal to *N-1*.

We denote by **A** the adjacency matrix of the graph with the entries$a_{i,j}$ given by $a_{i,j} = \begin{cases} 1, & if\ (i,j) \in E \\ 0, & otherwise \end{cases}$.The graph Laplacian matrix **L** is defined as the matrix with entries$l_{i,j}$

given by $l_{i,j} = \begin{cases} \sum_{k=1,k\neq i}^{N} a_{i,k} & if\ j = i \\ -a_{i,j} & if\ j \neq i \end{cases}$. Degree matrix **D** of the graph has vertex degree d_i, $i \in V$ on its diagonal and zeros elsewhere.

3. Consensus problem

Consensus issue in networks of autonomous agents has been widely investigated in various fields, including computer science and engineering. In such networks, according to an a-priori specified rule, also called protocol, each agent updates its rate based on the information received from its neighbors with the aim of reaching an agreement to a common value. When the common value corresponds to the average of the initial states average consensus is to be achieved.

Example 1: Consider an arbitrary network of 5 agents communicating with each other as described in Figure 5. Each agent has an initial value. A consensus protocol is an interaction rule that specifies the information exchange between an agent and all of its neighbors on the network to reach an agreement regarding a certain quantity of interest that depends on the state of all agents. Informally, despite the initial values of all agents, the output of the given network is converged to the common value (in the case, the average of initial values).

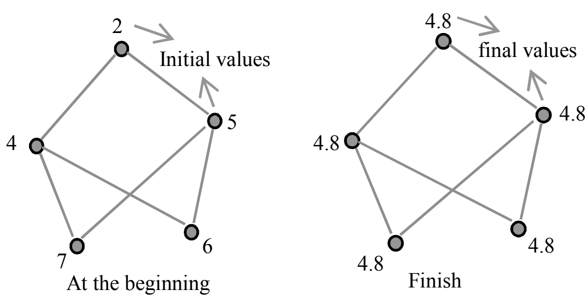


Figure 5: Average Consensus in a network: initial condition (left) and steady state (right)

In the literature, consensus protocols can be classified as follows [13]:

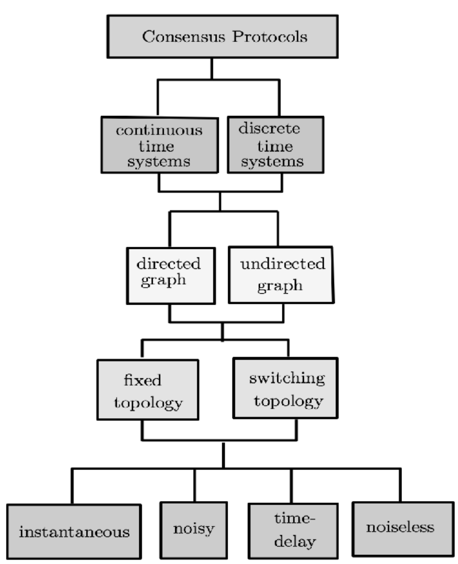


Figure 6: Classification of Consensus protocols

3.1. Definition

In this part, the definition of the consensus problem is given according to discrete-time systems and continuous-time systems.

Given a graph G (V, E), each node has an associated value x_i defined as the state of node *i*. Let $\boldsymbol{x}(0) = [x_1(0) \quad x_2(0) \quad \dots \quad x_N(0)]^T$ be the vector of initial states of the given network. In general, given the initial states at each node $x_i(0), i \in V$, the main task is to compute the final consensus value using *distributed linear iterations.*Each iteration involves local communication between nodes. In particular, each node repeatedly updates its value as a linear combination of its own value and those of its neighbors. The main benefit of using a linear iterations scheme is that, at each time-step, each node only has to transmit a single value to each of its neighbors.

3.1.1. Discrete-time systems

The linear iteration-based consensus update equation is:

$$x_i(k+1) = w_{ii}(k)x_i(k) + \sum_{j\in N_i} w_{ij}(k)x_j(k), \quad (1)$$
$$i = 1,2,\dots,N$$

Or equivalently in matrix form:

$$\boldsymbol{x}(k+1) = \boldsymbol{W}(k)\boldsymbol{x}(k), \quad (2)$$

Where **W**(k) is the matrix with entries $w_{ij}(k) = 0\ if\ (i,j) \notin E$and $\sum_{j\in N_i\cup\{i\}} w_{ij}(k) = 1$.

The system is said to achieve the distributed consensus asymptotically if $\lim_{k\to\infty} \boldsymbol{x}(k) = \mu\boldsymbol{1}$, meaning that all nodes agreed on the value μ. When μ is equal to the average of the initial values, i.e. $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i(0)$, the system is said to achieve the average consensus, meaning that:

$$\lim_{k\to\infty} \boldsymbol{W}^k = \frac{\boldsymbol{1}}{\boldsymbol{N}}\boldsymbol{1}\boldsymbol{1}^{\boldsymbol{T}}. \quad (3)$$

The convergence conditions are described as follows:

Theorem 1[15]: Consider linear iteration protocol (2), the distributed consensus is achieved if and only if the weighted consensus matrix **W** satisfies the following conditions:

a. $\boldsymbol{W1} = \boldsymbol{1}$

b. $\boldsymbol{\rho}(\boldsymbol{W} - \boldsymbol{1c^T}) < \boldsymbol{1}$

Where $\rho(\boldsymbol{W} - \boldsymbol{1c^T})$ is the spectral radius of $\boldsymbol{W} - \boldsymbol{1c^T}$ and **c** is chosen so that $\boldsymbol{1c^T} = 1$.

Then, the weighted matrix has row-sum equal to 1 and 1 is a simple eigenvalue of **W** and that all other eigenvalues are strictly less than one in magnitude. It is said that the weighted matrix is a row-stochastic matrix.

Theorem 2 [15]: Equation (3) holds if and only if:

c. $\boldsymbol{1^T W} = \boldsymbol{1}$

d. $\boldsymbol{W1} = \boldsymbol{1}$

e. $\boldsymbol{\rho}\left(\boldsymbol{W} - \frac{\boldsymbol{1}}{\boldsymbol{N}}\boldsymbol{1}\boldsymbol{1^T}\right) < \boldsymbol{1}$

Meaning that **W** is a doubly stochastic matrix.

3.1.2. Continuous-time systems

We still consider a system modeled as a graph $G(V,E)$ with N agents, in which each agent has a value$x_i \in R$. In

[13] a continuous-time consensus protocol can be expressed as follows:

$$\dot{x}_i(t) = -\sum_{j \in J_i(t)} a_{ij}(t)\left(x_i(t) - x_j(t)\right), \quad (4)$$

Where $J_i(t)$ represents the set of agents whose information is available to agent i at time t and $a_{ij}(t)$ denotes a positive time-varying weighting factor. In other words, the process of calculation is implemented by the fact that the node just integrates its values or, the information state of each agent is driven toward the states of its neighbors at each time.

The protocol (4) can be expressed in matrix form as $\dot{\boldsymbol{x}} = -\boldsymbol{L}\boldsymbol{x},$ where **L** is the graph Laplacian and $\boldsymbol{x} = [x_1 \quad x_2 \quad \cdots \quad x_N]^T$.

3.1.3 Finite-time consensus problems [14]

In actual complicated systems, the execution time is getting more and more impact. Therefore, the purpose is now to design a finite-time average consensus algorithm allowing all nodes to reach the average consensus value in a finite number of steps D for self-configuration protocols, i.e.

$$\boldsymbol{x}(D) = \frac{1}{N}\boldsymbol{1}\boldsymbol{1}^{\boldsymbol{T}}\boldsymbol{x}(0) \quad (5)$$

Meaning that, we are about to design a set of consensus matrices $\boldsymbol{W}_1, \boldsymbol{W}_2, \ldots, \boldsymbol{W}_D$ such that $\prod_{i=D}^{1} \boldsymbol{W}_i = \frac{1}{N}\boldsymbol{1}\boldsymbol{1}^{\boldsymbol{T}}$.

3.2. Consensus Matrix Design

In the literature, there are some works devoted to the design of the weighted matrix **W** that satisfies the convergence conditions of the consensus protocols. For instance, in [15]:

A. Maximum-degree weights:

An approach to design the weighted matrix **W** in a graph with fixed topology consists of assigning a weight on each edge equal to the maximum-degree of the network, i.e.

$$w_{ij} = \begin{cases} \dfrac{1}{d_{max}+1} & if\ j \in N_i \\ 1 - \dfrac{d_i}{d_{max}+1} & if\ i = j \\ 0 & otherwise \end{cases}$$

Where $d_{max} = \max d_i \leq N$.

B. Metropolis weights:

The metropolis weights matrix **W** for a graph with a time-invariant topology is proposed with the entries:

$$w_{ij} = \begin{cases} \dfrac{1}{\max\{d_i, d_j\}+1} & if\ j \in N_i \\ 1 - \sum\limits_{j \neq i} w_{ij} & if\ i = j \\ 0 & otherwise \end{cases}$$

C. Constant edge weights:

This is widely model for the weight matrix in both time-varying and time-invariant topologies. The **W** is defined as follows:

$$w_{ij} = \begin{cases} \alpha & if\ j \in N_i \\ 1 - \alpha|N_i| & if\ i = j \\ 0 & otherwise \end{cases}$$

The weighted matrix can be expressed in matrix form as $\boldsymbol{W} = \boldsymbol{I}_N - \alpha\boldsymbol{L}$ with $\boldsymbol{I}_N$ being identity matrix.

D. In analysis of consensus problem, convergence rate is an important index that evaluates the performance of the proposed consensus protocol. Therefore, there are some works dealing with accelerating the rate of convergence of the consensus protocol by solving some optimization problems in centralized way.

In [15], the authors have proposed an optimization method to obtain the optimum weighted matrix **W** achieving the average consensus in linear time invariant topologies as the solution of a semi-definite convex programming.

$$\min_{\boldsymbol{W}} \quad \rho\left(\boldsymbol{W} - \frac{1}{N}\boldsymbol{1}\boldsymbol{1}^{\boldsymbol{T}}\right)$$
$$subject\ to \quad \boldsymbol{W} \in \boldsymbol{S}_{\boldsymbol{G}}, \quad \boldsymbol{1}^{\boldsymbol{T}}\boldsymbol{W} = \boldsymbol{1}^{\boldsymbol{T}}, \quad \boldsymbol{W}\boldsymbol{1} = \boldsymbol{1}$$

Where $\boldsymbol{W} \in \boldsymbol{S}_{\boldsymbol{G}}$ expresses the constraint on the sparsely pattern of the matrix **W** with the set $\boldsymbol{S}_{\boldsymbol{G}}$ defined as follows:

$$\boldsymbol{S}_{\boldsymbol{G}} = \{\boldsymbol{W} \in R^{N\times N} | w_{ij} = 0\ if\ (i,j) \notin E\ and\ i \neq j\}.$$

4. Conclusion

In this paper, we have reviewed the consensus protocols in the context of Multi-agent systems (MAS), in particular for Wireless Sensor Network (WSN). In addition, we have cited out some application domains of the consensus protocols that can be embedded in various important fields such as military, environment, health, automation control, and formation control, etc. Since the research on consensus is ongoing, this survey is waiting for future contributions to the literature.

Moreover, we have pointed out the general picture of consensus protocols and some designs of the consensus matrix. In fact, most applications of consensus are asymptotic convergence that is not suitable for these kinds of sophisticated distributed algorithms. A low asymptotic convergence cannot ensure the efficiency and the accuracy of the algorithm, which can lead to other unexpected effects. Therefore, the orientation of research now is shifted to the protocols that guarantee a minimal execution time. For this purpose, some works are dedicated to accelerate the convergence rate of the algorithms or finite-time consensus.

REFERENCES

[1] R. Olfati-Saber and J.S. Shamma. Consensus filters for sensor networks and distributed sensor fusion. In Proc. of the 44th IEEE Conference on Decision and Control and European Control Conference. CDC-ECC ’05, pages 6698–6703, Dec 2005.

[2] R. Olfati-Saber. Flocking for multi-agent dynamic systems: algorithms and theory. The IEEE Transactions on Automatic Control, 51(3): 401–420, March 2006.

[3] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis. Convergence in multi-agent coordination, consensus, and flocking. In Proc. of the 44th IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC ’05, pages 2996–3000, December 2005.

[4] D.V. Dimarogonas and K.J. Kyriakopoulos. Formation control and collision avoidance for multi-agent systems and a connection between formation infeasibility and flocking behavior. In Proc. of the 44th IEEE Conference on Decision and Control and European Control Conference. CDC-ECC ’05, pages 84–89, Dec 2005.

[5] V. Borkar and P.P. Varaiya. Asymptotic agreement in distributed estimation. IEEE Transactions on Automatic Control, 27(3): 650–655, Jun 1982.

[6] J. Cortes, S. Martinez, and F. Bullo. Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions. IEEE Transactions on Automatic Control, 51(8): 1289–1298, Aug 2006.

[7] R. Olfati-Saber. Distributed Kalman filtering for sensor networks. In Proc. of the 46th IEEE Conf. on Decision and Control, pages 5492–5498, New Orleans, LA, USA, December 12–14 2007.

[8] S. Bolognani, S. Del Favero, L. Schenato, and D. Varagnolo. Distributed sensor calibration and least-squares parameter identification in WSNs using consensus algorithms. In Proc. of 46th annual Allerton Conference, pages 1191–1198, Allerton House, UIUC, Illinois, USA, 2008.

[9] A.Y. Kibangou and A.L.F. de Almeida. Distributed PARAFAC based DS-CDMA blind receiver for wireless sensor networks. In Proc. of the IEEE Workshop SPAWC, Marrakech, Morocco, June 20-23 2010.

[10] S. V. Macua, P. Belanovic, and S. Zazo. Consensus-based distributed principal component analysis in wireless sensor networks. In Proc. of the IEEE Workshop SPAWC, Marrakech, Morocco, June 20-23 2010.

[11] A. Esna-Ashari, A.Y. Kibangou, and F. Garin. Distributed input and state estimation for linear discrete-time systems. In Proc. 51st IEEE Conf. on Decision and Control (CDC), Maui, Hawaii, USA, December 2012.

[12] S. Sundaram and C. Hadjcotis. Finite-time distributed consensus in graphs with time-invariant topologies. In Proc. of the American Conference on Control (ACC), pages 711–716, New York, USA, July 2007.

[13] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. IEEE Trans. on Automatic Control, vol. 49, pp. 1520-1533, Sep. 2004.

[14] A. Y. Kibangou. Step-size sequence design for finite-time average consensus in secure wireless sensor networks. Systems and Control Letters, 67:19–23, March 2014.

[15] L. Xiao, S. Boyd, and S. Kim. Distributed average consensus with least-mean-square deviation. Journal on Parallel Distributed Computation, 67(1): 33–46, January 2007.

(The Board of Editors received the paper on 14/04/2016, its review was completed on 25/04/2016)