Acta Universitatis Sapientiae

Informatica

Volume 13, Number 2, 2021

Sapientia Hungarian University of Transylvania
Scientia Publishing House

Acta Universitatis Sapientiae, Informatica is covered by the following services:

DOAJ (Directory of Open Access Journals)
EBSCO (relevant databases)
EBSCO Discovery Service
Japan Science and Technology Agency (JST)
Microsoft Academic
Ulrich's Periodicals Directory/ulrichsweb
Web of Science – Emerging Sources Citation Index
Zentralblatt für Mathematik

Contents

Acta Univ. Sapientiae, Informatica **13,** 2 (2021) 195–219

DOI: 10.2478/ausi-2021-0009

On A-energy and S-energy of certain class of graphs

Harishchandra S. RAMANE
Department of Mathematics
Karnatak University
Dharwad, India
email: hsramane@yahoo.com

K. ASHOKA
Department of Mathematics
Karnatak University
Dharwad, India
email: ashokagonal@gmail.com

B. PARVATHALU
Department of Mathematics
Karnatak University's
Karnatak Arts College
Dharwad, India
email: bparvathalu@gmail.com

Daneshwari PATIL
Department of Mathematics
Karnatak University
Dharwad, India
email: daneshwarip@gmail.com

Abstract. Let A and S be the adjacency and the Seidel matrix of a graph G respectively. A-energy is the ordinary energy $E(G)$ of a graph G defined as the sum of the absolute values of eigenvalues of A. Analogously, S-energy is the Seidel energy $E_S(G)$ of a graph G defined to be the sum of the absolute values of eigenvalues of the Seidel matrix S. In this article, certain class of A-equienergetic and S-equienergetic graphs are presented. Also some linear relations on A-energies and S-energies are given.

Computing Classification System 1998: G.2.2
Mathematics Subject Classification 2010: 05C50
Key words and phrases: graph energy, Seidel energy, equienergetic graphs, Seidel equienergetic graphs, energy linear relations, graph products

1 Introduction

Let G be a simple, finite and undirected graph of order n with vertex set $V(G) = \{v_1, v_2, \ldots, v_n\}$. The adjacency matrix $A = [a_{ij}]$ of G is a square matrix of order n whose (i,j)-th entry $a_{ij} = 1$ if v_i and v_j are adjacent and 0 otherwise. The eigenvalues $\theta_1, \theta_2, \ldots, \theta_n$ of A are called the A-eigenvalues of G and their collection is called the spectrum or A-spectrum of G. If $\theta_1, \theta_2, \ldots, \theta_k$ are the distinct A-eigenvalues of G of order n with respective multiplicities $m_1, m_2, \ldots, m_k$, then the A-spectrum of G is denoted by

$$\mathrm{Spec}(G) = \begin{pmatrix} \theta_1 & \theta_2 & \cdots & \theta_k \\ m_1 & m_2 & \cdots & m_k \end{pmatrix}, \text{ where } \sum_{j=1}^{k} m_j = n.$$

In 1966 J. H. van Lint and J. J. Seidel introduced real symmetric $\{0, \pm 1\}$-matrix called the Seidel matrix S is defined as $S = J - I - 2A$, where J is the matrix of order n whose all entries are equal to 1 and I is the identity matrix of order n. The eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ of S are called the Seidel eigenvalues or S-eigenvalues of G and their collection is called the Seidel spectrum or S-spectrum of G. If $\lambda_1, \lambda_2, \ldots, \lambda_k$ are the distinct S-eigenvalues of G of order n with respective multiplicities $m_1, m_2, \ldots, m_k$, then the Seidel spectrum or S-spectrum of G is denoted by

$$\mathrm{Spec}_S(G) = \begin{pmatrix} \lambda_1 & \lambda_2 & \cdots & \lambda_k \\ m_1 & m_2 & \cdots & m_k \end{pmatrix}, \text{ where } \sum_{j=1}^{k} m_j = n.$$

The number of positive and negative A-eigenvalues of G are denoted by n^+ and n^- respectively. The complement of a graph G is denoted by $\overline{G}$. A graph G is an r-regular graph if all its vertices have same degree equal to r. The line graph of G, denoted by $L(G)$ is a graph whose vertex set has one-to-one correspondence with the edge set of G and two vertices are adjacent in $L(G)$ if the corresponding edges are adjacent in G. For $k = 1, 2, \ldots$, the k-th iterated line graph of G is defined as $L^k(G) = L(L^{k-1}(G))$, where $L^0(G) = G$ and $L^1(G) = L(G)$ [10]. Let K_n be the complete graph of order n and $K_{n_1,n_2,\ldots,n_k}$ be the complete multipartite graph of order $n = \sum_{j=1}^{k} n_j$.
If $\theta_1 \geq \theta_2 \geq \cdots \geq \theta_n$ be the A-eigenvalues of G, then the energy or A-energy is defined as

$$E(G) = \sum_{j=1}^{n} |\theta_j| = 2\sum_{j=1}^{n^+} \theta_j = -2\sum_{j=1}^{n^-} \theta_{n-j+1}.$$

Several researchers have introduced many graph operations such as complement, disjoint union, join, graph products etc. The graph products Cartesian product, tensor product and strong product are known as the standard graph products and have been well studied in the graph theory. The energy of a graph introduced in 1978 [8] in connection with molecular chemistry and gained its own importance in the spectral graph theory.

Two graphs G_1 and G_2 of same order are said to be equienergetic or A-equienergetic if $E(G_1) = E(G_2)$. Similar to A-energy, the Seidel energy or S-energy $E_S(G)$ [9] of a graph G is defined as the sum of the absolute values of the eigenvalues of Seidel matrix S. Two graphs G_1 and G_2 of same order are said to be Seidel equienergetic or S-equienergetic if $E_S(G_1) = E_S(G_2)$. Numerous results dealing with the non-cospectral, A-equienergetic graphs have been appeared in the literature. Balakrishnan [2] and Stevanović [23] constructed A-equienergetic graphs using tensor product. Ramane and Walikar [20] and Liu and Liu [12] constructed A-equienergetic graphs by join of two graphs. Bonifácio et al. [3] and Ramane et al. [17] obtained some class of A-equienergetic graphs through Cartesian product, tensor product and strong product. Ramane et al. [21] obtained non-cospectral A-equienergetic iterated line graphs from regular graphs. For other results on A-equienergetic and S-equienergetic graphs one can see [1, 4, 7, 11, 12, 13, 14, 15, 18, 22, 24]. For other notation, terminology and results related to the spectra of graphs we follow [6]. One of the interesting and difficult problem in the study of energy of a graph in spectral graph theory is to find non-isomorphic graphs of same order with same energy. So for in the literature the linear relations on energies of two non isomorphic graphs are not well studied except A-equienergetic or S-equienergetic graphs. This motivates to find some class of graphs which satisfies the linear relations on energies of different graphs.

This article is organized as follows. In section 2, basic definitions, known results on A-eigenvalues of graph products, A-energy of a graph, S-eigenvalues and S-energy of a graph are presented. In section 3, certain class of A-equienergetic graphs are constructed and obtained some linear relations on the A-energies. In section 4, some class of the S-equienergetic graphs are constructed and obtained some linear relations on the S-energies.

2 Preliminaries

In this section, we shall list some known results which are needed in the next two sections.

Theorem 1 [6] *Let* G *be an* r*-regular graph of order* n *with the* A*-eigenvalues* $\theta_1 = r, \theta_2, \ldots, \theta_n$*. Then the* A*-eigenvalues of* $\overline{G}$ *are* $n-r-1, -\theta_2-1, \ldots, -\theta_n-1$*.*

Theorem 2 [6] *Let* G *be an* r*-regular graph on* n *vertices and* m *edges with the* A*-eigenvalues* $\theta_1 = r, \theta_2, \ldots, \theta_n$*. Then the* A*-eigenvalues of* $L(G)$ *are* θ_i+r-2*,* $i = 1, 2, \ldots, n$ *and* -2 $(m-n$ *times).*

Theorem 3 [5] *Let* G *is an* r*-regular graph of order* n *with the* A*-eigenvalues* $\theta_1 = r, \theta_2, \ldots, \theta_n$*. Then the eigenvalues of* S *are* $n-2r-1, -1-2\theta_2, \ldots, -1-2\theta_n$*.*

Theorem 4 [16] *Let* G_1 *and* G_2 *be two* r*-regular graphs of same order* n*,* $r \geq 3$*. Then for* $k \geq 2$*,* $L^k(G_1)$ *and* $L^k(G_2)$ *are* S*-equienergetic.*

The Cartesian product of two graphs G_1 and G_2 is the graph $G_1 \square G_2$ with vertex set $V(G_1) \times V(G_2)$, in which the vertices (u_1, u_2) and (v_1, v_2) are adjacent if either u_1 is adjacent to v_1 in G_1 and u_2 is equal to v_2 or u_1 is equal to v_1 and u_2 is adjacent to v_2 in G_2.

The tensor product of two graphs G_1 and G_2 is the graph $G_1 \otimes G_2$ with vertex set $V(G_1) \times V(G_2)$, in which the vertices (u_1, u_2) and (v_1, v_2) are adjacent if u_1 is adjacent to v_1 in G_1 and u_2 is adjacent to v_2 in G_2.

The strong product of two graphs G_1 and G_2 is the graph $G_1 \boxtimes G_2$ with vertex set $V(G_1) \times V(G_2)$, in which the vertices (u_1, u_2) and (v_1, v_2) are adjacent whenever u_1 and v_1 are equal or adjacent in G_1, and u_2 and v_2 are equal or adjacent in G_2. If G_1 and G_2 are two regular graphs then $G_1 \square G_2$, $G_1 \otimes G_2$ and $G_1 \boxtimes G_2$ are also regular graphs.

Lemma 5 [5] *If* $\mu_1, \mu_2, \ldots, \mu_n$ *are the* A*-eigenvalues of a graph* G_1 *and* $\sigma_1, \sigma_2, \ldots, \sigma_m$ *are the* A*-eigenvalues of a graph* G_2 *then*
(i) the A*-eigenvalues of* $G_1 \square G_2$ *are* $\mu_i + \sigma_j$*,* $i = 1, 2, \ldots, n$*;* $j = 1, 2, \ldots, m$*,*
(ii) the A*-eigenvalues of* $G_1 \otimes G_2$ *are* $\mu_i\sigma_j$*,* $i = 1, 2, \ldots, n$*;* $j = 1, 2, \ldots, m$*,*
(iii) the A*-eigenvalues of* $G_1 \boxtimes G_2$ *are* $\mu_i\sigma_j + \mu_i + \sigma_j$*,* $i = 1, 2, \ldots, n$*;* $j = 1, 2, \ldots, m$*.*

Lemma 6 [17] *The* A*-spectrum of the line graph of a complete bipartite graph* $K_{p,q}$*, where* $p, q \geq 2$ *is*

$$\mathrm{Spec}(L(K_{p,q})) = \begin{pmatrix} p+q-2 & p-2 & q-2 & -2 \\ 1 & q-1 & p-1 & (p-1)(q-1) \end{pmatrix}.$$

3 A-equienergetic graphs and linear relations on A-energies of certain class of graphs

3.1 A-equienergetic graphs

Theorem 7 *Let* G_1 *and* G_2 *be two* r*-regular* A*-equienergetic graphs of order* n*. Then for* $p \geq r$, $E(G_1 \Box K_{\underbrace{p,p,\ldots,p}_{k \text{ times}}}) = E(G_2 \Box K_{\underbrace{p,p,\ldots,p}_{k \text{ times}}})$.

Proof. We have, $\mathrm{Spec}(K_{\underbrace{p,p,\ldots,p}_{k \text{ times}}}) = \begin{pmatrix} p(k-1) & 0 & -p \\ 1 & k(p-1) & k-1 \end{pmatrix}$.

Let $\mathrm{Spec}(G_1) = \begin{pmatrix} r & \theta_2 & \ldots & \theta_k \\ 1 & m_2 & \ldots & m_k \end{pmatrix}$, where $1 + \sum_{j=2}^{k} m_j = n$.

By (i) of Lemma 5,

$$\mathrm{Spec}(G_1 \Box K_{\underbrace{p,p,\ldots,p}_{k \text{ times}}}) = \begin{pmatrix} r+p(k-1) & r & r-p & \theta_2+p(k-1) & \cdots \\ 1 & k(p-1) & k-1 & m_2 & \cdots \end{pmatrix}$$

$$\begin{pmatrix} \theta_k+p(k-1) & \theta_2 & \cdots & \theta_k & \theta_2-p & \cdots & \theta_k-p \\ m_k & km_2(p-1) & \cdots & km_k(p-1) & m_2(k-1) & \cdots & m_k(k-1) \end{pmatrix}.$$

Therefore

$E(G_1 \Box K_{\underbrace{p,p,\ldots,p}_{k \text{ times}}})$

$$\begin{aligned}
&= |r+p(k-1)| + |r|k(p-1) + |r-p|(k-1) + \sum_{i=2}^{k} m_i|\theta_i + p(k-1)| \\
&\quad + \sum_{i=2}^{k} km_i(p-1)|\theta_i| + \sum_{i=2}^{k} m_i(k-1)|\theta_i - p| \\
&= r + p(k-1) + kr(p-1) + (p-r)(k-1) + \sum_{i=2}^{k} m_i(\theta_i + p(k-1)) \\
&\quad + k(p-1)\sum_{i=2}^{k} m_i|\theta_i| + \sum_{i=2}^{k} m_i(k-1)(p-\theta_i) \\
&\qquad \text{since } \theta_i + p(k-1) \geq -r + p \geq 0 \text{ and } \theta_i - p \leq r - p \leq 0 \\
&= 2np(k-1) + k(p-1)E(G_1).
\end{aligned}$$

Since G_2 is also an r-regular graph of order n, we have

$$E(G_2 \Box K_{\underbrace{p,p,\ldots,p}_{k \text{ times}}}) = 2np(k-1) + k(p-1)E(G_2).$$

If G_1 and G_2 are A-equienergetic then $E(G_1 \Box K_{\underbrace{p,p,\ldots,p}_{k \text{ times}}}) = E(G_2 \Box K_{\underbrace{p,p,\ldots,p}_{k \text{ times}}})$ which completes the proof. □

Remark 8 *Recently in* [19] *Ramane et al. proved that* $E(G_1 \Box K_{p,p}) = E(G_2 \Box K_{p,p})$. *It is noted that this result becomes particular case of Theorem 7.*

Theorem 9 *If* $p \geq n \geq 2$ *and* $k \geq 2$ *then* $E(K_{\underbrace{p,p,\ldots,p}_{k \text{ times}}} \Box K_{n-1}) = E(K_{\underbrace{p-1,p-1,\ldots,p-1}_{k \text{ times}}} \Box K_n)$ *if and only if* $p = n$.

Proof. We have

$$\mathrm{Spec}(K_{\underbrace{p,p,\ldots,p}_{k \text{ times}}}) = \begin{pmatrix} p(k-1) & 0 & -p \\ 1 & k(p-1) & k-1 \end{pmatrix}$$

and

$$\mathrm{Spec}(K_{n-1}) = \begin{pmatrix} n-2 & -1 \\ 1 & n-2 \end{pmatrix}.$$

By (i) of Lemma 5,

$$\mathrm{Spec}(K_{\underbrace{p,p,\ldots,p}_{k \text{ times}}} \Box K_{n-1}) = \begin{pmatrix} pk-p+n-2 & pk-p-1 & n-2 & -1 & n-p-2 & -p-1 \\ 1 & n-2 & k(p-1) & k(n-2)(p-1) & k-1 & (k-1)(n-2) \end{pmatrix}.$$

If $p \geq n \geq 2$ then $pk-p+n-2, pk-p-1$ and $n-2$ are only the positive A-eigenvalues of $K_{\underbrace{p,p,\ldots,p}_{k \text{ times}}} \Box K_{n-1}$. Therefore from definition of A-energy, we get

$$\begin{aligned} &E(K_{\underbrace{p,p,\ldots,p}_{k \text{ times}}} \Box K_{n-1}) \\ &= 2[pk-p+n-2+(n-2)(pk-p-1)+k(n-2)(p-1)] \\ &= 2[2npk-3pk-np+p-nk+2k]. \end{aligned}$$

Next, we have

$$\mathrm{Spec}(K_{\underbrace{p-1,p-1,\ldots,p-1}_{k \text{ times}}}) = \begin{pmatrix} (p-1)(k-1) & 0 & -(p-1) \\ 1 & k(p-2) & k-1 \end{pmatrix}$$

and

$$\mathrm{Spec}(K_n) = \begin{pmatrix} n-1 & -1 \\ 1 & n-1 \end{pmatrix}.$$

By (i) of Lemma 5,
$\mathrm{Spec}(K_{\underbrace{p-1,p-1,\ldots,p-1}_{k \text{ times}}} \Box K_n)$

$$= \begin{pmatrix} pk-p+n-k & pk-p-k & n-1 \\ 1 & n-1 & k(p-2) \\ & & -1 \\ & & k(n-1)(p-2) \end{pmatrix} \quad \begin{matrix} & & \\ & & \\ n-p & -p \\ k-1 & (k-1)(n-1) \end{matrix}.$$

If $p \geq n \geq 2$ then $pk-p+n-k, pk-p-k$ and $n-1$ are only the positive A-eigenvalues of $K_{\underbrace{p-1,p-1,\ldots,p-1}_{k \text{ times}}} \Box K_n$. Therefore from definition of A-energy, we get
$E(K_{\underbrace{p-1,p-1,\ldots,p-1}_{k \text{ times}}} \Box K_n)$

$$\begin{aligned} &= 2[pk-p-k+n+(n-1)(pk-p-k)+k(n-1)(p-2)] \\ &= 2[2npk-3nk-np+n-pk+2k]. \end{aligned}$$

The graphs $K_{\underbrace{p,p,\ldots,p}_{k \text{ times}}} \Box K_{n-1}$ and $K_{\underbrace{p-1,p-1,\ldots,p-1}_{k \text{ times}}} \Box K_n$ are A-equienergetic if and only if

$$E(K_{\underbrace{p,p,\ldots,p}_{k \text{ times}}} \Box K_{n-1}) = E(K_{\underbrace{p-1,p-1,\ldots,p-1}_{k \text{ times}}} \Box K_n).$$

That is, $2[2npk-3pk-np+p-nk+2k] = 2[2npk-3nk-np+n-pk+2k]$ or $p+2nk = n+2pk$, which implies $(n-p)(2k-1) = 0$. Since k is positive integer, we get $p = n$. This completes the proof. □

Theorem 10 *Let* G *be the Petersen graph and* $p, q \geq 4$. *Then* $E\left(G\Box\overline{L(K_{p,q})}\right) = E\left(\overline{G}\Box L(K_{p,q})\right)$.

Proof. The A-spectrum of Petersen graph is

$$\mathrm{Spec}(G) = \begin{pmatrix} 3 & 1 & -2 \\ 1 & 5 & 4 \end{pmatrix}.$$

From Lemma 6 and Theorem 1, A-spectrum of $\overline{L(K_{p,q})}$ is

$$Spec\left(\overline{L(K_{p,q})}\right) = \begin{pmatrix} pq-p-q+1 & 1-p & 1-q & 1 \\ 1 & q-1 & p-1 & (p-1)(q-1) \end{pmatrix}.$$

Now by (i) of Lemma 5, A-spectrum of $G\Box\overline{L(K_{p,q})}$ is

$$\begin{pmatrix} pq-p-q+4 & 4-p & 4-q & 4 & pq-p-q+2 & 2-p \\ 1 & q-1 & p-1 & pq-p-q+1 & 5 & 5q-5 \end{pmatrix}$$
$$\begin{pmatrix} 2-q & 2 & pq-p-q-1 & -1-p & -1-q \\ 5p-5 & 5pq-5p-5q+5 & 4 & 4q-4 & 4p-4 \end{pmatrix}$$
$$\begin{pmatrix} -1 \\ 4pq-4p-4q+4 \end{pmatrix}.$$

If $p, q \geq 4$ then $pq-p-q+4, 4, pq-p-q+2, 2$ and $pq-p-q-1$ are only the positive A-eigenvalues of $G\Box\overline{L(K_{p,q})}$. Therefore from definition of A-energy

$$\begin{aligned} E\left(G\Box\overline{L(K_{p,q})}\right) &= 2[pq-p-q+4+4(pq-p-q-1)+5(pq-p-q+2) \\ &\quad + 2(5pq-5p-5q+5)+4(pq-p-q+1)] \\ &= 48(p-1)(q-1). \end{aligned}$$

Now

$$\mathrm{Spec}(\overline{G}) = \begin{pmatrix} 6 & 1 & -2 \\ 1 & 4 & 5 \end{pmatrix}.$$

By using (i) of Lemma 5 and Lemma 6, $\overline{G}\Box L(K_{p,q})$ has A-spectrum,

$$\begin{pmatrix} p+q+4 & p+4 & q+4 & 4 & p+q-4 & p-4 & q-4 \\ 1 & q-1 & p-1 & pq-p-q+1 & 5 & 5q-5 & 5p-5 \end{pmatrix}$$
$$\begin{pmatrix} -4 & p+q-1 & p-1 & q-1 & -1 \\ 5pq-5p-5q+5 & 4 & 4q-4 & 4p-4 & 4pq-4p-4q+4 \end{pmatrix}.$$

If $p, q \geq 4$ then -4 and -1 are only the negative A-eigenvalues of $\overline{G}\Box L(K_{p,q})$. Therefore from definition of A-energy

$$\begin{aligned} E(\overline{G}\Box L(K_{p,q})) &= 2[4(5pq - 5p - 5q + 5) + 4pq - 4p - 4q + 4] \\ &= 48(p-1)(q-1) \end{aligned}$$

which completes the proof. □

Recently in [17] Ramane et al. proved that $E(L(K_{p,q})) = E(\overline{L(K_{p,q})})$. In the following A-equienergetic graphs with the help of these graphs are given.

Theorem 11 *If* $p, q \geq 5$ *then* $E(L(K_{p,q})\Box L(K_4)) = E(\overline{L(K_{p,q})}\Box L(K_4))$.

Proof. The A-spectrum of K_4 is

$$\mathrm{Spec}(K_4) = \begin{pmatrix} 3 & -1 \\ 1 & 3 \end{pmatrix}.$$

From Theorem 2,

$$\mathrm{Spec}(L(K_4)) = \begin{pmatrix} 4 & 0 & -2 \\ 1 & 3 & 2 \end{pmatrix}.$$

By using (i) of Lemma 5 and Lemma 6, $L(K_{p,q})\Box L(K_4)$ has A-spectrum

$$\begin{pmatrix} p+q+2 & p+q-2 & p+q-4 & p+2 & p-2 & p-4 & q+2 & q-2 \\ 1 & 3 & 2 & q-1 & 3q-3 & 2q-2 & p-1 & 3p-3 \end{pmatrix}$$
$$\begin{pmatrix} q-4 & -4 & 2 & -2 \\ 2p-2 & 2pq-2p-2q+2 & pq-p-q+1 & 3pq-3p-3q+3 \end{pmatrix}.$$

If $p, q \geq 5$ then -4 and -2 are only the negative A-eigenvalues of $L(K_{p,q})\Box L(K_4)$. Therefore from definition of A-energy

$$\begin{aligned} E(L(K_{p,q})\Box L(K_4)) &= 2[4(2pq - 2p - 2q + 2) + 2(3pq - 3p - 3q + 3)] \\ &= 28(p-1)(q-1). \end{aligned}$$

By using Theorem 1, (i) of Lemma 5 and Lemma 6, $\overline{L(K_{p,q})}\Box L(K_4)$ has A-spectrum,

$$\begin{pmatrix} pq-p-q+5 & pq-p-q+1 & pq-p-q-1 & 5-p & 1-p & -1-p \\ 1 & 3 & 2 & q-1 & 3q-3 & 2q-2 \end{pmatrix}$$
$$\begin{pmatrix} 5-q & 1-q & -1-q & 5 & 1 \\ p-1 & 3p-3 & 2p-2 & pq-p-q+1 & 3pq-3p-3q+3 \end{pmatrix}$$
$$\begin{pmatrix} -1 \\ 2pq-2p-2q+2 \end{pmatrix}.$$

If $p, q \geq 5$ then $pq-p-q+5, pq-p-q+1, pq-p-q-1, 5$ and 1 are only the positive A-eigenvalues of $\overline{L(K_{p,q})}\Box L(K_4)$. Therefore from definition of A-energy

$$\begin{aligned} E(\overline{L(K_{p,q})}\Box L(K_4)) &= 2[pq-p-q+5+3(pq-p-q+1)+2(pq-p-q-1) \\ &\quad +5(pq-p-q+1)+3pq-3p-3q+3] \\ &= 28(p-1)(q-1) \end{aligned}$$

which completes the proof. □

Theorem 12 *If* $3 \leq p, q \leq 6$ *then* $E(\overline{L(K_{p,q})\Box L(K_4)}) = E(\overline{\overline{L(K_{p,q})}\Box L(K_4)})$.

Proof. We have

$$\mathrm{Spec}(L(K_4)) = \begin{pmatrix} 4 & 0 & -2 \\ 1 & 3 & 2 \end{pmatrix}.$$

By using (i) of Lemma 5, Lemma 6 and Theorem 1, $\overline{L(K_{p,q})\Box L(K_4)}$ has A-spectrum

$$\begin{pmatrix} 6pq-p-q-3 & -1-p-q+2 & -1-p-q+4 & -1-p-2 & -1-p+2 \\ 1 & 3 & 2 & q-1 & 3q-3 \end{pmatrix}$$

$$\begin{pmatrix} -1-p+4 & -1-q-2 & -1-q+2 & -1-q+4 & -3 \\ 2q-2 & p-1 & 3p-3 & 2p-2 & pq-p-q+1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 3 \\ 3pq-3p-3q+3 & 2pq-2p-2q+2 \end{pmatrix}.$$

If $p, q \geq 3$ then $6pq-p-q-3, 1$ and 3 are only the positive A-eigenvalues of $\overline{L(K_{p,q})\Box L(K_4)}$. Therefore from definition of A-energy, we have $E(\overline{L(K_{p,q})\Box L(K_4)})$

$$\begin{aligned} &= 2[6pq-p-q-3+1(3pq-3p-3q+3)+3(2pq-2p-2q+2)] \\ &= 2(15pq-10p-10q+6). \end{aligned}$$

By using (i) of Lemma 5, Lemma 6 and Theorem 1, $\overline{\overline{L(K_{p,q})}\Box L(K_4)}$ has A-spectrum

$$\begin{pmatrix} 5pq+p+q-6 & p+q-pq-2 & p+q-pq & p-6 & p-2 & p \\ 1 & 3 & 2 & q-1 & 3q-3 & 2q-2 \end{pmatrix}$$

$$\begin{pmatrix} q-6 & q-2 & q & -6 & -2 \\ p-1 & 3p-3 & 2p-2 & pq-p-q+1 & 3pq-3p-3q+3 \end{pmatrix}$$

$$\begin{pmatrix} 0 \\ 2pq-2p-2q+2 \end{pmatrix}.$$

If $3 \leq p, q \leq 6$ then $\overline{5pq + p + q - 6, p - 2, p, q - 2}$ and q are only the positive A-eigenvalues of $\overline{\overline{L(K_{p,q})}\Box L(K_4)}$. Therefore from definition of A-energy

$$\begin{aligned} E(\overline{\overline{L(K_{p,q})}\Box L(K_4)}) &= 2[5pq + p + q - 6 + (p-2)(3q-3) + p(2q-2) \\ &\quad + (q-2)(3p-3) + q(2p-2)] \\ &= 2(15pq - 10p - 10q + 6) \end{aligned}$$

which completes the proof. □

Theorem 13 *If* $p, q \geq 3$ *then* $E(L(K_{p,q})\Box\overline{L(K_4)}) = E(\overline{L(K_{p,q})}\Box\overline{L(K_4)})$.

Proof. We have

$$\mathrm{Spec}(L(K_4)) = \begin{pmatrix} 4 & 0 & -2 \\ 1 & 3 & 2 \end{pmatrix}$$

and from Theorem 1

$$\mathrm{Spec}(\overline{L(K_4)}) = \begin{pmatrix} 1 & -1 \\ 3 & 3 \end{pmatrix}.$$

By using (i) of Lemma 5 and Lemma 6, $L(K_{p,q})\Box\overline{L(K_4)}$ has A-spectrum

$$\begin{pmatrix} p+q-1 & p+q-3 & p-1 & p-3 & q-1 & q-3 \\ 3 & 3 & 3q-3 & 3q-3 & 3p-3 & 3p-3 \\ -1 & & -3 & & & \\ 3pq-3p-3q+3 & & 3pq-3p-3q+3 & & & \end{pmatrix}.$$

If $p, q \geq 3$ then -1 and -3 are only the negative A-eigenvalues of $L(K_{p,q})\Box\overline{L(K_4)}$. Therefore from definition of A-energy

$$\begin{aligned} E(L(K_{p,q})\Box\overline{L(K_4)}) &= 2[3(p-1)(q-1) + 9(p-1)(q-1)] \\ &= 24(p-1)(q-1). \end{aligned}$$

By using Lemma 6, Theorem 1 and (i) of Lemma 5, $\overline{L(K_{p,q})}\Box\overline{L(K_4)}$ has A-spectrum

$$\begin{pmatrix} pq-p-q+2 & pq-p-q & 2-p & -p & 2-q & -q \\ 3 & 3 & 3q-3 & 3q-3 & 3p-3 & 3p-3 \\ 2 & & 0 & & & \\ 3pq-3p-3q+3 & & 3pq-3p-3q+3 & & & \end{pmatrix}.$$

If $p, q \geq 3$ then $pq-p-q+2, pq-p-q$ and 2 are only the positive A-eigenvalues of $\overline{L(K_{p,q})}\Box\overline{L(K_4)}$. Therefore from definition of A-energy

$$\begin{aligned} E(\overline{L(K_{p,q})}\Box\overline{L(K_4)}) &= 2[3(pq-p-q+2)+3(pq-p-q) \\ &\quad +2(3pq-3p-3q+3)] \\ &= 24(p-1)(q-1) \end{aligned}$$

which completes the proof. □

Theorem 14 *If* $p, q \geq 4$ *then* $E(\overline{L(K_{p,q})}\Box C_6) = E\left(L(K_{p,q})\Box\overline{C_6}\right)$ *where* C_6 *is the cycle of order* 6.

Proof. We have

$$\mathrm{Spec}(C_6) = \begin{pmatrix} 2 & 1 & -1 & -2 \\ 1 & 2 & 2 & 1 \end{pmatrix}.$$

By using Lemma 6, Theorem 1 and (i) of Lemma 5, A-spectrum of $\overline{L(K_{p,q})}\Box C_6$ is

$$\begin{pmatrix} pq-p-q+3 & pq-p-q+2 & pq-p-q & pq-p-q-1 & 3-p \\ 1 & 2 & 2 & 1 & q-1 \end{pmatrix}$$
$$\begin{pmatrix} 2-p & -p & -1-p & 3-q & 2-q & -q & -1-q & 3 \\ 2q-2 & 2q-2 & q-1 & p-1 & 2p-2 & 2p-2 & p-1 & pq-p-q+1 \end{pmatrix}$$
$$\begin{pmatrix} 2 & 0 & -1 \\ 2pq-2p-2q+2 & 2pq-2p-2q+2 & pq-p-q+1 \end{pmatrix}.$$

If $p, q \geq 4$ then $pq-p-q+3, pq-p-q+2, pq-p-q, pq-p-q-1, 3$ and 2 are only the positive A-eigenvalues of $\overline{L(K_{p,q})}\Box C_6$. Therefore from definition of A-energy

$E\left(\overline{L(K_{p,q})}\Box C_6\right)$

$$\begin{aligned} &= 2[pq-p-q+3+2(pq-p-q+2)+2(pq-p-q) \\ &+ (pq-p-q-1)+3(pq-p-q+1)+2(2pq-2p-2q+2)] \\ &= 26(p-1)(q-1). \end{aligned}$$

Now by Theorem 1

$$\mathrm{Spec}(\overline{C_6}) = \begin{pmatrix} 3 & 1 & 0 & -2 \\ 1 & 1 & 2 & 2 \end{pmatrix}.$$

By using Lemma 6 and (i) of Lemma 5 $L(K_{p,q})\square\overline{C_6}$ has A-spectrum

$$\begin{pmatrix} p+q+1 & p+q-1 & p+q-2 & p+q-4 & p+1 & p-1 & p-2 & p-4 \\ 1 & 1 & 2 & 2 & q-1 & q-1 & 2q-2 & 2q-2 \end{pmatrix}$$
$$\begin{pmatrix} q+1 & q-1 & q-2 & q-4 & 1 & -1 \\ p-1 & p-1 & 2p-2 & 2p-2 & pq-p-q+1 & pq-p-q+1 \end{pmatrix}$$
$$\begin{pmatrix} -2 & -4 \\ 2pq-2p-2q+2 & 2pq-2p-2q+2 \end{pmatrix}.$$

If $p, q \geq 4$ then $-1, -2$ and -4 are only the negative A-eigenvalues of $L(K_{p,q})\square\overline{C_6}$. Therefore from definition of A-energy

$$\begin{aligned} E\left(L(K_{p,q})\square\overline{C_6}\right) &= 2[pq-p-q+1+2(2pq-2p-2q+2) \\ &\quad +4(2pq-2p-2q+2)] \\ &= 26(p-1)(q-1) \end{aligned}$$

which completes the proof. □

Theorem 15 *If* $p, q \geq 3$ *then* $E\left(L(K_{p,q})\square\overline{W_5}\right) = E(\overline{L(K_{p,q})}\square\overline{W_5})$ *where* W_5 *is the wheel of order* 5.

Proof. Since A-spectrum of $\overline{W_5}$ is

$$\begin{pmatrix} 1 & 0 & -1 \\ 2 & 1 & 2 \end{pmatrix}.$$

Now by (i) of Lemma 5, A-spectrum of $L(K_{p,q})\square\overline{W_5}$ is

$$\begin{pmatrix} p+q-1 & p+q-2 & p+q-3 & p-1 & p-2 & p-3 & q-1 & q-2 \\ 2 & 1 & 2 & 2q-2 & q-1 & 2q-2 & 2p-2 & p-1 \end{pmatrix}$$
$$\begin{pmatrix} q-3 & -1 & -2 & -3 \\ 2p-2 & 2pq-2p-2q+2 & pq-p-q+1 & 2pq-2p-2q+2 \end{pmatrix}.$$

If $p, q \geq 3$ then $-1, -2$ and -3 are only the negative A-eigenvalues of $L(K_{p,q})\square\overline{W_5}$. Therefore from definition of A-energy

$$\begin{aligned} E\left(L(K_{p,q})\square\overline{W_5}\right) &= 2[2pq-2p-2q+2+2(pq-p-q+1) \\ &\quad +3(2pq-2p-2q+2)] \\ &= 20(p-1)(q-1). \end{aligned}$$

By using Lemma 6, Theorem 1 and (i) of Lemma 5, A-spectrum of $\overline{L(K_{p,q})}\Box\overline{W_5}$ is

$$\begin{pmatrix} pq-p-q+2 & pq-p-q+1 & pq-p-q & 2-p & 1-p & -p & 2-q \\ 2 & 1 & 2 & 2q-2 & q-1 & 2q-2 & 2p-2 \\ 1-q & -q & 2 & 1 & 0 & & \\ p-1 & 2p-2 & 2pq-2p-2q+2 & pq-p-q+1 & 2pq-2p-2q+2 & & \end{pmatrix}.$$

If $p, q \geq 3$ then $pq-p-q+2, pq-p-q+1, pq-p-q, 2$ and 1 are only the positive A-eigenvalues of $\overline{L(K_{p,q})}\Box\overline{W_5}$. Therefore from definition of A-energy

$$\begin{aligned} E\left(\overline{L(K_{p,q})}\Box\overline{W_5}\right) &= 2[2(pq-p-q+2)+pq-p-q+1+2(pq-p-q) \\ &\quad +2(2pq-2p-2q+2)+(pq-p-q+1)] \\ &= 20(p-1)(q-1) \end{aligned}$$

which completes the proof. □

Theorem 16 *If* $n \geq 3$ *then* $E(\overline{K_{n,n} \otimes K_{n-1}}) = E(\overline{K_{n-1,n-1} \otimes K_n})$.

Proof. We have

$$\mathrm{Spec}(K_{n,n}) = \begin{pmatrix} n & 0 & -n \\ 1 & 2n-2 & 1 \end{pmatrix}$$

and

$$\mathrm{Spec}(K_{n-1}) = \begin{pmatrix} n-2 & -1 \\ 1 & n-2 \end{pmatrix}.$$

By using (ii) of Lemma 5 and Theorem 1,

$$\mathrm{Spec}(\overline{K_{n,n} \otimes K_{n-1}}) = \begin{pmatrix} n^2-1 & -1-n & -1 & n(n-2)-1 & n-1 \\ 1 & n-2 & (n-1)(2n-2) & 1 & n-2 \end{pmatrix}.$$

If $n \geq 3$ then $n^2-1, n(n-2)-1$ and $n-1$ are only the positive A-eigenvalues of $\overline{K_{n,n} \otimes K_{n-1}}$ and from definition of A-energy

$$\begin{aligned} E\left(\overline{K_{n,n} \otimes K_{n-1}}\right) &= 2[n^2-1-1+n(n-2)+(n-1)(n-2)] \\ &= 6n^2-10n. \end{aligned}$$

Now

$$\mathrm{Spec}(K_{n-1,n-1}) = \begin{pmatrix} n-1 & 0 & -(n-1) \\ 1 & 2n-4 & 1 \end{pmatrix}$$

and

$$\mathrm{Spec}(K_n) = \begin{pmatrix} n-1 & -1 \\ 1 & n-1 \end{pmatrix}.$$

By using (ii) of Lemma 5 and Theorem 1,

$$\mathrm{Spec}(\overline{K_{n-1,n-1} \otimes K_n}) = \begin{pmatrix} n^2-2 & -1+(n-1) & -1 & -1 & (n-1)^2-1 & -1-(n-1) \\ 1 & n-1 & 2n-4 & (2n-4)(n-1) & 1 & n-1 \end{pmatrix}.$$

If $n \geq 3$ then $n^2-2, n-2$ and $(n-1)^2-1$ are only the positive A-eigenvalues of $\overline{K_{n-1,n-1} \otimes K_n}$. Therefore from definition of A-energy

$$\begin{aligned} E\left(\overline{K_{n-1,n-1} \otimes K_n}\right) &= 2[n^2-2+(n-1)(n-2)+(n-1)^2-1] \\ &= 6n^2-10n \end{aligned}$$

which completes the proof. □

3.2 Linear relations on energies of graphs

Linear relations on energies of different graphs are not yet well studied in the study of graph energies except equienergetic graphs, that is $E(G_1)-E(G_2)=0$. In the following we present some linear relations on energies of different graphs of same order of the type $aE(G_1)+bE(G_2)=c$, where a, b and c are real numbers.

Theorem 17 *If* $n \geq 3$ *then* $E\left(\overline{L(K_4)\Box K_n}\right) - E\left(L(K_4)\Box K_n\right) = 2.$

Proof. We have

$$\mathrm{Spec}(L(K_4)) = \begin{pmatrix} 4 & 0 & -2 \\ 1 & 3 & 2 \end{pmatrix}$$

and

$$\mathrm{Spec}(K_n) = \begin{pmatrix} n-1 & -1 \\ 1 & n-1 \end{pmatrix}.$$

By using (i) of Lemma 5,

$$\mathrm{Spec}(L(K_4)\Box K_n) = \begin{pmatrix} n+3 & n-1 & n-3 & 3 & -1 & -3 \\ 1 & 3 & 2 & n-1 & 3n-3 & 2n-2 \end{pmatrix}.$$

If $n \geq 3$ then -1 and -3 are only the negative A-eigenvalues of $L(K_4)\Box K_n$. Therefore from definition of A-energy

$$\begin{aligned} E\left(L(K_4)\Box K_n\right) &= 2[(3n-3)+3(2n-2)] \\ &= 18n-18. \end{aligned} \tag{1}$$

By using Theorem 1

$$\mathrm{Spec}(\overline{L(K_4)\Box K_n}) = \begin{pmatrix} 5n-4 & -n & 2-n & -4 & 0 & 2 \\ 1 & 3 & 2 & n-1 & 3n-3 & 2n-2 \end{pmatrix}.$$

If $n \geq 3$ then $-n, 2-n$ and -4 are only the negative A-eigenvalues of $\overline{L(K_4)\Box K_n}$. Therefore from definition of A-energy

$$\begin{aligned} E\left(\overline{L(K_4)\Box K_n}\right) &= 2[3n + 2n - 4 + 4n - 4] \\ &= 18n - 16. \end{aligned} \tag{2}$$

From (1) and (2) the result follows. □

Theorem 18 *If* $n \geq 3$ *then* $E(K_{n-1,n-1} \otimes K_n) - E(K_{n,n} \otimes K_{n-1}) = 4$.

Proof. We have

$$\mathrm{Spec}(K_{n,n}) = \begin{pmatrix} n & 0 & -n \\ 1 & 2n-2 & 1 \end{pmatrix}$$

and

$$\mathrm{Spec}(K_{n-1}) = \begin{pmatrix} n-2 & -1 \\ 1 & n-2 \end{pmatrix}.$$

By using (ii) of Lemma 5,
$\mathrm{Spec}(K_{n,n} \otimes K_{n-1}) =$
$$\begin{pmatrix} n(n-2) & -n & 0 & 0 & -n(n-2) & n \\ 1 & n-2 & 2n-2 & (n-2)(2n-2) & 1 & n-2 \end{pmatrix}.$$
If $n \geq 3$ then $n(n-2)$ and n are only the positive A-eigenvalues of $K_{n,n} \otimes K_{n-1}$. Therefore from definition of A-energy

$$\begin{aligned} E(K_{n,n} \otimes K_{n-1}) &= 2[n(n-2) + n(n-2)] \\ &= 4n^2 - 8n. \end{aligned} \tag{3}$$

Now

$$\mathrm{Spec}(K_{n-1,n-1}) = \begin{pmatrix} n-1 & 0 & -(n-1) \\ 1 & 2n-4 & 1 \end{pmatrix}$$

and

$$\mathrm{Spec}(K_n) = \begin{pmatrix} n-1 & -1 \\ 1 & n-1 \end{pmatrix}.$$

By using (ii) of Lemma 5,
$\mathrm{Spec}(K_{n-1,n-1} \otimes K_n) =$

$$\begin{pmatrix} (n-1)^2 & -(n-1) & 0 & 0 & -(n-1)^2 & n-1 \\ 1 & n-1 & 2n-4 & (n-2)(2n-2) & 1 & n-1 \end{pmatrix}.$$

If $n \geq 3$ then $(n-1)^2$ and $n-1$ are only the positive A-eigenvalues of $K_{n-1,n-1} \otimes K_n$. Therefore from definition of A-energy

$$\begin{aligned} E(K_{n-1,n-1} \otimes K_n) &= 2[(n-1)^2 + (n-1)^2] \\ &= 4n^2 - 8n + 4. \end{aligned} \tag{4}$$

From (3) and (4) result follows. □

Theorem 19 *If* $n \geq 3$ *then*

$$E(K_{\underbrace{n-1, n-1, \ldots, n-1}_{k \text{ times}}} \boxtimes K_n) - E(K_{\underbrace{n, n, \ldots, n}_{k \text{ times}}} \boxtimes K_{n-1}) = 2.$$

Proof. We have

$$\mathrm{Spec}(K_{\underbrace{n, n, \ldots, n}_{k \text{ times}}}) = \begin{pmatrix} n(k-1) & 0 & -n \\ 1 & k(n-1) & k-1 \end{pmatrix}$$

and

$$\mathrm{Spec}(K_{n-1}) = \begin{pmatrix} n-2 & -1 \\ 1 & n-2 \end{pmatrix}.$$

By using (iii) of Lemma 5,

$\mathrm{Spec}(K_{\underbrace{n, n, \ldots, n}_{k \text{ times}}} \boxtimes K_{n-1}) =$

$$\begin{pmatrix} n^2k - n^2 - nk + 2n - 2 & n-2 & -1 & -n^2 + 2n - 2 \\ 1 & k(n-1) & kn(n-2) & k-1 \end{pmatrix}.$$

If $n \geq 3$ then $n^2k - n^2 - nk + 2n - 2$ and $n-2$ are only the positive A-eigenvalues of $K_{\underbrace{n, n, \ldots, n}_{k \text{ times}}} \boxtimes K_{n-1}$. Therefore from definition of A-energy

$$\begin{aligned} E(K_{\underbrace{n, n, \ldots, n}_{k \text{ times}}} \boxtimes K_{n-1}) &= 2[n^2k - n^2 - nk + 2n - 2 + k(n-2)(n-1)] \\ &= 4n^2k - 2n^2 - 8nk + 4n + 4k - 4. \end{aligned} \tag{5}$$

Now

$$\mathrm{Spec}(K_{\underbrace{n-1, n-1, \ldots, n-1}_{k \text{ times}}}) = \begin{pmatrix} (n-1)(k-1) & 0 & -(n-1) \\ 1 & k(n-2) & k-1 \end{pmatrix}$$

and

$$\mathrm{Spec}(K_n) = \begin{pmatrix} n-1 & -1 \\ 1 & n-1 \end{pmatrix}.$$

By using (iii) of Lemma 5,
$\mathrm{Spec}(K_{\underbrace{n-1, n-1, \ldots, n-1}_{k \text{ times}}} \boxtimes K_n) =$

$$\begin{pmatrix} n^2k - n^2 - nk + 2n - 1 & n-1 & -1 & -n^2 + 2n - 1 \\ 1 & k(n-2) & k(n-1)^2 & k-1 \end{pmatrix}.$$

If $n \geq 3$ then $n^2k - n^2 - nk + 2n - 1$ and $n-1$ are only the positive A-eigenvalues of $K_{\underbrace{n-1, n-1, \ldots, n-1}_{k \text{ times}}} \boxtimes K_n$. Therefore from definition of A-energy

$$\begin{aligned} E(K_{\underbrace{n-1, n-1, \ldots, n-1}_{k \text{ times}}} \boxtimes K_n) &= 2[n^2k - n^2 - nk + 2n - 1 + k(n-2)(n-1)] \\ &= 4n^2k - 2n^2 - 8nk + 4n + 4k - 2. \end{aligned} \tag{6}$$

From (5) and (6) the result follows. □

Theorem 20 *If* $n \geq 3$ *then* $E(\overline{K_{n,n} \boxtimes K_{n-1}}) - E(\overline{K_{n-1,n-1} \boxtimes K_n}) = 4$.

Proof. We have

$$\mathrm{Spec}(K_{n,n}) = \begin{pmatrix} n & 0 & -n \\ 1 & 2n-2 & 1 \end{pmatrix}$$

and

$$\mathrm{Spec}(K_{n-1}) = \begin{pmatrix} n-2 & -1 \\ 1 & n-2 \end{pmatrix}.$$

By using (iii) of Lemma 5 and Theorem 1,

$$\mathrm{Spec}(\overline{K_{n,n} \boxtimes K_{n-1}}) = \begin{pmatrix} n^2 - 2n + 1 & 0 & -n + 1 \\ 2 & 2n^2 - 2n & 2n - 2 \end{pmatrix}.$$

If $n \geq 3$ then $n^2 - 2n + 1$ is only the positive A-eigenvalues of $\overline{K_{n,n} \boxtimes K_{n-1}}$. Therefore from definition of A-energy

$$\begin{aligned} E\left(\overline{K_{n,n} \boxtimes K_{n-1}}\right) &= 2[2n^2 - 4n + 2] \\ &= 4n^2 - 8n + 4. \end{aligned} \tag{7}$$

Now

$$\mathrm{Spec}(K_{n-1,n-1}) = \begin{pmatrix} n-1 & 0 & -(n-1) \\ 1 & 2n-4 & 1 \end{pmatrix}$$

and

$$\mathrm{Spec}(K_n) = \begin{pmatrix} n-1 & -1 \\ 1 & n-1 \end{pmatrix}.$$

By using (iii) of Lemma 5 and Theorem 1,

$$\mathrm{Spec}(\overline{K_{n-1,n-1} \boxtimes K_n}) = \begin{pmatrix} n^2-2n & 0 & -n \\ 2 & 2n^2-4n+2 & 2n-4 \end{pmatrix}.$$

If $n \geq 3$ then $n^2 - 2n$ is only the positive A-eigenvalues of $\overline{K_{n-1,n-1} \boxtimes K_n}$. Therefore from definition of A-energy

$$\begin{aligned} E\left(\overline{K_{n-1,n-1} \boxtimes K_n}\right) &= 2[2n^2 - 4n] \\ &= 4n^2 - 8n. \end{aligned} \tag{8}$$

From (7) and (8) the result follows. □

4 S-equienergetic graphs and linear relations on S-energies of certain class of graphs

4.1 S-equienergetic graphs

In [16] Ramane et al. studied S-energy of $L^2(G)$ for an r-regular graph G, $r \geq 3$ and constructed a large class of S-equienergetic graphs. The following result provides S-equienergetic graphs with the help of iterated line graphs $L^k(G)$ even for $k \geq 1$, where $L^0(G) = G$.

Theorem 21 *If* $n \geq 5$, $k \geq 0$ *then* $E_S(L^k(K_{n,n} \square K_{n-1})) = E_S(L^k(K_{n-1,n-1} \square K_n))$.

Proof. As $K_{n,n} \square K_{n-1}$ and $K_{n-1,n-1} \square K_n$ are both regular graphs of same order and of same degree, by Theorems 3 and 4, the result is true for $k \geq 2$. Now, it is enough to prove for $k = 0, 1$.
When $k = 0$.

$$\mathrm{Spec}(K_{n,n}) = \begin{pmatrix} n & 0 & -n \\ 1 & 2n-2 & 1 \end{pmatrix}$$

and

$$\mathrm{Spec}(K_{n-1}) = \begin{pmatrix} n-2 & -1 \\ 1 & n-2 \end{pmatrix}.$$

Therefore by (i) of Lemma 5,

$$\mathrm{Spec}(K_{n,n}\Box K_{n-1}) = \begin{pmatrix} 2n-2 & n-1 & n-2 & -1 & -2 & -n-1 \\ 1 & n-2 & 2n-2 & (2n-2)(n-2) & 1 & n-2 \end{pmatrix}$$

and

$$\mathrm{Spec}(K_{n-1,n-1}\Box K_n) = \begin{pmatrix} 2n-2 & n-2 & n-1 & -1 & 0 & -n \\ 1 & n-1 & 2n-4 & (2n-4)(n-1) & 1 & n-1 \end{pmatrix}.$$

Therefore by Theorem 3,
$\mathrm{Spec}_S(K_{n,n}\Box K_{n-1}) =$

$$\begin{pmatrix} 2n^2-6n+3 & 1-2n & 3-2n & 1 & 3 & 2n+1 \\ 1 & n-2 & 2n-2 & (2n-2)(n-2) & 1 & n-2 \end{pmatrix}$$

and $\mathrm{Spec}_S(K_{n-1,n-1}\Box K_n) =$

$$\begin{pmatrix} 2n^2-6n+3 & 3-2n & 1-2n & 1 & -1 & 2n-1 \\ 1 & n-1 & 2n-4 & (2n-4)(n-1) & 1 & n-1 \end{pmatrix}.$$

If $n \geq 5$ then $2n^2-6n+3, 1, 3$ and $2n+1$ are only the positive S-eigenvalues of $K_{n,n}\Box K_{n-1}$. Therefore from definition of S-energy

$$\begin{aligned} E_S(K_{n,n}\Box K_{n-1}) &= 2[2n^2-6n+3+(2n-2)(n-2)+3+(n-2)(2n+1)] \\ &= 12n^2-26n+16. \end{aligned}$$

If $n \geq 5$ then $2n^2-6n+3, 1$ and $2n-1$ are only the positive S-eigenvalues of $K_{n-1,n-1}\Box K_n$. Therefore from definition of S-energy

$$\begin{aligned} E_S(K_{n-1,n-1}\Box K_n) &= 2[2n^2-6n+3+(2n-4)(n-1)+(n-1)(2n-1)] \\ &= 12n^2-26n+16. \end{aligned}$$

Hence $E_S(K_{n,n}\Box K_{n-1}) = E_S(K_{n-1,n-1}\Box K_n)$.

When $k = 1$ both $K_{n,n}\Box K_{n-1}$ and $K_{n-1,n-1}\Box K_n$ are regular graphs of same order $2n(n-1)$ and of same degree $2n-2$. Hence by (i) of Lemma 5,
$\mathrm{Spec}(L(K_{n,n}\Box K_{n-1})) =$

$$\begin{pmatrix} 4n-6 & 3n-5 & 3n-6 & 2n-5 & 2n-6 & n-5 & -2 \\ 1 & n-2 & 2n-2 & (2n-2)(n-2) & 1 & n-2 & 2n(n-1)(n-2) \end{pmatrix}$$

and $\mathrm{Spec}(L(K_{n-1,n-1}\square K_n)) =$

$$\begin{pmatrix} 4n-6 & 3n-6 & 3n-5 & 2n-5 & 2n-4 & n-4 & -2 \\ 1 & n-1 & 2n-4 & (2n-4)(n-1) & 1 & n-1 & 2n(n-1)(n-2) \end{pmatrix}.$$

Therefore from Theorem 3,
$\mathrm{Spec}_S(L(K_{n,n}\square K_{n-1})) =$

$$\begin{pmatrix} 2n(n-1)^2-8n+11 & 9-6n & 11-6n & 9-4n & & & \\ 1 & n-2 & 2n-2 & (2n-2)(n-2) & & & \\ & & & & 11-4n & 9-2n & 3 \\ & & & & 1 & n-2 & 2n(n-1)(n-2) \end{pmatrix}$$

and $\mathrm{Spec}_S(L(K_{n-1,n-1}\square K_n)) =$

$$\begin{pmatrix} 2n(n-1)^2-8n+11 & 11-6n & 9-6n & 9-4n & & & \\ 1 & n-1 & 2n-4 & (2n-4)(n-1) & & & \\ & & & & 7-4n & 7-2n & 3 \\ & & & & 1 & n-1 & 2n(n-1)(n-2) \end{pmatrix}.$$

If $n \geq 5$ then $2n(n-1)^2-8n+11$ and 3 are only the positive S-eigenvalues of $L(K_{n,n}\square K_{n-1})$. Therefore from definition of S-energy

$$\begin{aligned} E_S(L(K_{n,n}\square K_{n-1})) &= 2[2n(n-1)^2-8n+11+6n(n^2-3n+2)] \\ &= 16n^3-44n^2+18n+22. \end{aligned}$$

If $n \geq 5$ then $2n(n-1)^2-8n+11$ and 3 are only the positive S-eigenvalues of $L(K_{n,n}\square K_{n-1})$. Therefore from definition of S-energy

$$\begin{aligned} E_S(L(K_{n,n}\square K_{n-1})) &= 2[2n(n-1)^2-8n+11+6n(n^2-3n+2)] \\ &= 16n^3-44n^2+18n+22. \end{aligned}$$

Hence $E_S(L(K_{n,n}\square K_{n-1})) = E_S(L(K_{n-1,n-1}\square K_n))$. □

Theorem 22 *If* $n \geq 3$ *then* $E_S(K_{n,n} \boxtimes K_{n-1}) = E_S(K_{n-1,n-1} \boxtimes K_n)$.

Proof. We have

$$\mathrm{Spec}(K_{n,n}) = \begin{pmatrix} n & 0 & -n \\ 1 & 2n-2 & 1 \end{pmatrix}$$

and

$$\mathrm{Spec}(K_{n-1}) = \begin{pmatrix} n-2 & -1 \\ 1 & n-2 \end{pmatrix}.$$

By using (iii) of Lemma 5 and Theorem 3,

$$\mathrm{Spec}_S(K_{n,n} \boxtimes K_{n-1}) = \begin{pmatrix} -2n+3 & 1 & 2n^2-4n+3 & -2n+3 \\ 1 & 2n^2-2n & 1 & 2n-2 \end{pmatrix}.$$

If $n \geq 3$ then $2n^2 - 4n + 3$ and 1 are only the positive S-eigenvalues of $K_{n,n} \boxtimes K_{n-1}$. Therefore from definition of S-energy

$$\begin{aligned} E_S(K_{n,n} \boxtimes K_{n-1}) &= 2[2n^2 - 2n + 2n^2 - 4n + 3] \\ &= 8n^2 - 16n + 6. \end{aligned}$$

Now

$$\mathrm{Spec}(K_{n-1,n-1}) = \begin{pmatrix} n-1 & 0 & -(n-1) \\ 1 & 2n-4 & 1 \end{pmatrix}$$

and

$$\mathrm{Spec}(K_n) = \begin{pmatrix} n-1 & -1 \\ 1 & n-1 \end{pmatrix}.$$

By using (iii) of Lemma 5 and Theorem 3,

$$\mathrm{Spec}_S(K_{n-1,n-1} \boxtimes K_n) = \begin{pmatrix} -2n+1 & 1 & 2n^2-4n+1 & -2n+1 \\ 1 & (n-1)(2n-2) & 1 & 2n-4 \end{pmatrix}.$$

If $n \geq 5$ then $2n^2 - 4n + 1$ and 1 are only the positive S-eigenvalues of $K_{n-1,n-1} \boxtimes K_n$. Therefore from definition of S-energy

$$\begin{aligned} E_S(K_{n-1,n-1} \boxtimes K_n) &= 2[(n-1)(2n-2) + 2n^2 - 4n + 1] \\ &= 8n^2 - 16n + 6 \end{aligned}$$

which completes the proof. □

4.2 Linear relations on S-energies of graphs

Linear relations on S-energies of different graphs are not yet well studied in the study of S-energies. In the following we present some linear relations on S-energies of different graphs of same order of the type $aE_S(G_1)+bE_S(G_2) = c$, where a, b and c are real numbers.

Theorem 23 *If* $n \geq 3$ *then* $E_S(K_{n-1,n-1} \otimes K_n) - E_S(K_{n,n} \otimes K_{n-1}) = 2$.

Proof. We have

$$\mathrm{Spec}(K_{n,n}) = \begin{pmatrix} n & 0 & -n \\ 1 & 2n-2 & 1 \end{pmatrix}$$

and

$$\mathrm{Spec}(K_{n-1}) = \begin{pmatrix} n-2 & -1 \\ 1 & n-2 \end{pmatrix}.$$

By using (ii) of Lemma 5 and Theorem 3,

$$\mathrm{Spec}_S(K_{n,n} \otimes K_{n-1}) = \begin{pmatrix} 2n-1 & -1-2n & -1 & 2n(n-2)-1 \\ n-1 & n-2 & (n-1)(2n-2) & 1 \end{pmatrix}.$$

If $n \geq 5$ then $2n-1$ and $2n(n-2)-1$ are only the positive S-eigenvalues of $K_{n,n} \otimes K_{n-1}$. Therefore from definition of S-energy

$$\begin{aligned} E_S(K_{n,n} \otimes K_{n-1}) &= 2[(2n-1)(n-1) + 2n(n-2) - 1] \\ &= 8n^2 - 14n. \end{aligned} \tag{9}$$

Now

$$\mathrm{Spec}(K_{n-1,n-1}) = \begin{pmatrix} n-1 & 0 & -(n-1) \\ 1 & 2n-4 & 1 \end{pmatrix}$$

and

$$\mathrm{Spec}(K_n) = \begin{pmatrix} n-1 & -1 \\ 1 & n-1 \end{pmatrix}.$$

By using (ii) of Lemma 5 and Theorem 3,

$$\mathrm{Spec}_S(K_{n-1,n-1} \otimes K_n) = \begin{pmatrix} 2n-3 & 1-2n & -1 & 2n^2-4n+1 \\ n & n-1 & n(2n-4) & 1 \end{pmatrix}.$$

If $n \geq 5$ then $2n-3$ and $2n^2-4n+1$ are only the positive S-eigenvalues of $K_{n-1,n-1} \otimes K_n$. Therefore from definition of S-energy

$$\begin{aligned} E_S(K_{n-1,n-1} \otimes K_n) &= 2[n(2n-3) + 2n^2 - 4n + 1] \\ &= 8n^2 - 14n + 2. \end{aligned} \tag{10}$$

From (9) and (10) the result follows. □

Conclusion

In this article we have obtained several classes of A-equienergetic and S-equienergetic graphs by using Cartesian product, tensor product and strong product. The results can be further extended to the other class of graphs. Also some linear relations between A-energies and S-energies of graphs has been established which shows a possible new direction in the study of relation between energies of different graphs of same order.

Acknowledgements

The authors would like to thank the anonymous referees for their helpful suggestions.

References

[1] C. Adiga, B. R. Rakshith, On spectra of variants of the corona of two graphs and some new equienergetic graphs, *Discuss. Math. Graph Theory* **36** (2016) 127–140. ⇒197

[2] R. Balakrishnan, The energy of a graph, *Linear Algebra Appl.* **387** (2004) 287–295. ⇒197

[3] A. S. Bonifácio, C. T. M. Vinagre, N. M. M. de Abreu, Constructing pairs of equienergetic and non-cospectral graphs, *Appl. Math. Lett.* **21** (2008) 338–341. ⇒197

[4] V. Brankov, D. Stevanović, I. Gutman, Equienergetic chemical trees, *J. Serb. Chem. Soc.* **69** (2004) 549–553. ⇒197

[5] A. E. Brouwer, W. H. Haemers, *Spectra of Graphs*, Springer, Berlin, 2012. ⇒198

[6] D. Cvetković, P. Rowlinson, S. Simić, *An Introduction to the Theory of Graph Spectra*, Cambridge University Press, Cambridge, 2010. ⇒197, 198

[7] H. A. Ganie, S. Pirzada, A. Iványi, Energy, Laplacian energy of double graphs and new families of equienergetic graphs, *Acta Univ. Sapientiae, Informatica.* **6** (2014) 89–116. ⇒197

[8] I. Gutman, The energy of a graph, *Ber. Math. Stat. Sekt. Forschungszentrum Graz.* **103** (1978) 1–22. ⇒197

[9] W. H. Haemers, Seidel switching and graph energy, *MATCH Commun. Math. Comput. Chem.* **68,** 3 (2012) 653-659. ⇒197

[10] F. Harary, *Graph Theory*, Narosa Publishing House, New Delhi, 1998. ⇒196

[11] G. Indulal, A.Vijayakumar, On a pair of equienergetic graphs, *MATCH Commun. Math. Comput. Chem.* **55,** 1 (2006) 83–90. ⇒197

[12] J. Liu, B. Liu, Note on a pair of equienergetic graphs, *MATCH Commun. Math. Comput. Chem.* **59,** 2 (2008) 275–278. ⇒197

[13] O. Miljković, B. Furtula, S. Radenković, I. Gutman, Equienergetic and almost equienergetic trees, *MATCH Commun. Math. Comput. Chem.* **61,** 2 (2009) 451–461. ⇒197

[14] H. S. Ramane, I. Gutman, H. B. Walikar, S. B. Halkarni, Equienergetic complement graphs, *Kragujevac J. Sci.* **27** (2005) 67–74. ⇒197

[15] H. S. Ramane, M. M. Gundloor, S. M. Hosamani, Seidel equienergetic graphs, *Bull. Math. Sci. Appl.* **16** (2016) 62–69. ⇒197

[16] H. S. Ramane, I. Gutman, M. M. Gundloor, Seidel energy of iterated line graphs of regular graphs, *Kragujevac J. Math.* **39,** 1 (2015) 7–12. ⇒198, 213

[17] H. S. Ramane, B. Parvathalu, D. D. Patil, K. Ashoka, Graphs equienergetic with their complements, *MATCH Commun. Math. Comput. Chem.* **82,** 2 (2019) 471–480. ⇒197, 198, 203

[18] H. S. Ramane, K. Ashoka, D. Patil, B. Parvathalu, I. Gutman, On complementary equienergetic strongly regular graphs, *Discrete Math. Lett.* **4** (2020) 50–55. ⇒197

[19] H. S. Ramane, D. Patil, K. Ashoka, B. Parvathalu, Equienergetic graphs using Cartesian product and generalized composition, *Sarajevo J. Math.* **17,** 30 No.1 (2021) 7–21. ⇒200

[20] H. S. Ramane, H. B. Walikar, Construction of equienergetic graphs, *MATCH Commun. Math. Comput. Chem.* **57,** 1 (2007) 203–210. ⇒197

[21] H. S. Ramane, H. B. Walikar, S. B. Rao, B. D. Acharya, P. R. Hampiholi, S. R. Jog, I. Gutman, Spectra and energies of iterated line graphs of regular graphs, *Appl. Math. Lett.* **18** (2005) 679–682. ⇒197

[22] I. Stanković, M. Milošević, D. Stevanović, Small and not so small equienergetic graphs, *MATCH Commun. Math. Comput. Chem.* **61,** 2 (2009) 443–450. ⇒197

[23] D. Stevanović, Energy and NEPS of graphs, *Linear Multilinear Algebra* **53** (2005) 67–74. ⇒197

[24] S. K. Vaidya, K. M. Popat, Some new results on Seidel equienergetic graphs, *Kyungpook Math. J.* **59,** 2 (2019) 335–340. ⇒197

Received: June 2. 2021 • Revised: August 18, 2021

Acta Univ. Sapientiae, Informatica **13,** 2 (2021) 220–250

DOI: 10.2478/ausi-2021-0010

Incremental methods for community detection in both fully and growing dynamic networks

Fariza BOUHATEM
Mouloud Mammeri University
of Tizi-Ouzou, Algeria
email: fariza.bouhatem@ummto.dz

Fatiha SOUAM
Mouloud Mammeri University
of Tizi-Ouzou, Algeria
email: fatiha.souam@ummto.dz

Ali AIT EL HADJ
Mouloud Mammeri University
of Tizi-Ouzou, Algeria
email: ali.aitelhadj.@ummto.dz

Abdelhakim DAFEUR
Mouloud Mammeri University
of Tizi-Ouzou, Algeria
email: abdelhakim.dafeur@ummto.dz

Abstract. In recent years, community detection in dynamic networks has received great interest. Due to its importance, many surveys have been suggested. In these surveys, the authors present and detail a number of methods that identify a community without taking into account the incremental methods which, in turn, also take an important place in dynamic community detection methods. In this survey, we provide a review of incremental approaches to community detection in both fully and growing dynamic networks. To do this, we have classified the methods according to the type of network. For each type of network, we describe three main approaches: the first one is based on modularity optimization; the second is based on density; finally, the third is based on label propagation. For each method, we list the studies available in the literature and state their drawbacks and advantages.

Computing Classification System 1998: G.2.2
Mathematics Subject Classification 2010: 68R15
Key words and phrases: community detection, incremental methods, fully dynamic networks, growing dynamic networks

1 Introduction

In real world that represents complex networks, nodes and edges change over time making the network dynamic. In reality there are two types of dynamic networks: fully dynamic networks [46, 70, 6] and growing dynamic networks [10, 65]. In the first case, like social networks, individuals (nodes) and their relationships (edges) can appear and/or disappear at any time. In the second type like citation networks, nodes represent articles and the links represent citations between the articles. Articles and citations can only be added to the network and they cannot be deleted later. In such type of network, the discovery of so-called community [27] is very hard problem and a number of methods have been proposed to solve it.

Among these methods proposed in the literature to explore community in dynamic networks we find [34, 11, 15] These methods apply static community detection algorithms multiple times to snapshots of the network. These methods are more expensive and it is more effective to incrementally review the community structure of the old network and update the community structure in a timely manner [40, 21, 36].

Several surveys have been proposed in the literature that focus on classifying existing methods designed for detecting community in dynamic networks ignoring incremental methods. Note that these methods , known as incremental methods, came to solve the problems of the traditional approaches while no researcher has classified them in any paper. In addition, in the existing surveys, the authors only deal with fully dynamic networks and do not take into account the growing dynamic networks in which incremental methods are best used for discovering communities.

In this paper, we present a survey of incremental methods for community detection in both fully and growing dynamic networks. This survey can be helpful for understanding of several incremental methods for choosing appropriate method. It is organized as follows. In Section 2, we present a history of community discovery through the listing and classification of methods used in the literature. In Section 3, we talk about dynamic networks and mention the events that can occur in community. We present in Section 4 our classification of various incremental approaches for community detection in both fully and growing dynamic networks as we present in Section 5 a discussion study that explores the advantages and the weaknesses of each approach. Finally, the conclusions section comes to wrap up the paper.

2 Historic for community detection

The graph partitioning methods [7, 38, 24] are originally the first solutions solving the problem of community detection. Graph partitioning relates to supervised methods[1] [31], it allows the classification of large sets and imposes the number of groups to be identified.

In real networks like social networks, the number of communities is not known in advance. In these networks, the number of groups is in itself an important result. A new problem thus posed is to decompose the network into a set of interconnected subgraphs, each constituting what is called a community[2] without knowing a priori the number of communities. Hence the birth of unsupervised methods like static community detection methods [60, 44, 2] and dynamic community detection methods [10, 34, 15].

It was after the publication of the work of Girvan-Newman [28] and Fortunato [27] that the problem of community detection gained interest. [44, 59] use spectral clustering which takes into account the spectral properties based on the eigenvalues of the Laplacian matrix of the data considered [56]. Another type of clustering has been used by the authors of [60] is hierarchical clustering. This approach are devided into two categories: the first category [9, 17, 43] groups the pairs of nodes until all nodes are in the same community [67]. Conversely compared to the first category, the second class [28, 49, 5] consists in dividing the graph into several communities [66] by iteratively eliminating the edges between the nodes until obtaining a singleton node [22, 1]. The authors of [20, 3] use density-based methods which locate regions of high density separated from each other by regions of low density. The DBSCAN[3] algorithm [20] uses parameters as input, the setting of these parameters is difficult to determine. In order to overcome this difficulty, the same authors [3] proposed the method called OPTICS[4] with a basic idea similar to DBSCAN. Another algorithm called DENCLUE[5] [33] measures the effect of each object in its neighborhood. The algorithm is faster than [20] and [3]. In addition to density-based methods, there are other, diffusion-based methods. These consist of the propagation of information to all nodes of the network. Nodes with the same information are grouped together in the same community. Raghavan [50]

[1] In supervised methods, the number and size of communities are known in advance.

[2] Community, cluster, group means the same.

[3] DBSCAN is the abbreviation of Density-Based Spatial Clustering of Applications with Noise.

[4] OPTICS is the abbreviation of Ordering Points to Identify the Clustering Structure.

[5] DENCLUE is the abbreviation of DENsity-based CLUstEring.

and others [61] have proposed the LPA[6] algorithm based on label propagation. As an extension of these algorithms [29] and [62] have presented respectively the COPRA[7] and SLPA[8] algorithms to detect overlapping communities.

In fact, real networks are dynamic, not static, as they evolve over time. This dynamic requires the use of effective methods to detect communities in the network. Among these ways we find, a two-stage approach [34, 11] is proposed to uncover dynamic networks. It independently detects the communities for each snapshot and then matches them. Since the two-stage approach is not entirely satisfactory, dependent community detection methods are introduced [15]. These methods used also snapshots when identifying groups, taking into account the communities found in the previous snapshot but avoiding the need to match them. Other methods designed for detecting community in dynamic networks work directly on temporal networks are the incremental approaches [10, 42, 57]. They start with an initial community, and then update for each incoming change the community structure. We are interested in our survey to dynamic community detection, precisely to incremental methods. Figure 1 summarizes the historic for community detection. The incremental approaches is our aims, so we have detailed it in Section 4.

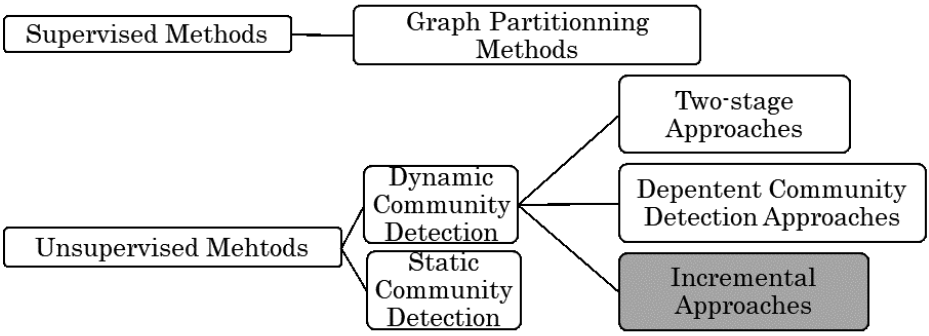


Figure 1: Organization diagram of the historic for community detection

[6]LPA is the abbreviation of Label Propagation Algorithm.
[7]COPRA is the abbreviation of Community Overlap PRopagation Algorithm.
[8]SLPA is the abbreviation of Speaker-listener Label Propagation Algorithm.

3 Dynamic networks

In social networks, the interaction between individual changes over time due to the change in subjects of interest and relationships. The dynamic nature of these networks makes difficult the detection of communities. These dynamic networks can be represented by:

- A time series of static networks, called snapshots (Figure 2) that poorly supports the dynamic aspect. Each snapshot corresponding to interactions derived from a daily, weekly, or monthly collection of data.
- Gathering information in real time as a stream of edges that integrates directly the evolutionary aspect of networks in a two-dimensional space. The nodes are classified on the axis of ordinates and their temporal links between two are represented by arcs over time (Figure 3).

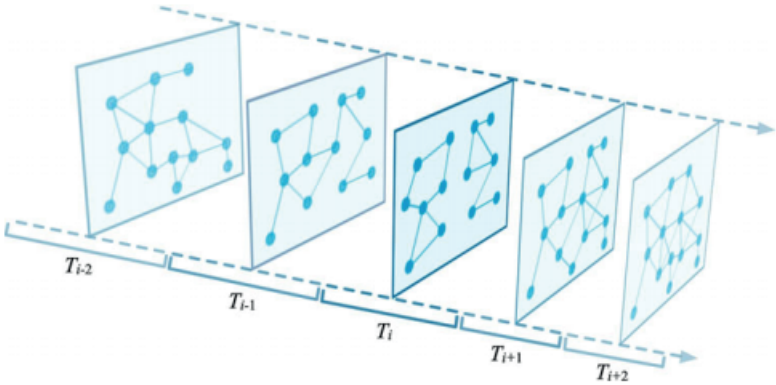


Figure 2: A temporal social network consisting of five timeframes [13]

3.1 Community events

With dynamic networks, nodes and edges can be added or removed at any time. These operations are called "events" and several of them are likely to appear in the life cycle of a community [55, 13, 48, 58]. The examples are shown in Figure 4 and a brief description can be found below:

- Continuing: the community continues its existence in the next time window.

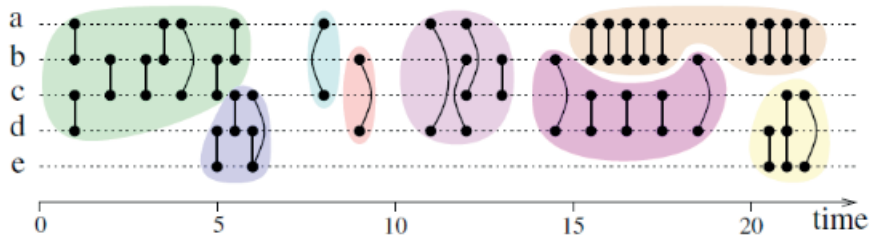


Figure 3: Diagram of a stream of edges with 5 vertices interacting with each other for 20 minutes (communities are colored differently) [32]

- Shrinking: the community shrinks or contracts when it loses some of its members.
- Growing: the community grows when some nodes (member) acquire.
- Splitting: the community is divided into several new communities.
- Merging: the community is divided into several new communities.
- Dissolving: when a community disappears (ends its life). All nodes belonging to this community disappear because they are dispersed between other communities.
- Forming: A community is formed or given birth when it appears at a given time.

3.2 Examination of dynamic community detection

Each algorithm or method designed for community detection must test its performance in a set of networks. These networks can be artificial networks or real networks. In this section, we will present most of the network models proposed in the literature to evaluate the result of the algorithm and the quality of the community structure.

In artificial networks or synthetic network generators, network data (benchmarks) are produced to model real-world networks and used to compare and assess the results of the algorithms. The best known of them are GN benchmark [28] and LFR benchmark [39] but there are others like ABCD [35] and [53] dedicated to community detection. These benchmarks are listed in Table 1. For each benchmark, we present its reference, its name, brief description and the type of generator.

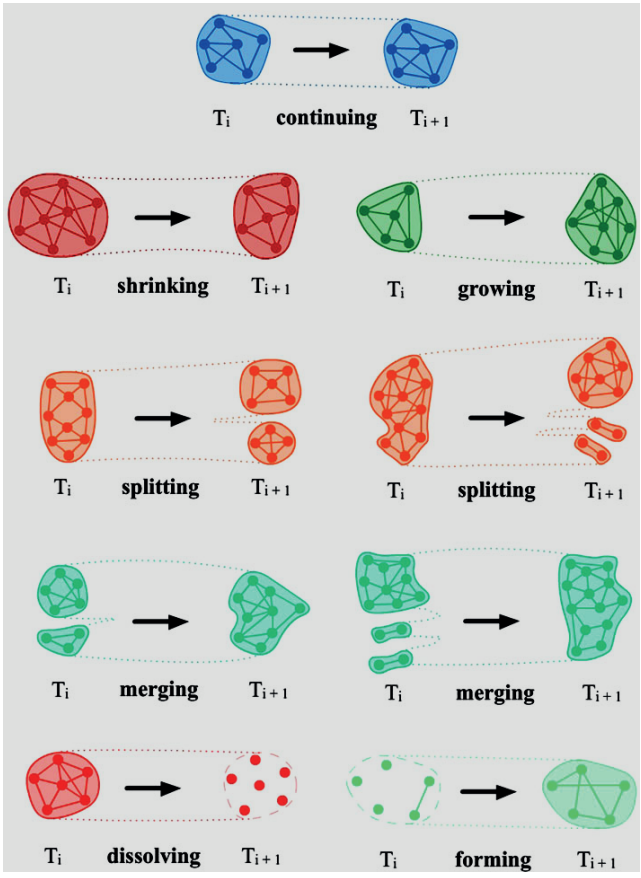


Figure 4: Seven possible type of events in the group evolution [13]

To evaluate the quality of a detected community, one method is to consider ground truth communities. In ground truth, community structure of the network may or may not be known in advance.

- If community structure of the network is known at advance, supervised measures are used. These measures compare the divergence between the community structure produced by community detection algorithms and the effective one. The most famous supervised measure used for the evaluation of quality of detected communities is NMI [39] which represent the degree of dependence between to partition. Equal to 1 if the two partitions are identical or to 0 otherwise. Rand proposed another metric

Reference	Name	Description	Type of generator
[5]	GN (Girvan-Newman benchmark)	- Based on the input of two values: mix parameter and average degree	Static (used also to evaluate dynamic community detection at each snapshot).
[62]	LFR (Lancichinetti Fortunato-Radicchi benchmark	- Community known in advance. - Depend on the mixing parameter.	Static (used also to evaluate dynamic community detection at each snapshot).
[63]	ABCD (Artificial Benchmark for Community Detection)	- Controls the fraction of edges between communities.	Static
[64]	A dynamic graph generator with overlapping community	- The node degrees, the community sizes, and the number of communities per node all follow power law distributions.	Dynamic graph generator with overlapping communities which simulate community scale events.

Table 1: Some benchmarks used in literature [28, 39, 35, 53]

similar to NMI called ARI [51] to measure the similarity between the detected community structure and a gold standard.

- If the ground-truth communities are unknown, unsupervised measures are often used. These measures estimate the quality of the partition without knowing the veritable partitions. Modularity proposed by Newman et al [43] is the most widely used metric quality (or quality function) to compare different community detection algorithm (cf. Section 4.2.1). However, it is not the only one: Kanna et al suggested Conductance metric [37] based on the density of community and the number of links outside. Radicchi et al defined the Internal Density measure [49], it is the ratio between the number of internal edges of the cluster and the number of its all possible internal edges.

All of these metrics used to evaluate the quality of a partition have some drawbacks. We cannot say that one metric is better than another, but certain selection criteria can be used for the choice of the metric. Some drawbacks and advantages of these measures are summarized in Table 2.

4 Incremental approaches for community detection in dynamic networks

Before describing the incremental approaches, we first list some notations and definitions used in this section.

Measure name	Measure type	Drawbacks	Advantages
NMI (Normalized Mutual Information)	Supervised	-You must know the basic community structure while it is unknown in many real dynamic networks. -High computational complexity -Impracticable for large-scale networks.	-Handle small networks.
ARI (Adjusted Rand Index)	Supervised	-Suffer from the problem of deciding if a pair of nodes should be linked or remain unlinked.	-Avoid the problem of unknown cluster. -ARI has the advantage of generating block model data comparing to other link prediction methods.
Modularity	Unsupervised	-Small communities may be not found.	-Handle large-scale network.
Conductance	Unsupervised	-Partition is good if it has both high values of intra and inter community conductance at the same time.	-Efficient in assessing community structure of disjoint communities.
Internal Density	Unsupervised	-Just considers internal relations between nodes of a community without any attention to external ones.	-Possibility to identify small and large communities.

Table 2: Some metric quality used to evaluate community detection algorithm (Drawbacks and advantages)

4.1 Notations and definitions

In this sub-section, we list out notations (Table 3) and introduce some definitions.

Notations	Description
C/C_S	Community / Communities
D	Disjoint communities
O	Overlap communities
In	Intrinsic communities
$V/v/V_{-}^{+}$	Number of vertices / Number of changed (new) vertices / Number of vertices added and deleted
E/ϵ	Number of edges / Number of changed edges
$\langle E \rangle / E_{-}^{+}$	Number of edges in each community / Number of edges added and deleted
I/i	Number of linear iterations in the number of edges E / Number of iterations
k_V	Number of degrees of the updated vertex
$d/d_t/d_v$	Average vertex degree / Average vertex degree at time slice t / the average degree of all vertices
$NbCinf$	Number of communities affected by the change in the network
S	Number of incremental sub-graphs
K	Number of communities
L	$L \ll V$ Maximum of vertices between the infected communities and incremental sub-graph

Table 3: Some notations used in this paper

Definition 1. Community
A group of nodes with similar characteristics or function that interact more densely among themselves.

Definition 2. Disjoint community
A subset of members that belong to one and only one community.

Definition 3. Overlap community
Communities that share common members (interests) across different communities.

Definition 4. Intrinsic (embedded) community
Sub-community in which members are strongly connected to each other forming a compact group within the parental community to which they belong.

4.2 Incremental approaches

The incremental approach considers the evolution of network as a series of modification in the network. The input in the network is a sequence of events such as addition and suppression of nodes and edges. The approach starts by finding communities for the initial state of the network. These communities are generally obtained by applying static methods [9, 17], then update, for each incoming change, the community structure by using community structure of the previous time step. As a consequence, the most topologies of network remained unchanged except for a small portion. Moreover, the incremental method often has a memory of what happened and take into account the history and the dynamic of network. The principle of incremental approaches is illustrated in Figure 5. Figure 5a shows the initial network at snapshot t and the sequence of modifications at snapshot t+1 and t+2. In snapshot t+1, a node and its links are added/deleted to / from the initial network. At snapshot t+2, two nodes and its links are added to the network of the snapshot t+1. In Figure 5b, the incremental approaches detect the first communities at t, two communities. After having detected communities at t, the incremental methods update the initial network according to the modification of network at t+1 (Figure 5c) and detect two communities. In Figure 5d, the same process in Figure 5c is applied to detect community at t+2 after updating communities of t+1 based on network changes at t+2.

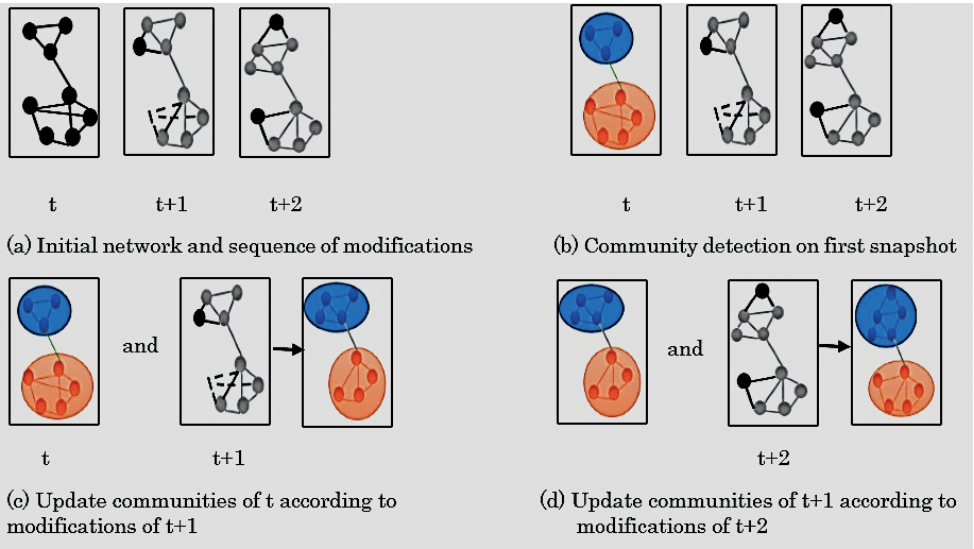


Figure 5: Principle of incremental approaches

Below is a classification of the existing methods for community detection according to the type of network and their based method. We present the set of existing methods by adopting the following hierarchical organization (Figure 6).

– Community detection in fully dynamic networks
- Modularity based optimization methods
- Density based methods
- Label propagation-based methods

– Community detection in growing dynamic networks
- Modularity based optimization methods
- Density based methods
- Label propagation-based methods
- Other methods

4.2.1 Incremental community detection methods in fully dynamic networks

In fully dynamic networks, communities can change or evolve over time by adding and removing nodes and edges. For example, in social networks, indi-

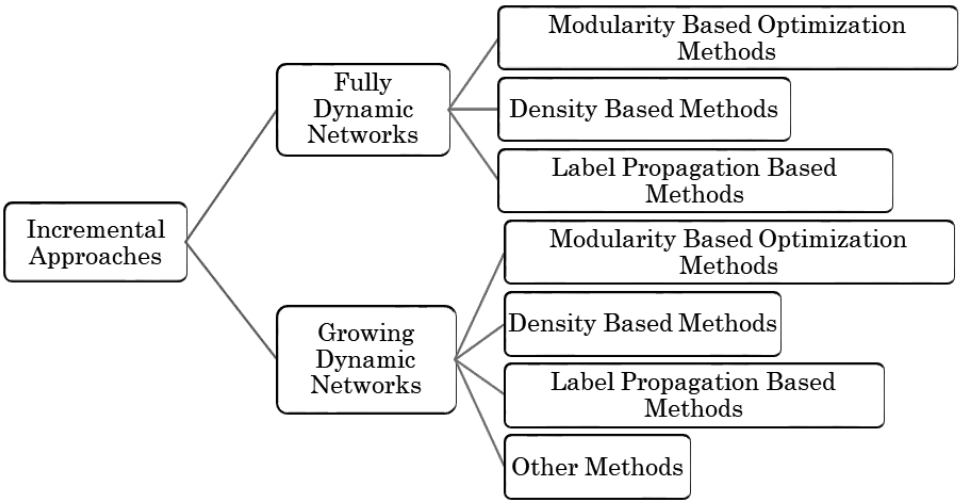


Figure 6: Classification chart of the existing incremental approaches for community detection

viduals (nodes) and their relationships (edges) can be added and/or removed at any time. From this explanation of fully dynamic networks, we present the existing incremental approaches for community detection on this field of network. For each category of these methods, we present a list of studies available in the literature. We provide tables (Table 4, 5, 6) in each table we present the following information: the algorithm name and its reference, the type of network in which the approach operates (weighted, unweighted, directed, undirected), the technique used by the algorithm to discover communities when changes occur in the network, the algorithmic complexity (if the algorithmic complexity is not available in the original paper we present it by ——— or by .....), the community type that the algorithm can detect (disjoint, overlap, intrinsic) and the algorithm used by the study during the initial phase to detect initial communities.

- *Modularity based optimization methods*

  The first approach is viewed as one major approach to incremental community detection. It consists on optimization of the metric modularity to detect communities on the network. This metric is proposed in [45] and used to measure the community structure in large scale networks. It is a difference between the fraction of edges inside the community and the fraction of edges expected by random version of the network.

In general methods, the process of maximization of modularity stars by assigning each node in network to different communities and then merge the nodes together until no gain of modularity is possible. Table 4 gives a summary of papers belonging to this category.

[6] introduced a Fast Community Detection for Dynamic Complex Network (FCDDCN) that is a real-time online community detection method in which network changes by the addition and the deletion of edges. The method is based on the modularity optimization using heuristic search and on the greedy agglomerative technique CNM [17]. It starts with an initial community structure obtained by applying CNM[9] algorithm and updates the network structure by using adjacency lists to add and delete elements to the lists.

[46] proposed a Quick Community Adaptation (QCA) for identifying and tracking community structure of dynamic network. The network changes are a collection of simple events such as newVertex, newEdge, removeVertex, and removeEdge. QCA requires an initial community structure that is obtained by performing a static community detection method "Louvain" [9] and the community assignments of neighbor nodes are adjusted by maximizing the modularity.

[18] introduced a new method which is a modification of the original Louvain method [9]. The proposed method keeps the community structure always updated after adding or removing edges and nodes by maximizing the local modularity gain function only for those communities that are affected by modifications to the network.

Contrary to the rule-based[10] methods [46] which considers each network change as an independent event, [16] proposed an Incremental Batch (InBatch) Technique for Community Detection. The method considers the network changes as a batch of network changes that appear in the same network snapshot. The changes that can undergo to the network are vertices/edges addition/deletion and communities can merge, split due to intra edge deletion/addition respectively. To update the community structure, this method initializes all of the new and changed nodes of the current snapshot as singleton communities, then applies Louvain technique until no further increase modularity is possible.

[9] CNM is the abbreviation of the names of the authors Clauset, Newman and Moore, who proposed this algorithm.

[10] Uses predefined rules to specify how to revise vertices' assignment.

Algorithm name and Reference	Network type	Technique used	Algorithmic complexity	Community type		Algorithm used during the initial stage
				D	O	
[41] FCDDCN	Undirected and unweighted	-List of modified edges over time steps.	$O(EdlogV)$	✔		CNM [3].
[35] QCA	Undirected and unweighted	-Considers each network change as an independent event.	$O(\epsilon)$	✔		Louvain [1].
[36] Dynamic Louvain	Undirected and unweighted	-Continuous arrival of information on singular events.	.....	✔		Louvain [1].
[39] IncBatch	Undirected and unweighted	-Considers a network changes as a batch appearing in the same network snapshot.	$O((v+\epsilon)*\frac{E}{V}+\lvert V\rvert^*)$ $\lvert V\rvert^*$: number of unique vertices after initialization phase	✔		InBatch initialization phase.
[40] LBTR	Undirected and unweighted	-Predict vertices new community assignments after each round of network change.	$O(v*\frac{drR}{(1-drR)P})$ R and P: the recall and precision of the classifier r: the probability that an examined vertex actually needs community assignment revision.	✔		Louvain [1] or any static community detection algorithm.
[38] DynaMo	Undirected and weighted	-Processing a set of network changes as a batch.	$O(\epsilon+\lvert E\rvert^*)$ $\lvert E\rvert^*$: number of unique edges after initialization phase	✔		Louvain [1].

Table 4: Incremental community detection methods in fully dynamic networks based on modularity optimization [46, 18, 70, 16, 55, 6]

[55] proposed a Learning-Based Target Revision (LBTR) approach that uses machine learning classifiers and historical community structure information to identify vertices whose community assignment needs to be revised and filters unchanged vertices. To build the vertex classifier LBTR uses Logistic Regression (LR) and Support Vector Machine (SVM), namely LBTR-LR and LBTR-SVM. Similar to previous study [16], this method put the new vertices into singleton communities and move the community assignment of changed vertices to maximize the modularity gain.

The authors of [70] proposed DynaMo[11], a new method designed for maximizing the modularity gain while updating the community structures. The dynamic network is modelled as a sequence of incremental changes: intra-community edge addition/weight increase, cross-community edge addition/weight increase, intra-community edge deletion/weight decrease, cross-community edge deletion/weight decrease, vertex addition, and vertex deletion. For each incremental network change, DynaMo maximizes the modularity in two steps. The first one initializes an intermedi-

[11]DynaMo: Dynamic Community Detection by Incrementally Maximizing Modularity.

ate community structure, depending on the incremental network changes and on the previous network community structure. In the second step, the last two phases of Louvain algorithm [9] are repeated on the intermediate community structure until no gain of modularity is possible.

- *Density based methods*

  The basic idea of density-based approach is to form a group that is dense enough and separated by sparse or low-density region. In other words, the connection of nodes inside a group is greater than the connection of nodes outside the group. Table 5 gives a summary of studies in the literature for community detection based on density.

  In [23] the graph mining algorithm DenGraph[12] is proposed which is inspired by Incremental DBSCAN [21] the incremental version of DBSCAN [20]. The author's transfer the basic concepts of these algorithms to graph mining by defining proximity for graph nodes. In others words, they transfer the idea of density based incremental clustering of spatial data to social network structures. The algorithm adjusted locally the vertices (Border vertices, noise vertices) whose distances to the core vertices were changed when the network updates. The same authors proposed DenGraph-HO[13][52] based on the concepts of the DenGraph algorithm to detect overlapping communities.

  [47] used a density [25] as an objective function to Adaptively Finding Overlapping Community Structures (AFOCS) and tracing their evolution over time. The framework identifies the basic overlapped community structure in a network as a collection of dense parts of the network using FOCS[14] and then combines overlapping communities if they share significant substructure and if they are highly overlapped. The changes that can occur to the network are summarized into four events: adding and deleting nodes, adding and deleting edges.

  Lastly, [40] proposed iDBLINK an incremental Density Based LINK clustering algorithm which is an extended version of the static algorithm DBLINK[15] [41]. The algorithm can effectively update the current community structure according to the change of the community structure and network topology a moment before. It focuses on the change of link

[12]DenGraph: A Density based Community Detection Algorithm.
[13]DenGraph-HO: A Density-based Hierarchical Graph Clustering Algorithm.
[14]FOCS is the abbreviation of Finding Overlapping Community Structures.
[15]DBLINK is the abbreviation of Density Based LINK clustering algorithm.

communities through the change of similarity between the edges at the adjacent moments. The changes of link communities are divided into positive changes and negative changes. The network topology updates with positive changes mean that communities grow, merge or create and negative changes mean that communities delete, split or decline. The overlapping community detection quality was improved despite the fact that this algorithm kept the advantage of DBLINK algorithm.

Algorithm name and Reference	Network Type	Technique used	Algorithmic complexity	Community type		Algorithm used during the initial stage
				D	O	
[19] DenGraph	Undirected and Weighted	-Continuous arrival of information on singular events.	O(E+V)	✓		IDBSCAN [46].
[44] AFOCS	Undirected and Unweighted	-Considers each network change as an independent event.	$O(E^2)$		✓	FOCS [45].
[31] iDBLINK	Undirected and Unweighted	-A collection of simpler events at each time step.	-------	✓	✓	DBLINK [30.]

Table 5: Incremental community detection methods in fully dynamic networks based on density [52, 40, 47]

- *Label/information propagation-based methods*

  In label propagation methods, each node has its own label which changes by interaction with its neighborhood. The nodes can share the same label to identify a disjoint group or allowing multiple labels to detect overlapping community. Instead of propagating of labels, the information is propagated and exchanged between nodes in the network. The propagation of information between nodes with similar interests tends to be more frequent, as a result, node in the same community have almost the same quantity of information, whereas those in various communities have different quantity of information. Table 6 gives the studies that have exploited these ideas.

  In their work, [63] proposed LabelRankT[16] to detect evolving communities in large scale dynamic network. The algorithm is based on the generalized LabelRank[17] [64] to incorporate important network features such as edge weights and directions. Since each node requires only local information during label propagation process. During the evolution,

[16]LabelRankT: Incremental Community Detection in Dynamic Networks via Label Propagation.

[17]LabelRank: A Stabilized Label Propagation Algorithm for Community Detection in Networks.

nodes and edges are added or removed and only the nodes that are changed between two consecutive snapshots are updated in the algorithm. To give an example, when a new edge is added in the stream, the algorithm updates only the nodes that are attached to this edge. Indeed, the communities can split, merge and dissolve.

The ALPA[18] algorithm was introduced by [30] to detect communities through a local label propagation process. The algorithm consists in adding (or removing) edges/nodes as it appears (or disappears). The edge can be intra-community edge or inter-community edge and the node as an isolated or with its adjacent edges. To update the community structure and avoid unnecessary updates ALPA apply local label propagation process [50]. ALPA perform the warm-up step to propagate labels only inside the target communities, then apply local label propagation process to involve some nodes outside of the target communities.

Other work belonging to this category is [4] that proposed an unsupervised machine learning algorithm based on SLPA [62]. The algorithm used an extended SLPA to detect communities at the first time slot. The first list of labels is obtained and is used to assign the labels to unlabeled nodes at the next time slot. To execute the algorithm, three input arguments are necessary: an input graph, a stop criterion and a threshold.

Lastly, [57] proposed a new framework to detect communities in dynamic networks based on information dynamics. The framework uses information dynamic model to identify initial communities and to incrementally discover community structure of the network. The nodes and edges are added and deleted to the network that leads to the merging, division, expansion, contraction, birth and death of communities.

[18] ALPA is the abbreviation of Adaptive Label Propagation Algorithm.

Algorithm name and Reference	Network type	Technique used	Algorithmic complexity	Community type		Algorithm used during the initial stage
				D	O	
[32] LabelRankT	Directed and weighted undirected and unweighted	-Considers a network changes as a batch appearing in the same network snapshot.	O(IE)	✓		LabelRank [33]
[29] ALPA	Undirected and unweighted	-Considers each network change as an independent event.	O(t*i < E >) t: time step	✓		Any available static methods or LPA [25].
[34] ISLPA	Directed and undirected	-A network changes is a batch of nodes and edges that are removed or added over time.	O(TE) T: a stop criterion	✓	✓	ESLPA [34].
[54] DCDID	Undirected	-Batch processing technique.	(1) In initial community partition: O(i*V*d) (2) In incremental community detection $O(E^+_- + i * V^+_- * d_t)$	✓		CDID [54].

Table 6: Incremental community detection methods in fully dynamic networks based on label/information propagation [30, 63, 4, 57]

Some abbreviations in Table 6 described in footnote: ISLPA[19], ESLPA[20], DCDID[21], CDID[22].

4.2.2 Incremental community detection methods in growing dynamic networks

We agree that all networks change over time are dynamic networks. In this type of network, there are networks that evolve only by adding nodes and their links and cannot be removed later. For example, in research article citations networks, most articles cite the previous work by other authors on the same topic. These citations form a network in which nodes represent articles and edges represent the oriented link from article A to article B which indicates that B is cited by A. These networks are a special case of dynamic networks and for this reason we have adopted the name "growing dynamic networks". Tables 7, 8, 9, and 10 list the studies belonging to this category, we present in each table : the algorithm name and its reference, the type of network in which the approach works (weighted, unweighted, directed, undirected), changes that refer to

[19]ISLPA is the abbreviation of Incremental Speaker-Listener Propagation Algorithm.

[20]ESLPA is the abbreviation of Extended Speaker-Listener Label Propagation Algorithm.

[21]DCDID is the abbreviation of Dynamic Community Detection based on Information Dynamics

[22]CDID is the abbreviation of Community Detection based on Information Dynamics.

the review strategies that an algorithm applies when updating networks (community events), the technique used by the algorithm to discover communities when changes occur in the network, the algorithmic complexity, if it is available in the original paper, the community type that the algorithm can detect (disjoint, overlap, intrinsic) and the algorithm used by the study during the initial phase to detect initial communities.

- *Modularity based optimization methods*

  Authors of [54] proposed GreMod[23], a real time detecting algorithm to track community structure of growing dynamic networks. Their method consists in adding of new edges between: (1) two nodes already exist and belong to the same community, (2) two nodes already exist and belong to different communities, (3) new node and old node, (4) two new nodes. It starts with initial communities calculated by Louvain [9] algorithm, then apply their incremental strategies in a way to increase the modularity if possible.

  In [65], Authors introduced NGI[24], a novel approach for node-grained streaming network in which changes arrive sequentially and frequently. The method adds a single node with its connecting edges in the network simultaneously and all the edges at the same time. The node added to the network can come: (1) without any neighbor, (2) with multiple neighbors belonging to an isolating singleton community, (3) with multiple neighbors belonging to different communities.

  [68] proposed Δ–screening technique to quickly identify the relevant parts of the graph that are potentially impacted by a batch of changes ((1) adding edges). The approach assigns firstly all vertices to a distinct community using any static community detection algorithm [9, 17]. After an iterative process, the method identifies all vertices whose community affiliation could potentially change and retrain the previous community assignments to the remaining vertices. The vertex that changes his affiliation migrates to a neighboring community that maximizes the modularity gain of that vertex. Table 7 lists the works talk over in this section.

[23]GreMod: A Real-Time Detecting Algorithm for Tracking Community Structure of Dynamic Networks.

[24]NGI is the abbreviation of Node-Grained Incremental community detection algorithm.

Algorithm name and Reference	Network type	Changes	Technique used	Algorithmic complexity	Community type		Algorithm used during the initial stage
					D	O	
[47] GreMod	Undirected and Weighted	-(1)-> No change. -(2)->No change/merge C_S. -(3)->assigning node to an existing C / create new C. -(4)-> assigning node to an existing C / create new C.	-The edges added to the network are processed one by one.	$O(\epsilon)$	✓		Louvain [1].
[57] NGI	Undirected and unweighted	-(1)- > New isolated C. -(2) ->Aggregate all nodes in the adjacent C_s with newly arrived node into a community. -(3) -> Insert a new node in one of its adjacent C_s or merge similar C_s with new coming node.	-Node with its connecting edges is added into network simultaneously and all edges arrive at the same time.	$O(E * d_v)$	✓		Detect the initial communities after the arriving of the first node in the network.
[43] Δ-screening	Undirected and weighted or unweighted	(1) (not specified the type of edges)->reevaluate all vertices whose community affiliation could potentially change.	-Considers the addition of edges as a batch appearing in the same network snapshot.	_____	✓		Any static community detection algorithm.

Table 7: Incremental community detection methods in growing dynamic networks based on modularity optimization [68, 54, 65]

- *Density based methods*

  To detect and track the evolution of hierarchical and overlapping communities in evolving networks [8] proposed a novel approach called HOC-Tracker. The approach identifies a cluster by detecting the neighborhood of each node in the network using a distance function. Then for each new event, it classifies the active nodes that have caused the network to change in order to track evolutionary events like birth, death, merge, split, growth and shrink of communities.

  [42] proposed InDEN (Intrinsic Community Detection in Evolving Networks) method to find intrinsic community in growing dynamic networks. The method starts with the initialization of community containing the first two nodes that are incoming with the first edge. Then, InDEN uses the membership score to assign any new incoming node into the community with maximum score. To detect intrinsic communities, the approach analyses the density variation in community at each time.

  In their work, [10] proposed a density-based approach with dual optimization to track and identify the community structure. The proposed method starts from an initial community obtained by Louvain [9], then for each event that occurs in the network, a new node and their links are added in the best community after having passed by two levels of

density optimization. This optimization is calculated only in the community affected by the changes. Table 8 lists the studies discussed in this section.

Algorithm name and Reference	Network Type	Changes	Technique used	Algorithmic complexity	Community type			Algorithm used during the initial stage
					D	O	In	
[53] HOCTracker	Directed/ undirected and Weighted/ unweighted	-New core node emerges ->expansion/birth/merge of C_S. -A core node becomes non-core->split/shrink of C_S. - A core node gains nodes/ loses nodes but remains a core->merge, growth, split, shrink of C_S.	-Considers the active node and their direct neighbors at each new state of the network.	O(E)		✓		Initial community obtained by HOCTracker [53].
[48] InDEN	Undirected and unweighted	-New edges between two existing nodes or a new edge between one existing node and new node ->movement of node from one C to another/ merge of C_S / create new C.	-Adds edges to the community independently.	_____	✓		✓	_____
[37]	Undirected and unweighted	-New isolated node -> create new C. -New node with its links in the same C -> reinforce the infected C. -New node with its links in different C_S-> split/merge /birth of infected C_S.	-For each time stamp adding node and its links simultaneously.	O(nbCinf)	✓			Louvain [1] or Any static community detection algorithm [3].

Table 8: Incremental community detection methods in growing dynamic networks based on density [10, 42, 8]

Some abbreviations in Table 8 described in footnote: HOCTracker[25], core node[26]

- *Label propagation-based methods*

  Hiroki et al. [36] proposed a new algorithm to detect communities for high volume graphs stream based on DEMON[27] [19] algorithm. It used three functions for this issue, two incremental functions ((1) Ego minus ego network, (2) Label propagation) and (3) optimized merge function. Table 9 summarizes the papers belonging to this class.

[25]HOCTracker: Tracking the Evolution of Hierarchical and Overlapping Communities in Dynamic Social Networks.

[26]Core node is a node having non-zero reciprocated interactions with any of its neighbor(s) in a set of nodes of the network.

[27]DEMON: a local-first discovery method for overlapping communities.

Algorithm name and Reference	Network Type	Changes	Technique used	Algorithmic complexity	Community type			Algorithm used during the initial stage
					D	O	In	
[51] Incremental DEMON	Undirected and unweighted	-Adding nodes and his edges/Adding edges. between existing nodes-> merge of C_S.	-Batch technique.	(1): O(V+E)*d (2): O(V+d^2) (3): $\|K_V\|$*V	✓	✓		DEMON [52]

Table 9: Incremental community detection methods in growing dynamic networks based on label propagation [36]

- *Other methods*

Algorithm name and Reference	Network type	Changes	Technique used	Algorithmic complexity	Community type			Algorithm used during the initial stage
					D	O	In	
iLCD [50]	Undirected and unweighted	-(1) and (2)->updating of existing C_S /create new C /merge similar C_S.	-For each time stamped adding a set of edges simultaneously.	It's nearly impossible to determine the complexity of iLCD.		✓	✓	_____
[49]	Undirected and unweighted	-(1) ->birth of new C. -(2) ->keep or not the edges in their C. -(3) ->birth of new C/ enlarging of the previous C_S. -(4) ->birth of new C/strengthening of previous C_S.	-Joining simultaneously incremental subgraph in the network.	O(V + S + K + L)	✓			_____

Table 10: Some other incremental community detection methods in growing dynamic networks [69, 14]

A part from that, [14] proposed iLCD (intrinsic Longitudinal Community Detection) algorithm to find intrinsic and overlapping communities. The events that can occur to the network are: (1) adding edges between existing nodes, (2) adding edges between new nodes and existing nodes. The community are detected by doing re-evaluation at each new iteration according to the path lengths between each node and its adjacent communities.

[69] proposed a new incremental method capable of handling subgraphs (including nodes and edges) addition. To do this, the method proposes four types of incremental elements: (1) complete independent, (2) complete contained, (3) mixed with new and old nodes and (4) multiple contained. It then applies its own update strategies based on the edge weights to determine the impact of historical relationship. Finally, it outputs the latest communities and updates

the evolutionary path accordingly. Table 10 reviews some other methods for community detection in growing dynamic networks.

5 Discussion study

We discuss in this section the weaknesses and strengths of each incremental approach based on modularity optimization, density and label propagation. An overview is summarized in Table 11. We present then in general terms the main advantages and disadvantages of using of incremental approaches for community detection in fully dynamic networks and in growing dynamic networks. The advantages and disadvantages are listed in Table 12.

5.1 Modularity based optimization methods

Most of these approaches aim to maximize modularity in order to assess the quality and robustness of their detected communities. This scale is one of the most important metrics because it has the unique advantage of being a universal standard at the same time for defining communities and it is a key component of the most common method of graph clustering. These methods are suitable for temporal networks whose stream changes or evolves over time and for a stochastic network that not expected to have a cluster structure. Despite these strengths, some limitations have been noted about its performance. First, small communities may not be found. This limitation is called resolution limit problem [12, 26] and is considered one of the most serious problems in detecting communities. Another problem known as degeneracy problem, when modularity maximization finds so many different partitions whose typical values are very close to each other. This problem is most grave when applied to networks with modular structure; it happens for weighted, directed, bipartite and multi-scale generalizations of modularity. Finally, maximizing modularity is NP-complete problem, so the solution cannot be found in polynomial-time.

5.2 Density based methods

Most of the algorithms belonging in this class are able to identify groups that are more connected inside and less connected outside. Its aim is to discover small and large clusters thus solving the resolution limit problem resulting from modularity optimization. Other strengths of these methods: is that the number of clusters is not required as an input to the algorithm but rather is disclosed in the algorithm based on data set characteristics, insensitive to

noise[28] and outliers[29] and capable to identify clusters of arbitrary shapes[30]. Even though these advantages, these approaches still have weaknesses it is unsuitable for high-dimensional datasets because of the inherent shortage of the feature space, which in turn, reduces any clustering propensity.

5.3 Label propagation-based methods

Label propagation approaches are considered as one of the fastest methods of community detection with a near linear time complexity. At this speed, these methods are suitable for large-scale networks. One of the main weaknesses of these approaches is that in the general case the algorithm is non-deterministic, the multiplicity execution of the same algorithm produces no unique solution, but an aggregate of many solutions. This problem remains important, and all label propagation-based methods attempt to overcome it. Another potential issue is the uncertainty and randomness in the label propagation process. This problem may affect the stability and accuracy of detected communities. Also, the bad propagation of labels can lead to the discovery of huge communities without sense, this problem is known as "Giant community's problem".

Method based on	Weaknesses	Strengths
Modularity optimization [35, 36, 38, 39, 40, 41, 43, 47, 57]	-Suffers from the resolution limit problem and degeneracy problem. -Less efficient in networks with a modular or hierarchical structure. -Maximizing modularity to detect communities is an NP -complete problem.	-The most common quality measure to evaluate community detection algorithm. -One of the most important criteria for community detection. -Communities are detected in real - world networks without knowing the community structure in advance.
Density [19, 31, 37, 44, 48, 53]	-Unsuitable for high -dimensional datasets	-Able to identify small and large groups. -Number of clusters is not predefined. -Clusters of arbitrary shape can be detected. -Outliers and noise do not a ffect the result of the algorithm.
Label propagation [29, 32, 34, 51, 54]	-Non-deterministic algorithm. - Giant community's problem. -Instability and inaccuracy of community detection algorithm.	-The fastest way to discover communities. -Low time complexity.

Table 11: Some weaknesses and strengths of incremental methods based on modularity optimization, density, and label propagation [52, 30, 40, 63, 4, 46, 18, 10, 70, 16, 55, 6, 68, 47, 54, 42, 36, 8, 57, 65]

[28] Node with area of low density.
[29] Nodes which cannot be grouped into any of the communities.
[30] Non-convex shape.

5.4 Advantages and disadvantages of incremental methods for community detection in both fully and growing dynamic networks

We return to the first tree level of our classification (Figure 6) to discuss the main advantages and disadvantages of incremental community detection methods in fully and growing dynamic networks (Table 12). In general, the incremental approach has the advantage in runtime and computational complexity because it updates the current community structure according to the change of the network and the previous community structure, thus it can also avoid the hassles of re-detection. There are other advantages of ensuring stable communities across time steps which allows avoiding the instability problem taken from methods that detect communities at each time steps undependably. As a result, it deals with temporal networks[31] that change frequently and works well in growing dynamic networks. Incremental clustering has been used as one of the most efficient methods; however, it requires initial community structure of the network, but this is unknown in many real networks. Another defect is the accumulation of errors resulting from the partition error which will lead to a discrepancy between the computed community structure and the underlying ground-truth. As a result, it is difficult to ensure the cohesion of communities in the set of steps in evolution because tracking a community is only done when switching from one snapshot to another. Another problem is the possibility of neglecting of some changes in the associated community structure due to the detection of only local structure of the network.

Advantages	Disadvantages
-It has low runtime and low complexity to detect community. -Ensure the stability of communities because unaffected communities keep unchanged, that simplify the p rocess of tracking of communities over time. -Avoid re-detection at each snapshot. -Effective compared to other methods that detect communities independently. -More valuable when the networks are in mega -scale or change frequently (streaming network). -Suitable to detect community of temporal dynamic networks but works well in growing dynamic networks.	-Requires to have the initial community structure of the network, but this is unknown in many real networks. -It is difficult to ensure the cohesion of commu nities in the set of steps in evolution. -Inconsistency between the computed community structure and the underlying ground-truth. -The processing order of nodes / edges deletion/ addition in fully dynamic networks may have an impact on the detection results and efficiency.

Table 12: Major disadvantages and advantages of incremental approaches in fully and growing dynamic networks

[31] A temporal network also known as time-varying network is a dynamic network in which both nodes and edges may appear and disappear as time goes.

Abbreviation in Table 12 described in footnote: Unaffected communities (column one)[32].

6 Conclusions

An incremental approach is one way to discover communities in both fully and growing dynamic networks. The idea of these approaches is to build and maintain communities in a network, following a series of changes that occur to the network itself.

In this paper, a study of incremental methods for detecting communities in both fully and growing dynamic networks is presented and discussed. The aim of this survey was to classify the incremental methods and discuss their disadvantages and advantages. Both points will help the reader to define and find direction for his future research and choose the appropriate method depending on the type of networks, type of communities and the technique used to identify community.

Detecting communities is NP-hard problem. Therefore, defining the community accurately and effectively in dynamic networks remains a very interesting and challenging task and incremental approaches try to solve it.

References

[1] Y. Y. Ahn, J. P. Bagrow, S. Lehmann, Link communities reveal multi-scale complexity in networks, *Nature*, **466**, 7307 (2010) 761–764. ⇒222

[2] E. Akachar, B. Ouhbi, B. Frikh, A new algorithm for detecting communities in social networks based on content and structure information, *International Journal of Web Information Systems*, 2019. ⇒222

[3] M. Ankerst, M. M. Breunig, H. P. Kriegel, J. Sander, OPTICS: Ordering Points To Identify the Clustering Structure, *ACM Sigmod record*, **28**, 2 (1999) 49–60. ⇒222

[4] M. Asadi, F. Ghaderi, Incremental community detection in social networks using label propagation method, *23rd Conference of Open Innovations Association (FRUCT)*, 2018, pp. 39–47. ⇒236, 237, 243

[5] J. Bagrow, E. Bollt, A local method for detecting communities, *Physical review E*, **72**, 4 (2005) 46–108. ⇒222

[6] S. Bansal, S. Bhowmick, P. Paymal, Fast community detection for dynamic complex networks, *Complex Networks, Springer*, Berlin, Heidelberg, 2011, pp. 196–207. ⇒221, 232, 233, 243

[32] Unaffected communities are communities those unconcerned by the change that occur to the network.

[7] E. R. Barnes, An Algorithm for Partitioning the Nodes of a Graph, *SIAM Journal on Algebraic and Discrete Methods*,**3**, 4(1982) 541–550. ⇒222

[8] S. Y. Bhat, M. Abulaish, HOCTracker: Tracking the evolution of hierarchical and overlapping communities in dynamic social networks, *IEEE Transactions on Knowledge and Data engineering*, **27**, 4(2014) 1019–1013. ⇒239, 240, 243

[9] V. D. Blondel, J. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, *Journal of Statistical Mechanics: Theory and Experiment*, **2008**, 10 (2008) P10008. ⇒222, 229, 232, 234, 238, 239

[10] F. Bouhatem, A. A. Ait El Hadj, F. Souam, Density-based Approach with Dual Optimization for Tracking Community Structure of Increasing Social Networks, *International Journal on Artificial Intelligence Tools*, **29**, 01(2020) 2050002. ⇒ 221, 222, 223, 239, 240, 243

[11] L. Boujlaleb, A. Idarrou, D. Mammass, Tracking community evolution in social networks, *Journal of theoretical and applied information technology*, **95**, 22 (2017) . ⇒221, 223

[12] U. Brandes, D. Delling, M. Gaertler, R. Gorke, M. Hoefer, Z. Nikoloski, D. Wagner, On modularity clustering, *IEEE transactions on knowledge and data engineering*, **20**, 2 (2007) 172–188. ⇒242

[13] P. Bródka, S. Saganowski, P. Kazienko, GED: the method for group evolution discovery in social networks, *Social Network Analysis and Mining*, **3**, 1(2013), 1–14, ⇒224, 226

[14] R. Cazabet, F. Amblard, C. Hanachi, Detection of overlapping communities in dynamical social networks, *IEEE second international conference on social computing*, 2010, pp. 309–314. ⇒241

[15] D. Chakrabarti, R. Kumar, A. Tomkins, Evolutionary clustering, *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006, pp.554–560, ⇒221, 222, 223

[16] W. H. Chong, L. N. Teow, An incremental batch technique for community detection, *Proceedings of the 16th international conference on information fusion*, 2013, pp. 750–757. ⇒232, 233, 243

[17] A. Clauset, M. E. J. Newman, C. Moore, Finding community structure in very large networks, *Physical review E*, **70**, 6 (2004) 066111. ⇒222, 229, 232, 238

[18] M. Cordeiro, R. P. Sarmento, J. Gama, Dynamic community detection in evolving networks using locality modularity optimization, *Social Network Analysis and Mining*, **6**, 1(2016) 15. ⇒232, 233, 243

[19] M. Coscia, G. Rossetti, F. Giannotti, D. Pedreschi, Demon: a local-first discovery method for overlapping communities, *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012, pp. 615–623. ⇒240

[20] M. Ester, H. P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, Proceedings of the 2nd International Conference on Knowledge Discovery and Data mining, 1996, pp. 226–231. ⇒222, 234

[21] M. Ester, R. Wittmann, Incremental generalization for mining in a data warehousing environment, *International Conference on Extending Database Technology, Springer*, Berlin, Heidelberg,1998, pp. 135–149. ⇒221, 234

[22] T. S. Evans, R. Lambiotte, Line graphs, link partitions and overlapping communities, *Physical review E*, **80**, 1 (2009) 016105. ⇒222

[23] T. Falkowski, A. Barth, M. Spiliopoulou, Dengraph: A density-based community detection algorithm, *IEEE/WIC/ACM International Conference on Web Intelligence (WI'07)*, 2007, pp. 112–115. ⇒234

[24] C. M. Fiduccia, R. M. Mattheyses, A Linear-Time Heuristic for Improving Network Partitions, *19th Design Automation Conference, IEEE*, 1982, pp. 175–181. ⇒222

[25] S. Fortunato, C. Castellano, Community structure in graphs, ArXiv preprint arXiv: 0712.2716, 2007. ⇒234

[26] S. Fortunato, M. Barthelemy, Resolution limit in community detection, *Proceedings of the national academy of sciences*, **104**, 1(2007) 36–41. ⇒242

[27] S. Fortunato, Community detection in graphs, *Physics Reports*, **486**, 3(2010) 75–174. ⇒221, 222

[28] M. Girvan, M. E. J. Newman, Community structure in social and biological networks, *Proceedings Of the National Academy Of Sciences* , **99**,12 (2002) 7821–7826. ⇒222, 225, 227

[29] S. Gregory, Finding overlapping communities in networks by label propagation, New Journal of Physics, **12**, 10 (2010) 103018. ⇒223

[30] J. Han, W. Li, L. Zhao, Z. Su, Y. Zou, W. Deng, Community detection in dynamic networks via adaptive label propagation, *PloS one*, **12**, 11 (2017) e0188655. ⇒236, 237, 243

[31] J. A. Hartigan, M. A. Wong, Algorithm AS 136: A K-Means Clustering Algorithm, *Journal of the Royal Statistical Society, Series C*, **28**, 1(1979) 100–108. ⇒222

[32] S. Heymann, *Exploratory link stream analysis for event detection*, Doctoral dissertation, Université Pierre et Marie Curie-Paris VI, Paris, 2013. ⇒225

[33] A. Hinneburg, D. A. Keim, An Efficient Approach to Clustering in Large Multimedia Databases with Noise, *Knowledge Discovery and Data Mining*, New York City, USA, 1998, pp. 58–65. ⇒222

[34] J. Hopcroft, O. Khan, B. Kulis, B. Selman,Tracking evolving communities in large linked networks, *Proceedings of the national academy of sciences of the United States of America*, 101 (2004) 5249–5253. ⇒221, 222, 223

[35] B. Kaminski, T. Olczak, Artificial Benchmark for Community Detection (ABCD), *Joint Mathematics Meetings (JMM), AMS*, 2021. ⇒225, 227

[36] H. Kanezashi, T. Suzumura, An incremental local-first community detection method for dynamic graphs, *IEEE International Conference on Big Data (Big Data)*, 2016, pp. 3318–3325. ⇒221, 240, 241, 243

[37] R. Kannan, S. Vempala, A. Vetta, On clusterings: Good, bad and spectral, *Journal of the ACM (JACM)*, **51**, 3(2004) 497–515. ⇒227

[38] B.W. Kernighan, S. Lin, An efficient heuristic procedure for partitioning graphs, *Bell System Technical Journal*, **49**, 2 (1970) 291–307. ⇒222

[39] A. Lancichinetti, S. Fortunato, F. Radicchi, Benchmark graphs for testing community detection algorithms, *Physical review E*, **78**, 4 (2008) 046110. ⇒ 225, 226, 227

[40] F. Meng, F. Zhang, M. Zhu, Y. Xing, Z. Wang, J. Shi, Incremental density-based link clustering algorithm for community detection in dynamic networks, *Mathematical Problems in Engineering*, 2016. ⇒221, 234, 235, 243

[41] Z. Mu, M. Fanrong, Z. Yong, Density-based link clustering algorithm for overlapping community detection, *Journal of computer research and development*, **50**, 12 (2013) 2520. ⇒234

[42] K. Nath, S. Roy, Detecting intrinsic communities in evolving networks, *Social Network Analysis and Mining*, **9**,1 (2019)1–15. ⇒223, 239, 240, 243

[43] M. E. J. Newman, M. Girvan, Finding and evaluating community structure in networks, *Physical review E*, **69**, 2 (2004) 026113. ⇒222, 227

[44] M. E. Newman, Spectral methods for community detection and graph partitioning, *Physical Review E*, **88**, 4 (2013) 042822. ⇒222

[45] M. E. Newman, Detecting community structure in networks, *The European Physical Journal B-Condensed Matter and Complex Systems*, **38**, 2 (2004) 321–330. ⇒231

[46] N. P. Nguyen, T. N. Dinh, Y. Xuan, , M. T. Thai, Adaptive algorithms for detecting community structure in dynamic social networks, *Proceedings IEEE INFOCOM*, 2011, pp. 2282–2290. ⇒221, 232, 233, 243

[47] N. P. Nguyen, T. N. Dinh, S. Tokala, M. T. Thai, Overlapping communities in dynamic networks: their detection and mobile applications, *Proceedings of the 17th annual international conference on Mobile computing and networking*, 2011, pp. 85–96. ⇒234, 235, 243

[48] G. Palla, A. L. Barabási, T. Vicsek, Quantifying social group evolution, *Nature*, **446**, 7136 (2007) 664–667. ⇒224

[49] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, D. Parisi, Defining And identifying communities in networks, *Proceedings of the National Academy of Sciences*, **101**, 9 (2004) 2658–2663. ⇒222, 227

[50] U. N. Raghavan, R. Albert, R. Kumara, Near linear time algorithm to detect community structures in large-scale networks, *Physical Review E*, **76**, 3 (2007) 036106. ⇒222, 236

[51] W. M. Rand, Objective criteria for the evaluation of clustering methods, *Journal of the American Statistical association*, **66**, 336 (1971) 846–850. ⇒227

[52] N. Schlitter, T. Falkowski, J. Lässig, DenGraph-HO: a density-based hierarchical graph clustering algorithm, *Expert Systems*, **31**, 5 (2014) 469–479. ⇒234, 235, 243

[53] N. Sengupta, M. Hamann, D. Wagner, Benchmark generator for dynamic overlapping communities in networks, *IEEE International Conference on Data Mining (ICDM)*, 2017, pp. 415–424. ⇒225, 227

[54] J. Shang, L. Liu, F. Xie, Z. Chen, J. Miao, X. Fang, C. Wu, A real-time detecting algorithm for tracking community structure of dynamic networks, *ArXiv preprint arXiv: 1407.2683*, 2014. ⇒238, 239, 243

[55] J. Shang, L. Liu, X. Li, F. Xie, C. Wu, Targeted revision: A learning-based approach for incremental community detection in dynamic networks, *Physica A: Statistical Mechanics and its Applications*, **443**, (2016)70–85. ⇒224, 233, 243

[56] J. Shi, J. Malik, Normalized cuts and image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**, 8, (2000) 888–905. ⇒222

[57] Z. Sun, J. Sheng, B. Wang, A. Ullah, F. Khawaja, Identifying Communities in Dynamic Networks Using Information Dynamics, *Entropy*, **22**, 4(2020) 425. ⇒ 223, 236, 237, 243

[58] M. Takaffoli, F. Sangi, J. Fagnan, O. R. Zäiane, Community evolution mining in dynamic social networks, *Procedia-Social and Behavioral Sciences*, **22**, (2011) 49–58. ⇒224

[59] Y. Van Gennip, B. Hunter, R. Ahn, P. Elliott, K. Luh, M. Halvorson, ... and P. J. Brantingham, Community detection using spectral clustering on sparse geosocial data, *SIAM Journal on Applied Mathematics*, **73**, 1, (2013) 67–83. ⇒ 222

[60] D. R. White, F. Harary, M. Sobel, M. Becker,The cohesiveness of blocks in social networks: node connectivity and conditional density, *Sociological Methodology*, **31**, 1(2001) p. 305–359. ⇒222

[61] J. Xie, B. K. Szymanski, Community detection using a neighborhood strength driven label propagation algorithm, *IEEE Network Science Workshop*, 2011, pp. 188–195. ⇒223

[62] J. Xie, B. K. Szymanski, Towards linear time overlapping community detection in social networks, *Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer*, Berlin, Heidelberg, 2012, pp. 25–36. ⇒223, 236

[63] J. Xie, M. Chen, B. K. Szymanski, LabelrankT: Incremental community detection in dynamic networks via label propagation, *Proceedings of the workshop on dynamic networks management and mining*, 2013, pp. 25–32. ⇒235, 237, 243

[64] J. Xie, B. K. Szymanski, Labelrank: A stabilized label propagation algorithm for community detection in networks, *IEEE 2nd Network Science Workshop (NSW)*, 2013, pp. 138–143. ⇒235

[65] S. Yin, S. Chen, Z. Feng, K. Huang, D. He, P. Zhao, M. Y. Yang, Node-grained incremental community detection for streaming networks, *IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*, 2016, pp. 585–592. ⇒ 221, 238, 239, 243

[66] N. Yuruk, M. Mete, X. Xu, T. A. J. Schweiger, A Divisive Hierarchical Structural Clustering Algorithm for Networks, *Seventh IEEE Int. Conf. on Data Mining Workshops (ICDMW 2007)*, Omaha, NE, 2007, pp. 441-448. ⇒222

[67] N. Yuruk, M. Mete, X. Xu, T. A. J. Schweiger, AHSCAN: Agglomerative Hierarchical Structural Clustering Algorithm for Networks, *International Conference on Advances in Social Network Analysis and Mining*, Athens, 2009, pp. 72–77. ⇒222

[68] N. Zarayeneh, A. Kalyanaraman, A fast and efficient incremental approach toward dynamic community detection, *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2019, pp. 9–16. ⇒238, 239, 243

[69] Z. Zhao, C. Li, X. Zhang, F. Chiclana, E. H. Viedma, An incremental method to detect communities in dynamic evolving social networks, *Knowledge-Based Systems*, **163**, 2019, pp. 404–415. ⇒241

[70] D. Zhuang, M. J. Chang, M. Li, DynaMo: Dynamic community detection by incrementally maximizing modularity, *IEEE Transactions on Knowledge and Data Engineering*, 2019. ⇒221, 233, 243

Received: June 2, 2021 • Revised: October 19, 2021

Acta Univ. Sapientiae, Informatica **13,** 2 (2021) 251–264

DOI: 10.2478/ausi-2021-0011

On Laplacian spectrum of unitary Cayley graphs

S. PIRZADA
Department of Mathematics, University of Kashmir, Srinagar, India
email: pirzadasd@kashmiruniversity.ac.in

Z. BARATI
Department of Mathematics, Kosar University of Bojnord, Bojnord, Iran
email: za.barati87@gmail.com

M. AFKHAMI
Department of Mathematics, University of Neyshabur, Neyshabur, Iran
email: mojgan.afkhami@yahoo.com

Abstract. Let R be a commutative ring with unity $1 \neq 0$ and let $R^\times$ be the set of all unit elements of R. The unitary Cayley graph of R, denoted by $G_R = \mathrm{Cay}(R, R^\times)$, is a simple graph whose vertex set is R and there is an edge between two distinct vertices x and y of R if and only if $x - y \in R^\times$. In this paper, we determine the Laplacian and signless Laplacian eigenvalues for the unitary Cayley graph of a commutative ring. Also, we compute the Laplacian and signless Laplacian energy of the graph G_R and its line graph.

1 Introduction

We consider finite commutative rings R with unit element $1 \neq 0$. Let $R^\times$ be the set of all unit elements of R. We know that an Artinian ring R can be written as $R \cong R_1 \times \cdots \times R_t$, where R_i is a finite local ring with maximal ideal $\mathfrak{M}_i$, for all $1 \leqslant i \leqslant t$. This decomposition is unique up to permutation of factors. We

Computing Classification System 1998: G.2.2
Mathematics Subject Classification 2010: 05C22, 05C50, 05C76
Key words and phrases: unitary Cayley graph, Laplacian spectrum, signless Laplacian spectrum, Laplacian energy

denote the (finite) residue field $\frac{R_i}{\mathfrak{M}_i}$ by K_i and $f_i = |K_i| = \frac{|R_i|}{|\mathfrak{M}_i|}$. Also, assume that $f_1 \leqslant f_2 \leqslant \cdots \leqslant f_t$.

A simple graph G consists of a vertex set $V(G) = \{v_1, v_2, \ldots, v_n\}$ and the edge set $E(G) = \{e_1, e_2, \ldots, e_m\}$. We call $|V(G) = n|$ and $|E(G)| = m$, respectively, as the order and the size of the graph G. The complement of G, denoted by $\overline{G}$, is the graph whose vertex set is same as that of G and two vertices are adjacent in $\overline{G}$ if and only if they are not adjacent in G. A complete graph on n vertices is denoted by K_n. A graph G is multipartite if its vertex set can be partitioned into non-empty subsets, called partite sets, such that no two vertices in the same part are adjacent. A multipartite graph is complete if every vertex of a partite set is adjacent to each vertex of the other partite sets. A complete multipartite graph with k parts is denoted by $K_{n_1,n_2,\ldots,n_k}$ where n_i is the number of vertices in the i-th part of the graph.

The join of two graphs G_1 and G_2, denoted by $G_1 \vee G_2$, is the graph with vertex set $V(G_1) \cup V(G_2)$ and edge set $E(G_1) \cup E(G_2) \cup \{xy; x \in V(G_1), y \in V(G_2)\}$. The direct product of G_1 and G_2, denoted by $G_1 \otimes G_2$, is the graph with vertex set $V(G_1) \times V(G_2)$ in which (u_1, v_1) and (u_2, v_2) are adjacent if u_1 and u_2 are adjacent in G_1 and v_1 and v_2 are adjacent in G_2. For other undefined notations and terminology from graph theory and spectral graph theory, the readers are referred to [6, 18].

The unitary Cayley graph of R, denoted by $G_R = \mathrm{Cay}(R, R^\times)$, is a (simple) graph whose vertex set is R and two distinct vertices x and y of R are adjacent if and only if $x - y \in R^\times$. Some recent results on unitary Cayley graphs can be seen in [16]. If $G = \mathbb{Z}_n$ is the finite cyclic group of order n and the set S consists of two elements, the standard generator of G and its inverse, then the Cayley graph is the cycle C_n. More generally, the Cayley graphs of finite cyclic groups are exactly the circulant graphs. Some examples of unitary Cayley graphs are given in Figure 1.

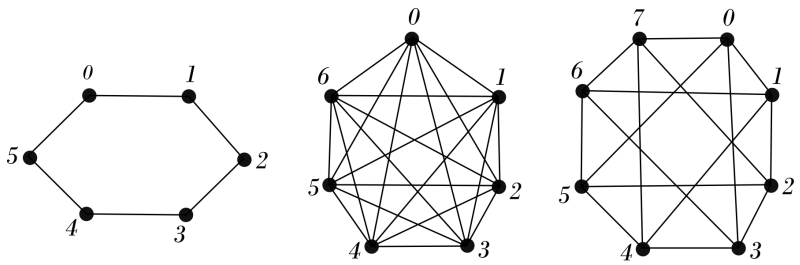


Figure 1: The unitary Cayley graphs for $\mathbb{Z}_6$, $\mathbb{Z}_7$, $\mathbb{Z}_8$

The following proposition is a basic consequence of the definition and it was illustrated in [1].

Proposition 1 *Let* R *be a commutative ring.*

(a) Then G_R *is a* $|R^\times|$*-regular graph.*

(b) If R *is a local ring with maximal ideal* $\mathfrak{M}$*, then* G_R *is a complete multipartite graph whose partite sets are the cosets of* $\mathfrak{M}$ *in* R*. In particular,* G_R *is a complete graph if and only if* R *is a field.*

(c) If R *is an Artinian ring and* $R \cong R_1 \times \ldots \times R_t$ *as a product of local rings, then* $G_R \cong \otimes_{i=1}^{t} G_{R_i}$*. Hence,* G_R *is a direct product of complete multipartite graphs.*

The adjacency matrix A of a graph G is a $(0,1)$-square matrix of order n whose (i,j)-entry is equal to 1, if v_i is adjacent to v_j and equal to 0, otherwise. The eigenvalues of A are the eigenvalues of the graph G. The set of all eigenvalues of G is called the *spectrum* of G. If $\lambda_1 \geq \cdots \geq \lambda_k$ are the eigenvalues of G with multiplicities $r_1, \ldots, r_k$, respectively, the spectrum of G is denoted by $\mathrm{Spec}(G) = \begin{pmatrix} \lambda_1 & \ldots & \lambda_k \\ r_1 & \ldots & r_k \end{pmatrix}$. The energy of a graph was introduced by Gutman [13] and is defined as the sum of the absolute values of all the eigenvalues of a graph G and it is denoted by $E(G)$.

Kiani et al. [15] obtained the following result about the eigenvalues of the unitary Cayley graph. Also, they computed the energy of the unitary Cayley graph of a finite commutative ring R.

Theorem 2 [15] *Let* R *be a finite ring.*

(a) If R *is a finite local ring with the maximal ideal* $\mathfrak{M}$ *of size* m *and* $\frac{|R|}{m} = f$*, then*

$$Spec(G_R) = \begin{pmatrix} |R^\times| & 0 & -m \\ 1 & |R| - f & f-1 \end{pmatrix}.$$

In particular, if $\mathbb{F}_q$ *is the field with* q *elements, then*

$$Spec(G_{\mathbb{F}_q}) = \begin{pmatrix} q-1 & -1 \\ 1 & q-1 \end{pmatrix}.$$

(b) Let R *be a finite commutative ring, where* $R \cong R_1 \times R_2 \times \ldots \times R_t$ *and* R_i *is a local ring with maximal ideal* $\mathfrak{M}_i$ *of size* m_i *for all* $1 \leqslant i \leqslant t$. *Then the eigenvalues of* G_R *are:*

(b-1) $(-1)^{|C|}\dfrac{|R^\times|}{\prod_{j\in C}|R_j^\times|/m_j}$ *with multiplicity* $\prod_{j\in C}|R_j^\times|/m_j$ *for all subsets* C *of the set* $\{1, 2, \ldots, t\}$.

(b-2) 0 *with multiplicity* $|R| - \prod_{i=1}^{t}(1 + |R_i^\times|/m_i)$

Theorem 3 [15] *Let* $R \cong R_1 \times R_2 \times \ldots \times R_t$ *be a finite commutative ring where* R_i *is a local ring for all* $1 \leqslant i \leqslant t$. *Then* $E(G_R) = 2^t|R^\times|$.

Let $D(G) = \mathrm{diag}(d_1, d_2, \ldots, d_n)$ be the diagonal matrix associated to the graph G, where $d_i = \deg(v_i)$ is the degree of the vertex v_i, for all $1 \leqslant i \leqslant n$. The matrices $L(G) = D(G) - A(G)$ and $|L|(G) = D(G) + A(G)$ are respectively, called the Laplacian and the signless Laplacian matrices of G. Their spectrum are respectively, the Laplacian spectrum and the signless Laplacian spectrum of the graph G. We denote the Laplacian spectrum and the signless Laplacian spectrum of the graph G by $\mathrm{Spec}_L(G)$ and $\mathrm{Spec}_{|L|}(G)$, respectively. Both the matrices $L(G)$ and $|L|(G)$ are real symmetric, positive semi-definite and therefore their eigenvalues are non-negative real numbers. Let $0 = \mu_1 \leqslant \mu_2 \leqslant \cdots \leqslant \mu_n$ and $\mu_n^+ \leqslant \mu_{n-1}^+ \leqslant \cdots \leqslant \mu_1^+$ be respectively, the Laplacian spectrum and the signless Laplacian spectrum of G. It is known that the smallest eigenvalue of $L(G)$ is 0 with multiplicity equal to the number of connected components of G. So, $\mu_2 > 0$ if and only if G is connected. Also, the least eigenvalue of the signless Laplacian matrix of a connected graph is 0 if and only if the graph is bipartite. In this case, 0 is a simple eigenvalue. Furthermore, it is easy to see that $\mathrm{tr}(L(G)) = \sum_{i=1}^{n}\mu_i = 2m$ and $\mathrm{tr}(|L|(G)) = \sum_{i=1}^{n}\mu_i^+ = 2m$. Recent work on Laplacian eigenvalues can be seen in [2, 5, 9, 10, 11, 12]. The Laplacian energy of a graph G defined by Gutman and Zhou [14] is $LE(G) = \sum_{i=1}^{n}|\mu_i - \frac{2m}{n}|$. The Laplacian energy, which is an extension of graph energy concept, has found remarkable chemical applications (see [24]). For recent development on $LE(G)$ see [7, 8] and the references therein. The signless Laplacian energy $|L|E(G)$ of G, in analogy to $LE(G)$, is defined as $|L|E(G) = \sum_{i=1}^{n}|\mu_i^+ - \frac{2m}{n}|$. Recent work on Laplacian eigenvalues can be seen in [19].

The rest of the paper is organized as follows. In Section 2, we determine the Laplacian spectrum and the Laplacian energy of the unitary Cayley graph G_R. Also, we completely obtain the signless Laplacian spectrum of the graph G_R

and compute the signless Laplacian energy of G_R. Further, we compute the Laplacian and signless Laplacian energy of the line graph of G_R.

2 Laplacian spectrum of unitary Cayley graphs

We begin with the following theorem, which gives the Laplacian spectrum of the join of two graphs G_1 and G_2.

Theorem 4 [17] *Let G_1 and G_2 be two graphs with n_1 and n_2 vertices, respectively. Suppose that $0 = \lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_{n_1}$ and $0 = \mu_1 \leqslant \mu_2 \leqslant \cdots \leqslant \mu_{n_2}$ are the Laplacian eigenvalues of G_1 and G_2, respectively. Then the Laplacian eigenvalues of the graph $G_1 \vee G_2$ are*

(i) 0 *with multiplicity* 1,

(ii) $\lambda_i + n_2$ *with multiplicity* 1 *for all* $2 \leqslant i \leqslant n_1$,

(iii) $\mu_j + n_1$ *with multiplicity* 1 *for all* $2 \leqslant j \leqslant n_2$,

(iv) $n_1 + n_2$ *with multiplicity* 1.

Now, we have the following observation.

Lemma 5 *If $G = K_{n_1,n_2,\ldots,n_k}$, where $n_i \in \mathbb{N}$ for all $1 \leqslant i \leqslant k$, then the Laplacian eigenvalues of G are*

(i) 0 *with multiplicity* 1,

(ii) $\alpha_i = \sum_{\substack{j=1 \\ j\neq i}}^{k} n_j$ *with multiplicity* $n_i - 1$ *for all* $1 \leqslant i \leqslant k$,

(iii) $n_1 + n_2 + \cdots + n_k$ *with multiplicity* $k - 1$.

Proof. We induct on k. For $k = 2$, we have $G = \overline{K}_{n_1} \vee \overline{K}_{n_2}$. So, by Theorem 4, we have that

$$\mathrm{Spec}_L(K_{n_1,n_2}) = \begin{pmatrix} 0 & n_1 & n_2 & n_1+n_2 \\ 1 & n_2-1 & n_1-1 & 1 \end{pmatrix}.$$

Assume that the hypothesis is true for $K_{n_1,n_2,\ldots,n_k}$.

We prove it for the graph $K_{n_1,n_2,\ldots,n_k,n_{k+1}}$.

Clearly, $K_{n_1,n_2,\ldots,n_k,n_{k+1}} \cong K_{n_1,n_2,\ldots,n_k} \vee \overline{K}_{n_k}$.

Now, by Theorem 4, it is easy to see that the Laplacian eigenvalues of $K_{n_1,n_2,\ldots,n_k,n_{k+1}}$ are

(i) 0 with multiplicity 1,

(ii) $\sum_{\substack{j=1 \\ j\neq i}}^{k+1} n_j$ with multiplicity $n_i - 1$ for all $1 \leqslant i \leqslant k+1$,

(iii) $n_1 + n_2 + \cdots + n_k + n_{k+1}$ with multiplicity k. □

At first, we assume that R is a local ring.

Proposition 6 *Let $(R, \mathfrak{M})$ be a local ring with $|\mathfrak{M}| = m$ and $|\frac{R}{\mathfrak{M}}| = f$. Then*

$$Spec_L(G_R) = \begin{pmatrix} 0 & |R^\times| & |R| \\ 1 & |R| - f & f - 1 \end{pmatrix}.$$

In particular, if $R = \mathbb{F}_q$ is the field with q elements, then

$$Spec_L(G_{\mathbb{F}_q}) = \begin{pmatrix} 0 & q \\ 1 & q - 1 \end{pmatrix}.$$

Proof. It is easy to see that G_R is a complete multipartite graph in which every partite set is a coset of $\mathfrak{M}$. So, G_R is the join of f copies of the empty graph $\overline{K}_m$. Now, by Lemma 5, we have

$$\mathrm{Spec}_L(G_R) = \begin{pmatrix} 0 & |R| - m & |R| \\ 1 & |R| - f & f - 1 \end{pmatrix}.$$

Since $|R| - m = |R^\times|$, therefore

$$\mathrm{Spec}_L(G_R) = \begin{pmatrix} 0 & |R^\times| & |R| \\ 1 & |R| - f & f - 1 \end{pmatrix}.$$

□

The Laplacian spectrum of the direct product of graphs has been described completely only when the factor graphs are regular. The Laplacian eigenvalues of the direct product of two regular graphs are listed in the following theorem.

Theorem 7 [3] *Let G_1 be an r_1-regular graph with n_1 vertices and G_2 be an r_2-regular graph with n_2 vertices. Let $Spec_L(G_1) = (\lambda_1, \lambda_2, \ldots, \lambda_{n_1})$ and $Spec_L(G_2) = (\mu_1, \mu_2, \ldots, \mu_{n_2})$. Then the eigenvalues of the graph $G_1 \otimes G_2$ are $r_1\mu_j + r_2\lambda_i - \mu_j\lambda_i$ for all $1 \leqslant i \leqslant n_1$ and $1 \leqslant j \leqslant n_2$.*

In the following theorem, we obtain the Laplacian eigenvalues of G_R with their multiplicities. Here, $|R_S^\times|$ stands for $|R_{s_1}^\times \times R_{s_2}^\times \ldots \times R_{s_k}^\times|$, where $S = \{s_1, \ldots, s_k\} \subseteq \{1, 2, \ldots, t\}$ (if $S = \emptyset$, then we define $|R_S^\times| = 1$).

Theorem 8 *Let R be a finite commutative ring such that $R \cong R_1 \times R_2 \times \cdots \times R_t$, where $(R_i, \mathfrak{M}_i)$ is a local ring with $|\mathfrak{M}_i| = m_i$ and $|\frac{R_i}{\mathfrak{M}_i}| = f_i$. Then the Laplacian eigenvalues of G_R are*

(i) 0 *with multiplicity* 1,

(ii) $|R^\times|$ *with multiplicity* $|R| - \prod_{i=1}^{t} f_i$,

(iii) λ_A *with multiplicity* $\prod_{i \in A'}(f_i - 1)$ *for all* $A \subsetneq \{1, 2, \ldots, t\}$, *where*

$$\lambda_A = |R_A^\times| \sum_{\substack{C=\{i_1, i_2, \ldots, i_k\} \subseteq A' \\ k=1}}^{|A'|} (-1)^{|C|-1} |R_{i_1}||R_{i_2}| \ldots |R_{i_k}| \frac{|R_{A'}^\times|}{|R_C^\times|}$$

and A' is the complement of A.

Proof. We use induction on t. For $t = 1$ and the local ring $R \cong R_1$, by Proposition 6, we have

$$\mathrm{Spec}_L(G_R) = \begin{pmatrix} 0 & |R^\times| & |R| \\ 1 & |R| - f & f - 1 \end{pmatrix}.$$

Note that $\emptyset$ is the only proper subset of $\{1\}$ and $\lambda_\emptyset = |R|$. So, we are done in this case. Now, assume that the Laplacian eigenvalues of $R_1 \times R_2 \times \cdots \times R_{t-1}$ are

(i) 0 with multiplicity 1,

(ii) $|R_1^\times \times R_2^\times \times \cdots \times R_{t-1}^\times|$ with multiplicity $|R_1 \times R_2 \times \cdots \times R_{t-1}| - \prod_{i=1}^{t-1} f_i$,

(iii) λ_A with multiplicity $\prod_{i \in A'}(f_i - 1)$ for all $A \subsetneq \{1, 2, \ldots, t-1\}$, where

$$\lambda_A = |R_A^\times| \sum_{\substack{C=\{i_1, i_2, \ldots, i_k\} \subseteq A' \\ k=1}}^{|A'|} (-1)^{|C|-1} |R_{i_1}||R_{i_2}| \ldots |R_{i_k}| \frac{|R_{A'}^\times|}{|R_C^\times|}.$$

Now, we determine the Laplacian eigenvalues of G_R when $R \cong R_1 \times R_2 \times \cdots \times R_{t-1} \times R_t$. We know that $G_R \cong G_{R_1 \times R_2 \times \cdots \times R_{t-1}} \otimes G_{R_t}$. Note that the graphs $G_{R_1 \times R_2 \times \cdots \times R_{t-1}}$ and G_{R_t} are regular, so we can use Theorem 7. Since

$$\mathrm{Spec}_L(G_{R_t}) = \begin{pmatrix} \mu_1 = 0 & \mu_2 = |R_t^\times| & \mu_3 = |R_t| \\ 1 & |R_t| - f_t & f_t - 1 \end{pmatrix},$$

we have the following cases to consider.

Case 1. For $\mu_1 = 0$, we have the following eigenvalues.

1.1. 0 with multiplicity 1,

1.2. $|R_1^\times \times R_2^\times \times \cdots \times R_{t-1}^\times| \times |R_t^\times|$ with multiplicity $|R_1 \times R_2 \times \cdots \times R_{t-1}| - \prod_{i=1}^{t-1} f_i$,

1.3. $\lambda_A \times |R_t^\times|$ with multiplicity $\prod_{i \in A'}(f_i - 1)$ for all $A \subsetneq \{1, 2, \ldots, t-1\}$.

Case 2. For $\mu_2 = |R_t^\times|$, we obtain the following eigenvalues.

2.1. $|R_1^\times \times R_2^\times \times \cdots \times R_{t-1}^\times||R_t^\times|$ with multiplicity $|R_t| - f_t$,

2.2. $|R_1^\times \times R_2^\times \times \cdots \times R_{t-1}^\times||R_t^\times|$ with multiplicity $(|R_1 \times R_2 \times \cdots \times R_{t-1}| - \prod_{i=1}^{t-1} f_i)(|R_t| - f_t)$,

2.3. $|R_1^\times \times R_2^\times \times \cdots \times R_{t-1}^\times||R_t^\times|$ with multiplicity $\sum_{A \subsetneq \{1,2,\ldots,t-1\}} \prod_{i \in A'}(f_i - 1)(|R_t| - f_t)$.

Therefore, in this case, we see that the eigenvalue is equal to $|R_1^\times \times R_2^\times \times \cdots \times R_{t-1}^\times||R_t^\times|$ and this implies that $|R_1^\times \times R_2^\times \times \cdots \times R_{t-1}^\times||R_t^\times|$ is an eigenvalue with multiplicity $|R_1 \times R_2 \times \cdots \times R_{t-1}|(|R_t| - f_t)$.

Case 3. For $\mu_3 = |R_t|$, the following eigenvalues can be obtained.

3.1. $|R_1^\times \times R_2^\times \times \cdots \times R_{t-1}^\times||R_t|$ with multiplicity $f_t - 1$,

3.2. $|R_1^\times \times R_2^\times \times \cdots \times R_{t-1}^\times||R_t^\times|$ with multiplicity $(|R_1 \times R_2 \times \cdots \times R_{t-1}| - \prod_{i=1}^{t-1} f_i)(f_t - 1)$,

3.3. $|R_1^\times \times R_2^\times \times \cdots \times R_{t-1}^\times||R_t| + \lambda_A|R_t^\times| - \lambda_A|R_t|$ with multiplicity $(\prod_{i \in A'}(f_i - 1))(f_t - 1)$ for all $A \subsetneq \{1, 2, \ldots, t-1\}$.

Thus, we conclude the following.

(i) By case (1.1), 0 with multiplicity 1 is a Laplacian eigenvalue of G_R.

(ii) By cases (1.2),(2.1), (2.2),(2.3) and (3.2), $|R_1^\times \times R_2^\times \times \cdots \times R_{t-1}^\times||R_t^\times|$ is a Laplacian eigenvalue of G_R. Its multiplicity is equal to

$$\left(|R_1 \times R_2 \times \cdots \times R_{t-1}| - \prod_{i=1}^{t-1} f_i\right) + \left(|R_1 \times R_2 \times \cdots \times R_{t-1}|(|R_t| - f_t)\right) +$$

$$\left(|R_1 \times R_2 \times \cdots \times R_{t-1}| - \prod_{i=1}^{t-1} f_i\right)(f_t - 1) = |R_1 \times R_2 \times \cdots \times R_t| - \prod_{i=1}^{t} f_i$$

(iii) For $\mathfrak{A} \subsetneq \{1, 2, \ldots, t\}$, three cases (1.3), (3.1) and (3.3) cover all eigenvalues with the type $\lambda_{\mathfrak{A}}$.

(a) From case (1.3), $\lambda_A \times |R_t^\times|$ with multiplicity $\prod_{i \in A'}(f_i - 1)$ is a Laplacian eigenvalue of G_R for all $A \subsetneq \{1, 2, \ldots, t-1\}$. Note that if we set $\mathfrak{A} = A \cup \{t\}$, then $\lambda_{\mathfrak{A}} = \lambda_A \times |R_t^\times|$.

(b) From case (3.1), $|R_1^\times \times R_2^\times \times \cdots \times R_{t-1}^\times||R_t|$ with multiplicity $f_t - 1$ is a Laplacian eigenvalue of G_R. By setting $\mathfrak{A} = \{1, 2, , \ldots, t-1\}$, we have $\lambda_{\mathfrak{A}} = |R_1^\times \times R_2^\times \times \cdots \times R_{t-1}^\times||R_t|$.

(c) From case (3.3), $|R_1^\times \times R_2^\times \times \cdots \times R_{t-1}^\times||R_t| + \lambda_A|R_t^\times| - \lambda_A|R_t|$ with multiplicity $(\prod_{i \in A'}(f_i - 1))(f_t - 1)$ is a Laplacian eigenvalue of G_R, for all $A \subsetneq \{1, 2, \ldots, t-1\}$. This case covers all eigenvalues like $\lambda_{\mathfrak{A}}$, when $\mathfrak{A}$ is a proper subset of the set $\{1, 2, \ldots, t\}$ and $t \notin \mathfrak{A}$. □

Now, we compute the Laplacian energy of the unitary Cayley graph, when R is a finite commutative ring. We start with the local case.

Lemma 9 *Let R be a finite local commutative ring. Then $LE(G_R) = 2|R^\times|$.*

Proof. First, note that in the graph G_R, we have $\frac{2m}{n} = |R^\times|$. Since the Laplacian spectrum of G_R is

$$\mathrm{Spec}_L(G_R) = \begin{pmatrix} 0 & |R^\times| & |R| \\ 1 & |R| - f & f - 1 \end{pmatrix},$$

we have $LE(G_R) = 2|R^\times|$. □

Lemma 10 *Let $R \cong R_1 \times R_2$, where $(R_1, \mathfrak{M}_1)$ and $(R_2, \mathfrak{M}_2)$ are local rings. Then*

$$LE(G_R) = 2^2|R^\times|.$$

Proof. We know that $G_R \cong G_{R_1} \otimes G_{R_2}$. Now, let $\mathrm{Spec}_L(G_{R_1}) = (\lambda_1, \lambda_2, \ldots, \lambda_{|R_1|})$ and $\mathrm{Spec}_L(G_{R_2}) = (\mu_1, \mu_2, \ldots, \mu_{|R_2|})$. Then, by Theorem 7, we have

$$\begin{aligned} LE(G_R) &= \sum_{i=1}^{|R_1|}\sum_{j=1}^{|R_2|} \left||R_1^\times|\mu_j + |R_2^\times|\lambda_i - \mu_j\lambda_i - |R_1^\times||R_2^\times|\right| \\ &= \sum_{i=1}^{|R_1|}\sum_{j=1}^{|R_2|} \left|(\mu_j - |R_2^\times|)\right|\left|(\lambda_i - |R_1^\times|)\right| \\ &= LE(G_{R_1})LE(G_{R_2}) = (2|R_1^\times|)(2|R_2^\times|) = 2^2|R^\times|. \end{aligned}$$

□

Theorem 11 *Let R be a finite commutative ring such that $R \cong R_1 \times R_2 \times \cdots \times R_t$, where R_i is a local ring for all $1 \leqslant i \leqslant t$. Then $LE(G_R) = 2^t|R^\times|$.*

Proof. This follows by using induction on t and in view of Lemmas 9 and 10. □

The following results concern about the signless Laplacia spectrum of G_R. The proofs are omitted since they are similar to the proofs on the Laplacian spectrum.

Proposition 12 *Let $(R, \mathfrak{M})$ be a local ring with $|\mathfrak{M}| = m$ and $|\frac{R}{\mathfrak{M}}| = f$. Then*

$$Spec_{|L|}(G_R) = \begin{pmatrix} |R^\times| - m & |R^\times| & 2|R^\times| \\ f - 1 & |R| - f & 1 \end{pmatrix}.$$

In particular, if $R = \mathbb{F}_q$ is the field with q elements, then

$$Spec_{|L|}(G_{\mathbb{F}_q}) = \begin{pmatrix} q-2 & 2(q-1) \\ q-1 & 1 \end{pmatrix}.$$

Theorem 13 *Let R be a finite commutative ring such that $R \cong R_1 \times R_2 \times \cdots \times R_t$, where $(R_i, \mathfrak{M}_i)$ is a local ring with $|\mathfrak{M}_i| = m_i$ and $|\frac{R_i}{\mathfrak{M}_i}| = f_i$. Then the signless Laplacian eigenvalues of G_R are*

(i) $2|R^\times|$ *with multiplicity* 1,

(ii) $|R^\times|$ *with multiplicity* $|R| - \prod_{i=1}^t f_i$,

(iii) λ_A *with multiplicity* $\prod_{i\in A'}(f_i - 1)$ *for all* $A \subsetneq \{1, 2, \ldots, t\}$ *where*

$$\lambda_A = |R^\times| + (-1)^{|A'|}\prod_{i\in A}|R_i^\times|\prod_{j\in A'}|m_j|.$$

If R be a local finite commutative ring, it is easy to see that the signless Laplacian energy of G_R is given by $|L|E(G_R) = 2|R^\times|$. Further, if $R \cong R_1 \times R_2$, where R_1 and R_2 are local rings, then $|L|E(G_R) = 2^2|R^\times|$.

Thus, we have the following observation.

Theorem 14 *Let R be a finite commutative ring such that $R \cong R_1 \times R_2 \times \cdots \times R_t$, $(t \geqslant 2)$, where R_i is a local ring for all $1 \leqslant i \leqslant t$. Then $|L|E(G_R) = 2^t|R^\times|$.*

Let G be a graph with n vertices and m edges. The line graph $L(G)$ of G is a simple graph whose vertex set is the set of edges of G and two vertices of $L(G)$ are adjacent if and only if the corresponding edges in G have a vertex in common. So, $n_{L(G)}$ (the number of vertices of $L(G)$) equals m. Also, it is easy to see that if G is an r-regular graph, then $L(G)$ is a $(2r-2)$-regular graph.

Theorem 15 [4] *Let G be an r-regular graph $(r \geqslant 2)$ with n vertices and m edges. Then*

(a) The Laplacian eigenvalues of the graph $L(G)$ are

(i) $2 - \lambda_i$, where λ_i is a Laplacian eigenvalue of G for all $1 \leqslant i \leqslant n$,

(ii) $r - 2$ with multiplicity $m - n$.

(b) The signless Laplacian eigenvalues of the graph $L(G)$ are

(i) $\lambda_i^+ + 2r - 4$, where λ_i^+ is a signless Laplacian eigenvalue of G for all $1 \leqslant i \leqslant n$,

(ii) $2r - 4$ with multiplicity $m - n$.

Now, we compute the Laplacian energy of the line graph of the unitary Cayley graphs. If $|R^\times| = 1$, then $L(G_R)$ is an empty graph. So in this case, $LE(L(G_R)) = 0$. Thus, we suppose that $|R^\times| \geqslant 2$. Now, by Theorem 15, the spectrum of $L(G)$ consists of the following eigenvalues.

(i) $2 - \lambda_i$, where λ_i is a Laplacian eigenvalue of G_R for all $1 \leqslant i \leqslant |R|$,

(ii) $|R^\times| - 2$ with multiplicity $|R||R^\times|/2 - |R|$.

Proposition 16 *Let R be a finite commutative ring with $|R^\times| \geqslant 2$. Then*

$$LE(L(G_R)) = \frac{|R|\left(|R^\times|^2 + 4|R^\times| - 8\right)}{2}.$$

Proof. Since G_R is $|R^\times|$-regular, $L(G_R)$ is a $(2|R^\times|-2)$-regular graph. So,

$$2m_{L(G_R)}/n_{L(G_R)} = 2|R^\times| - 2,$$

where $n_{L(G_R)}$ and $m_{L(G_R)}$ are the number of vertices and edges of $L(G_R)$, respectively. We have

$$\begin{aligned}
LE(L(G_R)) &= \sum_{i=1}^{|R|} \left|2 - \lambda_i - (2|R^\times| - 2)\right| + \sum_{i=1}^{|R||R^\times|/2-|R|} \left||R^\times| - 2 - (2|R^\times| - 2))\right| \\
&= \sum_{i=1}^{|R|} \left|-\lambda_i - 2|R^\times| + 4\right| + \sum_{i=1}^{|R||R^\times|/2-|R|} |R^\times| \\
&= \sum_{i=1}^{|R|} (\lambda_i + 2|R^\times| - 4) + \sum_{i=1}^{|R||R^\times|/2-|R|} |R^\times| \qquad \text{(Since } |R^\times| \geqslant 2\text{)} \\
&= \sum_{i=1}^{|R|} \lambda_i + 2|R||R^\times| - 4|R| + (|R||R^\times|/2 - |R|)|R^\times| \\
&= |R||R^\times| + 2|R||R^\times| - 4|R| + (|R||R^\times|/2 - |R|)|R^\times| \\
&\text{(Since } \sum_{i=1}^{|R|} \lambda_i = |R||R^\times|\text{)} \\
&= \frac{|R|\left(|R^\times|^2 + 4|R^\times| - 8\right)}{2}.
\end{aligned}$$

□

The following result gives the signless Laplacian energy of the line graph of unitary Cayley graphs. The proof is similar to the Laplacian case.

Proposition 17 *Let R be a finite commutative ring with $|R^\times| \geqslant 2$. Then*

(i) If $f_1 = 2$, then $|L|E(L(G_R)) = 2\left(|R|(|R^\times| - 2) + 1\right)$.

(ii) $|L|E(L(G_R)) = 2|R|(|R^\times| - 2)$, otherwise.

References

[1] R. Akhtar, M. Boggess, T. Jackson-Henderson, I. Jiménez, R. Karpman, A. Kinzel, D. Pritikin, *On the unitary Cayley graph of a finite ring*, *Electron. J. Combin.* **16** (2009) R117. ⇒253

[2] A. Alhevaz, M. Baghipur, Hilal A. Ganie, S. Pirzada, Brouwer type conjecture for the eigenvalues of distance signless Laplacian matrix of a graph, *Linear Multilinear Algebra* **69, 13** (2021) 2423–2440. ⇒254

[3] S. Barik, R. B. Bapat, S. Pati, On the Laplacian spectra of product graphs, *Appl. Anal. Discrete Math.* **9** (2015) 39–58. ⇒256

[4] S. Barik, D. Kalita, S. Pati, G. Sahoo, Spectra of graphs resulting from various graph operations and products: a survey, *Spec. Matrices* **6** (2018) 323–342. ⇒261

[5] B. A. Rather, S. Pirzada, T. A. Naikoo, Y. Shang, On Laplacian eigenvalues of the zero-divisor graph associated to the ring of integers modulo n, *Mathematics* **9, 5** (2021) 482. ⇒254

[6] D. Cvetkovic, M. Doob, H. Sachs, Spectra of Graphs-Theory and Application, Academic Press, New York, 1980. ⇒252

[7] H. A. Ganie, S. Pirzada, A. Iványi, Energy, Laplacian energy of double graphs and new families of equienergetic graphs, *Acta Univ. Sapientiae, Inform.* **6** (2014) 89–117. ⇒254

[8] H. A. Ganie, S. Pirzada, E. T. Baskoro, On energy, Laplacian energy and p-fold graphs, *Electron. J. Graph Theory Appl.* **3** (2015) 94–107. ⇒254

[9] H. A. Ganie, S. Pirzada, R. Ul Shaban, X. Li, Upper bounds for the sum of Laplacian eigenvalues of a graph and Brouwer's conjecture, *Discrete Math. Algorithms Appl.* **11, 2** (2019) 195008 (15 pages). ⇒254

[10] H. A. Ganie, S. Pirzada, B. A. Rather, V. Trevisan, Further developments on Brouwer's conjecture for the sum of Laplacian eigenvalues of graphs, *Linear Algebra Appl.* **588** (2020) 1–18. ⇒254

[11] H. A. Ganie, S. Pirzada, V. Trevisan, On the sum of k largest Laplacian eigenvalues of a graph and clique number, *Mediterranean J. Mathematics* **18** (2021) Article No. 15. ⇒254

[12] H. A. Ganie, S. Pirzada, B. A. Rather, R. Ul Shaban, On Laplacian eigenvalues of graphs and Brouwer's conjecture, *J. Ramanujan Math. Soc.* **36,1** (2021) 13–21. ⇒254

[13] I. Gutman, The Energy of a Graph: Old and New Results, *Algebraic Combinatorics and Applications* (Springer, Berlin), 2001. ⇒253

[14] I. Gutman, B. Zhou, Laplacian energy of a graph, *Linear Algebra Appl.* **414** (2006) 29–37. ⇒254

[15] D. Kiani, M. M. H. Aghaei, Y. Meemark, B. Suntornpoch, Energy of unitary Cayley graphs and gcd-graphs, *Linear Algebra Appl.* **435** (2011) 1336–1343. ⇒253, 254

[16] W. Klotz, T. Sander, Some properties of unitary Cayley graphs, *Electron. J. Combin.* **14** (2007) Research Paper 45, 12 pages. ⇒252

[17] R. Merris, Laplacian graph eigenvectors, *Linear Algebra Appl.* **278** (1998) 221-236. ⇒255
[18] S. Pirzada, *An Introduction to Graph Theory*, Universities Press, Hyderabad, India, 2012. ⇒252
[19] S. Pirzada, H. A. Ganie, A. Alghamdi, On the sum of signless Laplacian spectra of graphs, *Carpathian Math. Publications* **11, 2** (2019) 407–417. ⇒254

Received: October 4, 2021 • Revised: October 11, 2021

Acta Univ. Sapientiae, Informatica **13,** 2 (2021) 265–287

DOI: 10.2478/ausi-2021-0012

Bitcoin daily close price prediction using optimized grid search method

Marzieh ROSTAMI
Department of Information Technology Engineering,
Raja University of Qazvin, Qazvin, Iran
email: mrz.rst@gmail.com

Mahdi BAHAGHIGHAT[1]
Department of Computer Engineering,
Imam Khomeini International University, Qazvin, Iran
email: Bahaghighat@eng.ikiu.ac.ir

Morteza MOHAMMADI ZANJIREH
Department of Computer Engineering,
Imam Khomeini International University, Qazvin, Iran
email: Zanjireh@eng.ikiu.ac.ir

Abstract. Cryptocurrencies are digital assets that can be stored and transferred electronically. Bitcoin (BTC) is one of the most popular cryptocurrencies that has attracted many attentions. The BTC price is considered as a high volatility time series with non-stationary and non-linear behavior. Therefore, the BTC price forecasting is a new, challenging, and open problem. In this research, we aim the predicting price using machine learning and statistical techniques. We deploy several robust approaches such as the Box-Jenkins, Autoregression (AR), Moving Average (MA), ARIMA, Autocorrelation Function (ACF), Partial Autocorrelation Function (PACF), and Grid Search algorithms to predict BTC price. To evaluate the performance of the proposed model, Forecast Error (FE), Mean

Computing Classification System 1998: G.2.2
Mathematics Subject Classification 2010: 68R15
Key words and phrases: bitcoin; cryptocurrency; price prediction; autoregression (AR); moving average (MA), ARIMA

[1]Corresponding author

Forecast Error (MFE), Mean Absolute Error (MAE), Mean Squared Error (MSE), as well as Root Mean Squared Error (RMSE), are considered in our study.

1 Introduction

Fiat currencies are currently used to exchange daily payments but the exponential growth of the cryptocurrency market is a phenomenon that has got much attention in recent years. It is a new emerging financial ecosystem, so its opportunities and threats are under evaluation in many academic studies. There are some critical issues that should be analyzed, and the primary question is that whether the price dynamic behavior is predictable or not? Given the efficient market hypothesis (EMH), they have non-deterministic variation patterns, and Bitcoin (BTC) price should be assumed as a stochastic signal. BTC was created and introduced to the world as the first cryptocurrency coin , but since then, many other coins/tokens, so called Altcoins, have generated [12]. Today, about 10,000 coins/tokens are actively traded, and the market capitalization increases noticeably. Many of them have different key features and applications. The various researches were conducted to answer whether BTC is a real currency or not? [43, 42]. Now, between three to six million investors in the private sector including institutions, and individuals (retailers) actively exchange different coins and tokens via well-known available trading networks [27]. In the second quartile (Q2) of 2017, the value of the available cryptocurrencies market, so-called Market Cap, exceeded 91 billion USD [19].

BTC is currently dominant in the market, its position as a strong leader (the king) is vulnerable due to technical issues of the first generation blockchain (G1), security, and the technological advances of new generations of cryptocurrencies (G2 and G3) [33, 13, 21]. Despite the relatively stable economic and gradual growing of interest in cryptocurrencies [3, 40, 14, 38], there is still no comprehensive analysis of cryptocurrency dynamics and ecosystems. In the research field, most of the existing studies focus on Bitcoin itself [34, 30], transaction network, BTC price behavior [18], BTC market trend, BTC dominance, regarding to a limited numbers of fiat currencies (in particular USD) [22, 21]. For example, there is even a disagreement about whether the dominant position of BTC may be compromised or not? Because, BTC dominance plays key role in the whole ecosystem [22]. BTC is the most famous and pioneer coin; however, recent studies on the BTC total market share and other altcoins (in

particular Ethereum (ETH)) indicate that its first rank would be in a real danger in the future.

BTC is an emerging digital currency with the very high volatility. In comparison to the legacy moneys, it leads to more complex challenges in the price prediction problem [11]. A lot of researches have been conducted for the traditional stock markets such as Nasdaq, Japan Exchange Group, New York Stock Exchange, etc. [28, 41], while there are a few studies for cryptocurrencies. It opened a new challenging problem in parallel. BTC price is a time series with a very high volatility and its forecasting problem is still in its early stage of gradual developments. Traditional methods for time series forecasting, such as Holt-Winters exponential models, basically assumes the linear behavior and needs data that can be divided into a trend, seasonal, and noise [15]. On the other hand, there are some approaches to predict BTC price based on Natural Language Processing (NLP) and sentiment analysis. Today, NLP as an AI (artificial intelligence) technology and Deep learning [9, 2] are used together in advanced text mining/analytic tools [23, 4, 26, 8, 7]. These approaches get social media text data from Twitter, Facebook, and etc., as the input and try to draw a link between the content of daily messages and the BTC price. These approaches are on their primal development steps now. The performance of Sentiment analysis are usually restricted to detect just big movements due to some important affecting news. In regular conditions, their performance degrades.

In this research, we aimed to predict the BTC price found on several different algorithms. The Box-Jenkins method, Autoregression, Autocorrelation and Partial Autocorrelation methods, ARIMA method, the Moving Average and Grid search approach are considered. In addition, the Forecast Error (FE), Mean Forecast Error (MFE), Mean Absolute Error (MAE), Mean Squared Error (MSE), as well as Root Mean Squared Error (RMSE), have been used to evaluate our proposed model.

The article is presented as follows: In Section 2, related works are mentioned, while in Section 3, the methodology of the current study is described. Section 4 also presents the simulation results. Finally, Section 5 summarizes and concludes this study.

2 Related works

In [32], the BTC price prediction was introduced using decision tree and regression techniques. The main idea was to get order book data and transfer them

into features over time. These features were referred as feature series which can be used to make prediction models with emphasizing on both volatility and feature series. Their method depends on local data of a special broker/exchanger so the scalability issue should be improved in the future works.

[5] proposed a new approach found on computational intelligence. It used a hybrid controller based on the Neuro-Fuzzy so-called PATIOS to predict the BTC daily price. The results of their work show that their proposed method outperforms the simple neuro-fuzzy approach or simple model of artificial neural networks. The research demonstrated the use of the closed-loop or feedback control technique to expand the BTC and fuzzy modeling literature and uncertainty related to the dynamic behavior of BTC prices to overcome and gain a relatively positive return. PATIOS does not have a user-friendly interface. To enhance it, more focus on creating a UI/UX interface is essential.

In [24], a wide-scale vector model has been proposed to explain how price information was transferred among different crypto market brokers, exchanges, and between traditional markets and crypto ones. Accordingly, they introduced a Vector Autoregressive model (VAR). Their empirical findings suggest that there is a robust correlation between the prices of BTC in different cryptocurrency markets. In contrast, BTC-price correlation with most traditional assets is relatively low. This model can also improve the BTC price forecasts concerning a simple autoregressive model.

In [39], they compared the volatility of one stage and BTC-VaR (BTC value-at-risk (VaR)) forecasting using some important volatility models. It also considered methods that actually involve the presence of outdated data and strongly estimate fluctuations and VaR. The achieved results explicitly suggested that noises and outliers can play an influential role in modeling and forecasting BTC-VaR.

In [29], some graph models had been investigated to analyze BTC price fluctuations. It was found that the optimal model was AR-CGARCH in terms of a good fit with the data. The result indicated well the importance of the existence of both short-term and long-term components for an accurate prediction.

Machine learning (ML) is known as a sub-domain of data science. It can improve software applications to get more accurate results in various applications [6, 35, 36, 25, 10]. It deploys historical perceptions (data) as input to make new predictions. [31] proposed classifier and regression models based on machine learning with high performance for both mid-term and short-term long lines. In their work, the prediction was not restricted to just daily data. They extended the work for monthly and yearly data as well. The classification

model can achieve above 65% accuracy for the next day price prediction and 64% to 62% accuracy for the 7 and 17 days forecasts. For the daily, the error rate is as low as 1.44%, while for the seven to ninety-day horizons, it varies from 2.88% to 4.10%.

To predict BTC price value at different frequencies using machine learning techniques, [17] first classified the price of BTC based on the price of the day and the high frequencies. A set of high- dimensional features include property, gold spot price, network, and trading market were used to predict BTC price on a daily time scale. Statistical methods (SM) include logistic regression (LR) along with linear analysis, were used to predict the daily price prediction with accuracy 66% accuracy, and more complex algorithms pass machine learning (ML). In comparison with results of the criterion for daily price prediction, the results are higher in SM and ML algorithms have 66% and 65.3%better performance, respectively. The machine learning models, such as Support Vector Machine (SVM), XGBoost, Random Forest (RF), and Quadratic Discriminant Analysis (QDA), were superior to SM for the 5 minutes price prediction (5m) with an accuracy of 67.2%.

In the next section, we are going to discuss about methodology and the proposed model.

3 Methodology

In this section, our proposed model is proposed to provide a detailed forecast of price. For this purpose, first, the datasets used are introduced, and then the performance evaluation criteria and our proposed forecasting models such as Box-Jenkins method, ACF, PACF, ARIMA, MA, AR, and Grid Search are investigated.

3.1 Dataset

We have used the BTC price dataset from [44]. It provides price values from 1 December 2014 to 29 may 2020. The data is available in the daily time scale. The statistical summary of the dataset is presented in Table 1.

count	mean	std	min	25%	50%	75%	max
1994	4213.51	4005.48	120.00	446.51	3425.41	7459.77	19650.00

Table 1: The statistical summary of the BTC price datasets from 1 December 2014 to 29 May 2020.

3.2 Performance evaluation criteria

The performance evaluation criteria of regression algorithms for time series forecasting problem are introduced as follows.

3.2.1 Forecast error (residual forecast error)

Forecast errors (FE) on a time series forecasting problem are considered as residuals (residual errors). According to Equation 1, the residual error at time $t(e^t)$ can be calculated as the expected outcome (y_t) minus the forecast ($f_t^{(m)}$):

$$e^t = (y_t - f_t^{(m)}) \tag{1}$$

e^t can be calculated for every observation of the time series. The much more closer the residual error to the zero, the better performance archives.

3.2.2 Mean forecast error (forecast bias)

This value is obtained from the average error residual.

$$BIAS = \frac{1}{n}\Sigma_{(i=1)}^{n}(e_i) \tag{2}$$

Where n is the number of samples and e_i is the ith error. The residual error is either positive or negative. The best BIAS value would be zero.

3.2.3 Mean absolute error (MAE)

One of the loss functions that have interesting properties is the mean absolute error which is also called the MAE. The loss function, such as the MSE, uses only the distance between predicted and expected values but does not consider the direction for this difference. The following formula is used to calculate MAE [20]:

$$MAE = \frac{1}{n}\Sigma_{(i=1)}^{n}|e_i| \tag{3}$$

When MAE is closer to zero; the ideal model is achieved.

3.2.4 Mean squared error (MSE)

In mean square errors or MSE, the values are positive. It also has a more significant impact on large errors. The loss function computes the mean square error between the predicted and expected values. The Equation 4 shows MSE formula [20]:

$$MSE = \frac{1}{n}\Sigma_{(i=1)}^{n}(e_i^2) \tag{4}$$

3.2.5 Root mean squared error (RMSE)

If the effect of the MSE is derived, another loss function is constructed as the "square root of error" which is briefly shown with RMSE.

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{(i=1)}^{n}(e_i^2)} \tag{5}$$

RMSE zero value means the model is actually error-free.

Where N is the total number of observations, (e_i^2) is the actual value.

3.3 Forecast models

Figure 1 shows the steps of the current research in which the Box-Jenkins method, ACF, PACF, ARIMA, MA, AR, and Grid Search are used.

3.3.1 Box-Jenkins method

The Box-Jenkins method consists of three essential steps. The first step is Model Identification/selection; the second one is Parameter estimation and the last one Statistical model checking. In the identification phase, one or more time series models will be selected by their graphs. The parameters of the selected model are determined in the identification phase. In the control phase (verification), statistical tests are performed to verify the selected model. These tests include the independence of model error values. If the control phase is not approved, then back to the identification phase and selects a new model, the above steps are repeated.

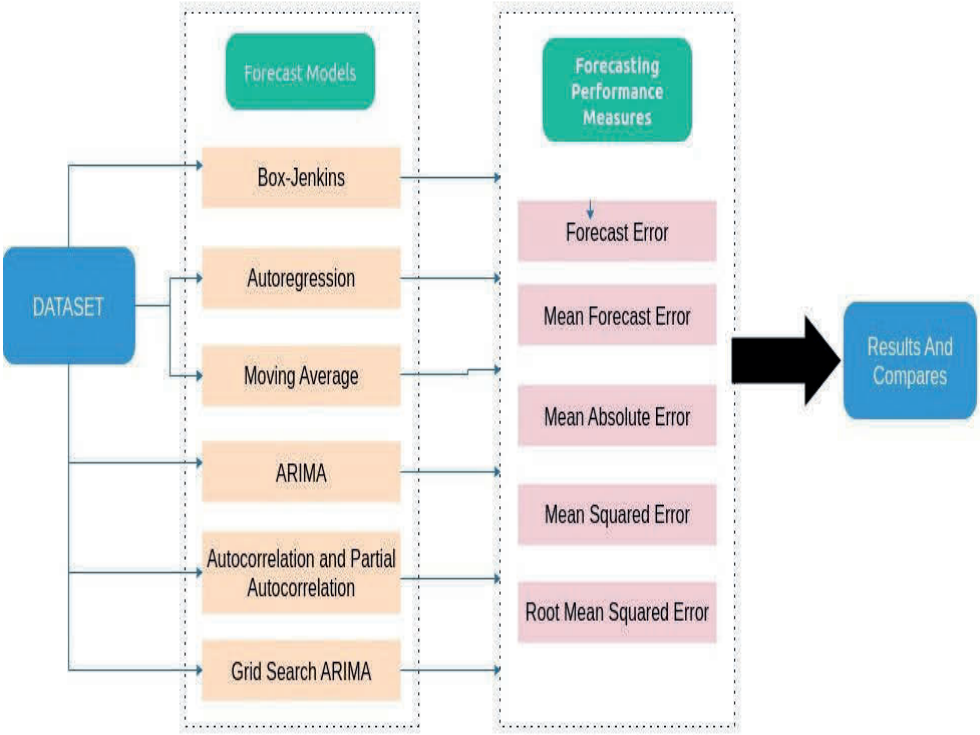


Figure 1: The workflow steps in our proposed model

3.3.2 Autoregression method

The Autoregressive (AR) model is used for stationary time series values depending on their previous values. In this case, we consider the number of past observations to predict a value. Therefore, it can be written as follows [37]:

$$x_t = c + \Sigma_{(i=1)}^{P} \phi_i x_{(t-i)} + \epsilon_t \tag{6}$$

Where x_t is the stationary variable, c is constant, the terms in ϕ_i are autocorrelation coefficients at lags $1, 2, p$ and ϵ_t.

3.3.3 Moving average algorithm

The moving average (MA) model is written as a linear combination, similar to the AR model, but it is written in terms of a linear combination of errors in

terms of time. In this case, we consider q the number of previous observations. The relationship between static time-series values in this model is formulated in Equation 7 [37]:

$$x_t = \mu + \Sigma_{(i=0)}^{q} \phi_i \epsilon_t - i \tag{7}$$

Where μ is the expectation of x_t (usually assumed equal to zero), the ϕ_i terms are the weights applied to the current Where μ is the expectation of x_t (usually assumed equal to zero), the ϕ_i terms are the weights applied to the current.

3.3.4 ARIMA method

The autoregressive moving average is an ARIMA (Autoregressive Integrated Moving Average) model, which is a more general model of ARMA. According to the Equation 8, ARIMA combines the combination of two methods of AR and MA [37].

$$x_t = c + \Sigma_{(i=1)}^{p} \phi_i x_{(t-i)} + \epsilon_t + \Sigma_{(i=0)}^{q} \phi_i \epsilon_t - i \tag{8}$$

An ARIMA model has coordinates (p, d, q):

- The p is defined as the total number of autoregressive terms. In other words, p is the number of previous observations (from past values) that are used to predict future values. For example, if the value p is equal to 2, this means that two previous temporal observations are used in the series to do forecast the future trends.

- d points to the number of differences which are required to have a stationary the time series (i.e. one with a constant mean, variance, and autocorrelation). For example, if d =1, then the first difference of the time series must be calculated to convert it to the stationary series should be calculated represents the "moving average" error in the previous prediction error in the model or the residual value.

- q represents the MA of the previous FE in the model (the lagged values of the error term). For example, if q has a high value, there is a lagged value of the error term in the model.

3.3.5 Autocorrelation and partial autocorrelation methods

Correlation and partial correlation diagrams are enormously used in time series analysis and forecasting. These are diagrams that summarize the strength of the relationship with observations in a time series with observations at the previous stages. Partial Autocorrelation and Autocorrelation index are used to determine whether the data are stationary or not? The autocorrelation and autocorrelation index for "different degrees" measures the correlation coefficient (CC) between the series and the delay of variables over time. A process is achieved when the time series follows a particular pattern in which the present value depends on the previous values.

3.3.6 Grid search method

Grid search (GS) is a nifty approach that tries to explore data space exhaustively using a manually specified hyperparameter subset of the search space for a selected algorithm while Random search choses values for all hyperparameters independently based on their probability distributions. Accurate tuning (Fine Tuning) means finding the best parameter for machine learning algorithms to improve the results. An optimized planning is an effective practical step that can lead to noticeable improvements at the output of the ARIMA method. The optimal parameters can be automatically found on the Grid Search.

4 Experimental results

As noted, the data set includes daily BTC close price from 1 December 2014 to 29 May 2020 [44]. In the simulation, we have used 70% of data as training data and 30% as the test data. We have used Python programing language and related libraries for time series to implement our simulations.

In time series, it is essential to evaluate and compare the results to spot the best predictor with a minimum error. In our study, the results obtained are compared using BIAS, MAE, MSE, and RMSE. In the term of error, a zero error indicates complete skill for prediction. In Tables 2 and 3, the obtained results are compared well. Figures 2 to 5 are derived from Table 2 and demonstrate the results separately for each algorithm.

In Figures 6, 7, 8, and 9, autocorrelation functions (ACF) and partial autocorrelation (PACF) are calculated for the AR, MA, ARIMA, and Box Jenkins algorithms. We know these functions reveal the logical link between data in

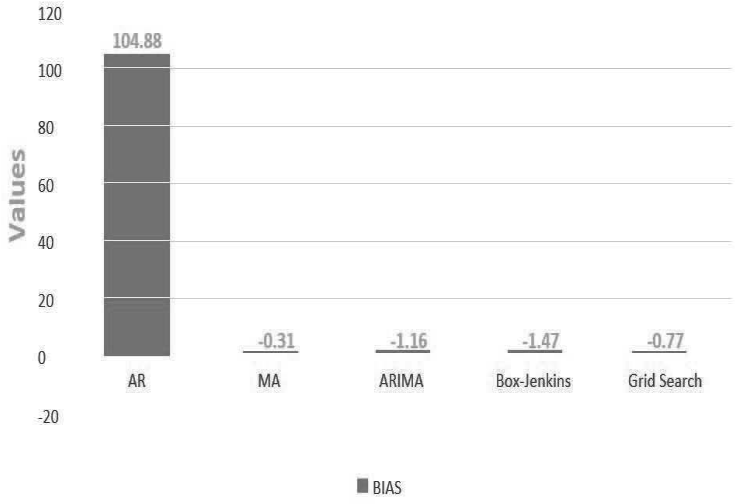


Figure 2: Bias comparison for all algorithms

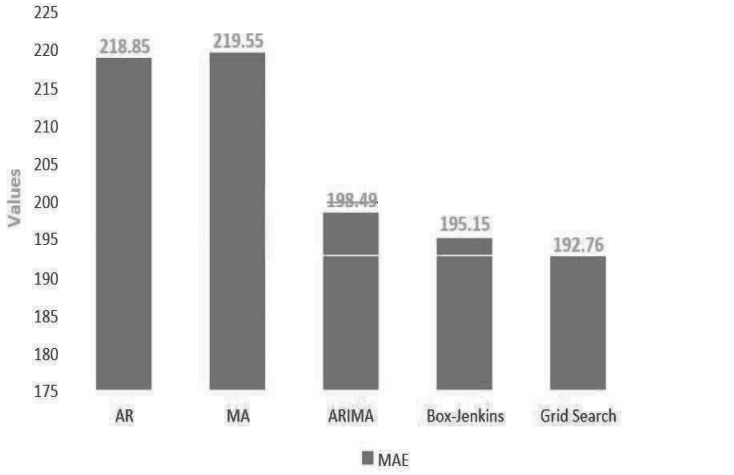


Figure 3: MAE comparison for all algorithms

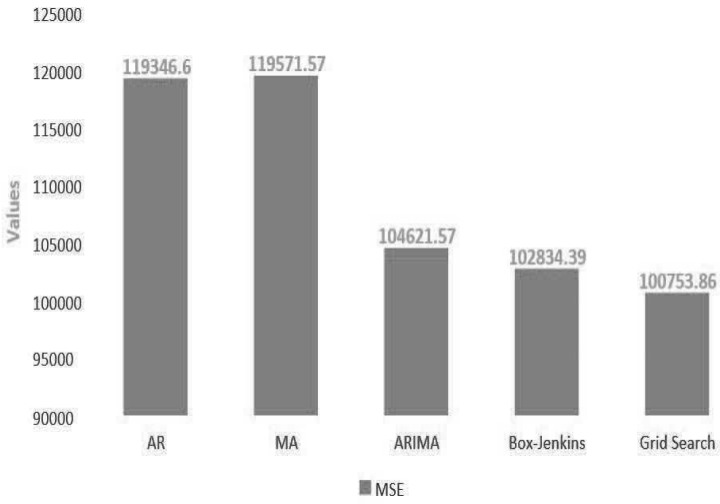


Figure 4: MSE comparison for all algorithms

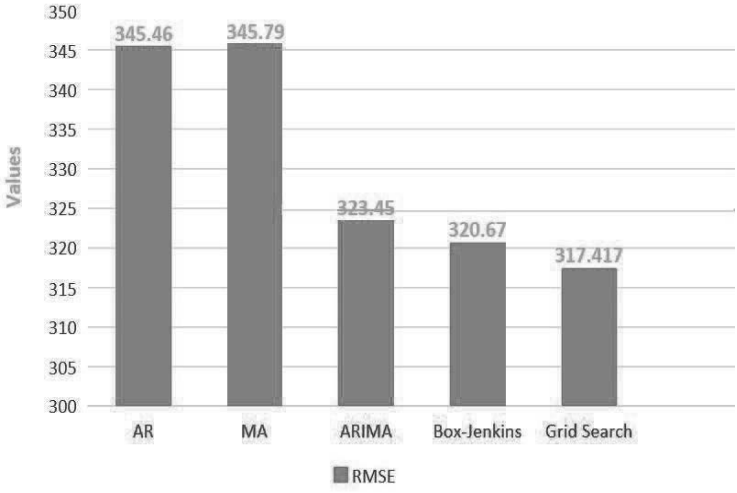


Figure 5: RMSE comparison for all algorithms

Forecast methods	RMSE	MSE	MAE	MFE
AR	345.46	119346.60	218.85	104.88
MA	345.79	119571.57	219.55	-0.31
ARIMA	323.45	104621.57	198.49	- 1.16
Box-Jenkins	320.67	102834.39	95.15	-1.47
Grid Search	317.417	100753.86	192.76	-0.77

Table 2: Different evaluation methods for various prediction algorithms.

AR	-425.71 , 148.50 , -112.42 , ... 315.33, 396.05, 89.32
MA	-435.46, 153.67 , -124.88 , ... 294.43 , 388.70 , 67.43
ARIMA	-351.55, 108.85, -191.76, ... 347.01, 362.92, -116.02
Box- Jenkins	-345.70 , 129.47 , -150.95 , ... 373.48, 381.12, -147.27
Grid Search	Best ARIMA(0, 2, 0), FE=-527.228

Table 3: Assessment of forecasting algorithms with the Forecast Error (FE).

time series. We first extracted prediction data based on the mentioned algorithms, and then we applied ACF and PACF. In Figure 6(a), an ACF plot demonstrates correlation coefficients (CC) bar chart of a time series and lagged values for the AR model. In Figure 6(b), a PACF plot indicates the partial correlation between the series and lags of itself. For the AR process, it can be seen that the ACF plot decreases gradually while simultaneously the PACF has a severe drop after p significant lags.

Figure 7 show ACF and PACF for a MA process. In this figure, the ACF fall down sharply after a q number of lags while PACF follows a gradual declining pattern.

As mentioned before, a model that can get lower errors represents the best model. In our research the Grid Search method is the only algorithm that led to the optimal values in most performance metrics such as MAE equal to 192.76, RMSE = 317.417, and MSE equal to 100753.86.

Ultimately, we compare our model with some available methods in Table 4 and Table 5. Table 4 indicates that our model outperforms others in term of MSE while in Table 5, it shows better achievement based on RMSE and MAE.

Performance metric	The proposed	[20]	[1]	[16]
MSE	104621.57	170962.195	21,215,311	2519603.08

Table 4: Comparing BTC price prediction algorithms based on the MSE.

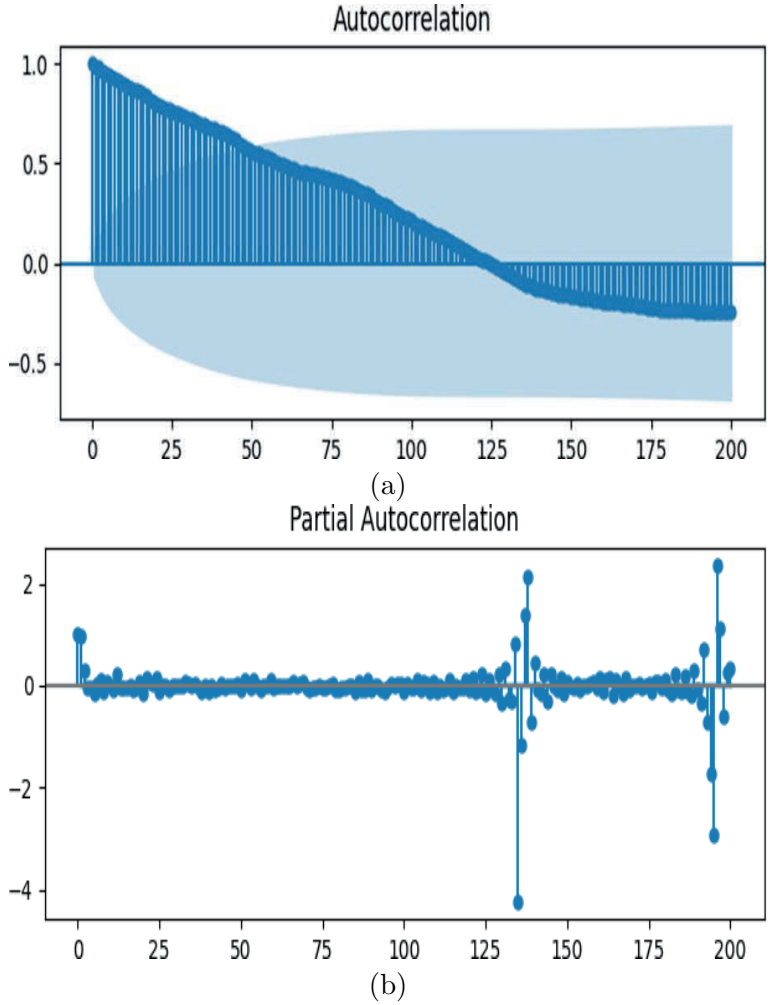
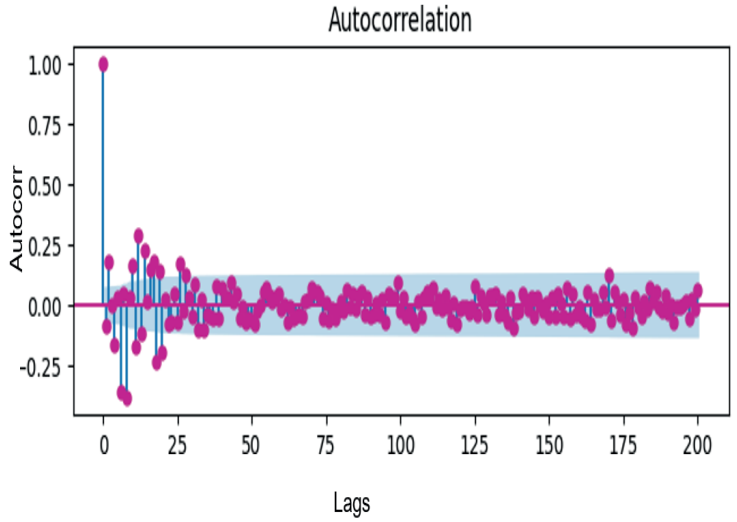
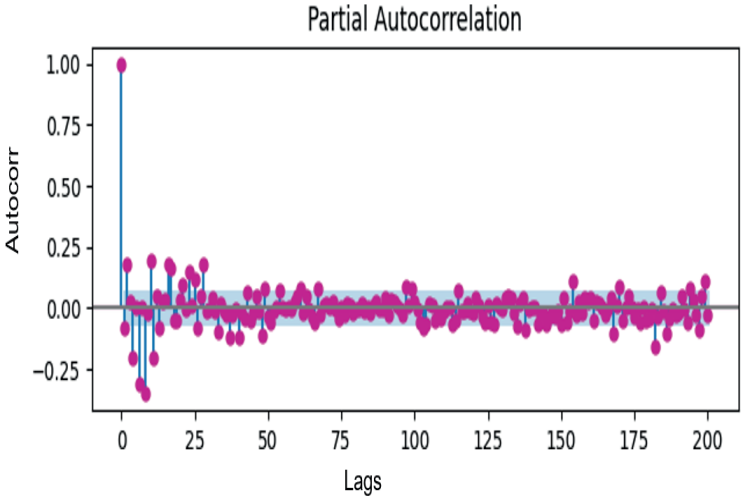


Figure 6: Results of the ACF and PACF methods for the predicted values by the AR method.
(a) ACF plots of the Prediction from AR models
(b) PACF plots of the Prediction from AR models

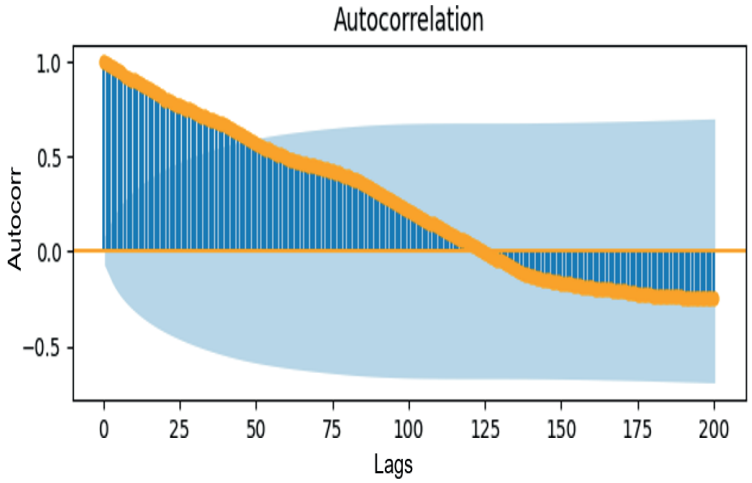


(a)

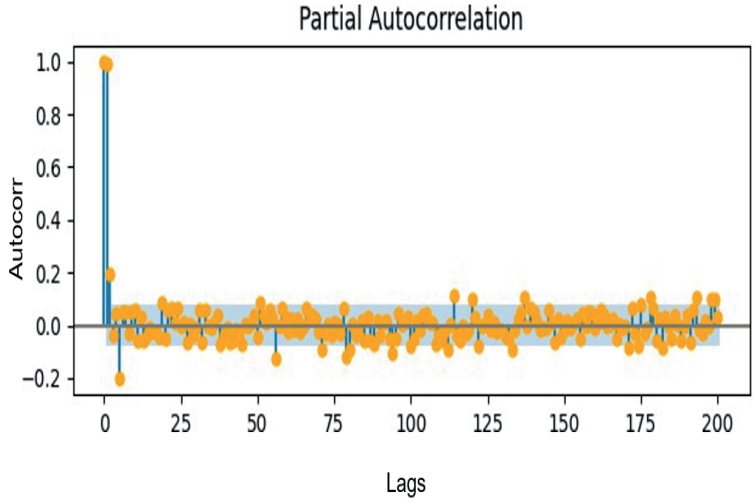


(b)

Figure 7: Results of the ACF and PACF methods for the predicted values by the MA method.
(a) ACF plots of the Prediction from MA models
(b) PACF plots of the Prediction from MA models

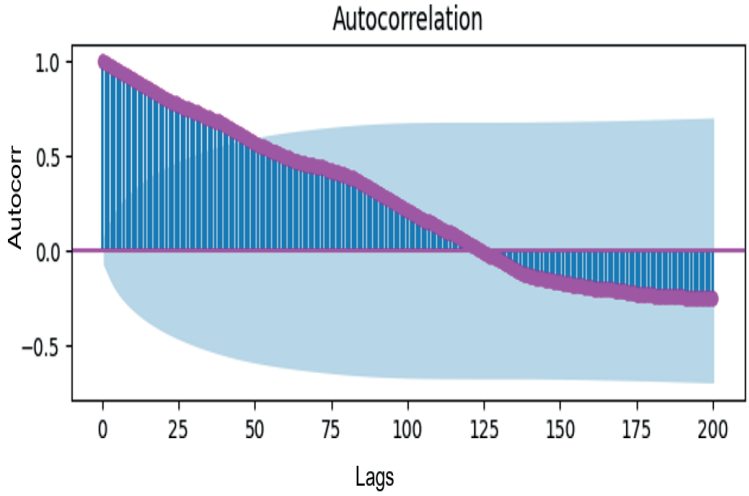


(a)

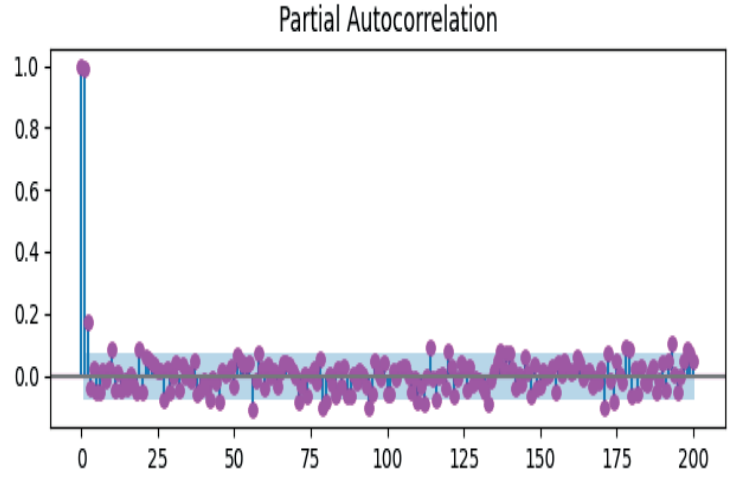


(b)

Figure 8: Results of the ACF and PACF method for ACF plots of the Prediction from ARIMA.
(a) ACF plots of the Prediction from ARIMA models
(b) PACF plots of the Prediction from ARIMA models



(a)



(b)

Figure 9: Results of the ACF and PACF method results for predicted values predicted by the Box-Jenkins method.
(a) ACF plots (b) PACF plots

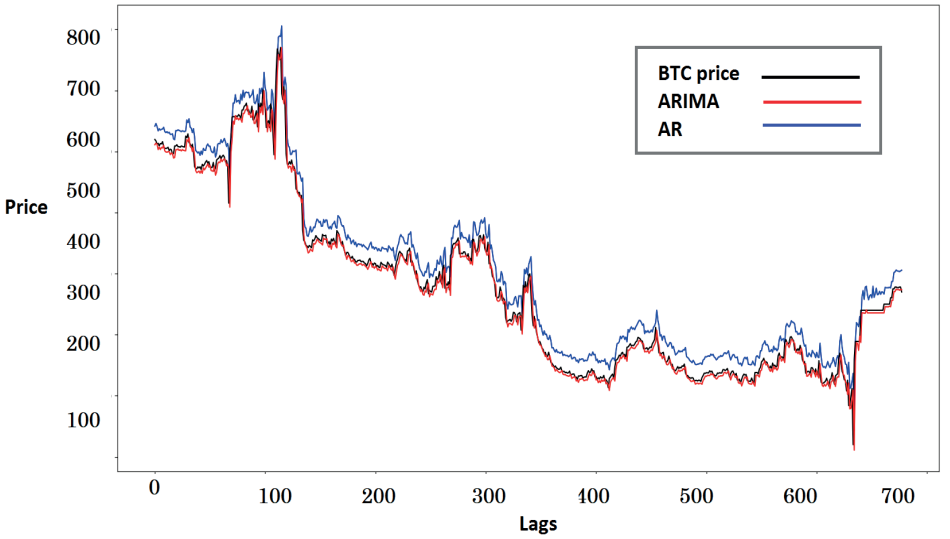


Figure 10: Comparison of ARIMA and AR algorithms with the current value of bitcoin price from 1 December 2014 to 29 May 2020 (Current price of Bitcoin (black), ARIMA (red), AR (blue)).

Performance metric	The proposed	[16]
RMSE	323.45	1587.32
MAE	198.49	920.45

Table 5: Comparison of BTC price prediction models using the RMSE and MAE.

In Figure 10, a comparison among real price of bitcoin and two prediction models is depicted. In this figure, our proposed algorithm based on grid search ARIMA (red line) greedily tries to track the real BTC price (black line) and it outperforms other models based on AR.

Finally, we have extended our dataset to cover forward test. In the early version, the BTC price datasets from 1 December 2014 to 29 May 2020 was used in training and validation steps. In the new version, we added some new daily price samples from 1 June to 30 June (2020) as unseen data. The unseen data has fitted to the model in order to evaluate the forward testing results.

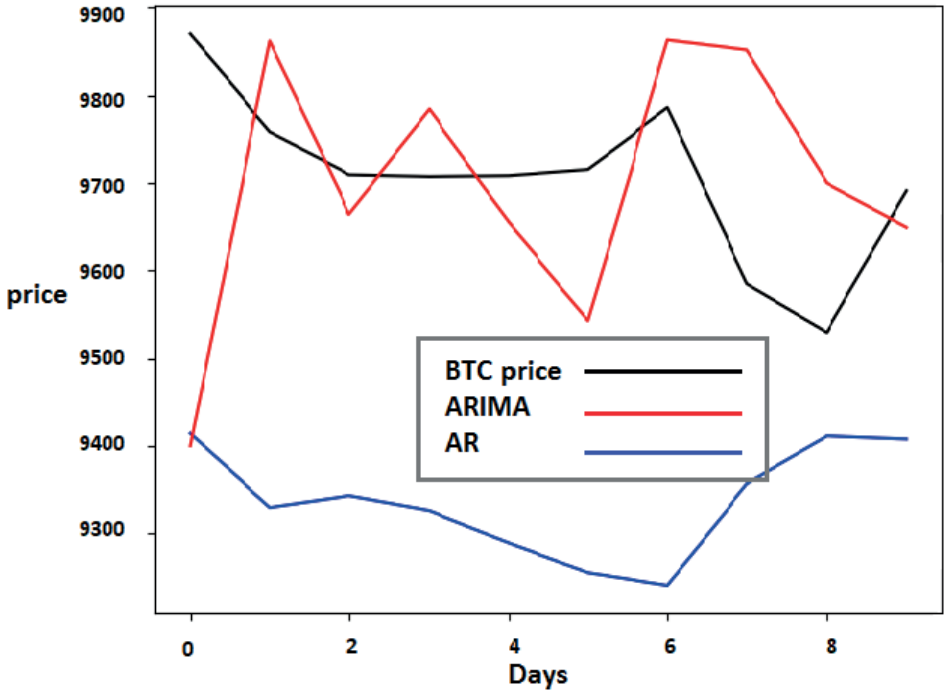


Figure 11: Comparison of ARIMA and AR algorithms with the current value of bitcoin price for June 2020 (Current price of Bitcoin (black), ARIMA (red), AR (blue)).

In Figure 11, it can be seen that the accuracy of our algorithm is much higher even for unseen data.

5 Conclusions and future work

As an emerging digital currency, Bitcoin has got much attention nowadays. In fact, BTC is the most valuable encrypted currency globally traded in almost all cryptocurrency exchanges. It provides a fantastic opportunity to make price prediction due to its relatively low-maturity technology with very high price volatility. Developing an accurate and automated predictive system for BTC with non-linear and high range variations is still an open challenging task.

In our study, we have evaluated the number of algorithms using several evaluation criteria. In summary, we have used the Box-Jenkins model, Au-

toregression, Moving Average, ARIMA model, Autocorrelation and Partial Autocorrelation model, and Grid Search model. Besides, to evaluate and compare them, we used FE, MFE, MAE, MSE, and RMSE. In our study, the Grid Search method has the best performance with lower errors than other methods. This method tries to find the best results with optimizing the ARIMA method. It achieved the minimum value in the MAE at 192.76, the MSE equal to 100753.86, and RMSE = 317.417. Furthermore, the values of the forecast error method in the Grid Search have the lowest values approximately close to zero. Consequently, the proposed model outperformed others in most performance metrics.

As both the test and train data are scattered randomly in the entire time-interval, the problem actually would be one of price interpolation or filling the gaps problems (not extrapolation). The main contribution is a thorough comparison of the performance of existing methods and also optimizing models based on Grid search optimization. The Grid searching is the process of scanning the data to configure optimal parameters for a given model. Based on the Grid search, the optimal hyperparameters of our model have tunned. As a result, the more accurate predictions have achieved in our study.

Price volatility has been extensively investigated on financial markets, but due to the recent emergence of Bitcoin market, researchers have started to scratch the surface in this area. Hence, the excessive volatility of Bitcoin and how determine it properly has not yet been sufficiently studied providing for an extensive research gap. Consequently, similar to the most available studies in BTC price prediction, we have not aimed to offer a discussion into Bitcoin price volatility but it may be our new study in the future to solve this issue using deep learning algorithms. Today, deep learning algorithms are deployed widely in various applications [23, 9, 2]. To continue this research, more studies can be done on the non-linear and even non-stationary BTC price time series in lower time scales using deep learning models.

Funding Statement. There is no specific funding for this study.
Conflicts of Interest. The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] G. Abdoli, Comparing the prediction accuracy of LSTM and ARIMA models for time-series with permanent fluctuation, *Periódico do Núcleo de Estudos e Pesquisas sobre Gênero e DireitovCentro de Ciências Jurídicas-Universidade Federal da Paraíba*, vol. 9, 2020. ⇒277

[2] F. Abedini, M. Bahaghighat, M. S'hoyan, Wind turbine tower detection using feature descriptors and deep learning, *Facta Universitatis, Series: Electronics and Energetics*, **33,** 1 (2019) 133–153. ⇒267, 284

[3] R. Ali, J. Barrdear, R. Clews, J. Southgate, Innovations in payment technologies and the emergence of digital currencies, *Bank of England Quarterly Bulletin*, Q3, 2014. ⇒266

[4] E. Amouee, M. M. Zanjireh, M. Bahaghighat, M. Ghorbani, A new anomalous text detection approach using unsupervised methods, *Facta universitatis-series: Electronics and Energetics*, **33,** 4 (2020) 631–653. ⇒267

[5] G. S. Atsalakis, I. G. Atsalaki, F. Pasiouras, C. Zopounidis, Bitcoin price forecasting with neuro-fuzzy techniques, *European Journal of Operational Research*, **276,** 2 (2019) 770–780. ⇒268

[6] M. K. Bahaghighat, R. Akbari *et al.*, Fingerprint image enhancement using GWT and DMF, in *2010 2nd International Conference on Signal Processing Systems*, vol. 1. IEEE, 2010, pp. V1-253–257. ⇒268

[7] M. K. Bahaghighat, J. Mohammadi, Novel approach for baseline detection and Text line segmentation, *International Journal of Computer Applications*, **51,** 2 (2012) 9–16. ⇒267

[8] M. K. Bahaghighat, F. Sahba, E. Tehrani, Textdependent Speaker Recognition by Combination of LBG VQ and DTW for Persian language. *International Journal of Computer Applications*, **51,** 16 (2012) 23–27. ⇒267

[9] M. Bahaghighat, Q. Xin, S. A. Motamedi, M. M. Zanjireh, A. Vacavant, Estimation of wind turbine angular velocity remotely found on video mining and convolutional neural network, *Applied Sciences*, **10,** 10 (2020) 3544. ⇒267, 284

[10] M. Bahaghighat, F. Abedini, Q. Xin, M. M. Zanjireh, S. Mirjalili, Using machine learning and computer vision to estimate the angular velocity of wind turbines in smart grids remotely, *Energy Reports*, 2021. ⇒268

[11] M. Briere, K. Oosterlinck, A. Szafarz, Virtual currency, tangible return: Portfolio diversification with bitcoin, *Journal of Asset Management*, **16,** 6 (2015) 365–373. ⇒267

[12] G. M. Caporale, L. Gil-Alana, A. Plastun, Persistence in the cryptocurrency market, *Research in International Business and Finance*, **46** (2018) 141–148. ⇒266

[13] G. Caffyn, What is the bitcoin block size debate and why does it matter, http://www.coindesk.com/, 2015. ⇒266

[14] M. J. Casey, P. Vigna, Bitcoin and the digital-currency revolution, *The Wall Street Journal*, Jan. 23, 2015. ⇒266

[15] C. Chatfield, M. Yar, Holt-Winters forecasting: some practical issues, *Journal of the Royal Statistical Society: Series D (The Statistician)*, **37,** 2 (1988) 129–140. ⇒267

[16] A. Chaudhari, *Forecasting Cryptocurrency Prices using Machine Learning*, 2020, Dublin, National College of Ireland, Ph.D. dissertation. ⇒277, 282

[17] Z. Chen, C. Li, W. Sun, Bitcoin price prediction using machine learning: An approach to sample dimension engineering, *Journal of Computational and Applied Mathematics*, **365** (2020) p. 112395. ⇒269

[18] P. Ciaian, M. Rajcaniova, D. Kancs, The economics of BitCoin price formation, *Applied Economics*, **48,** 19 (20160) 1799–1815. ⇒266

[19] J. Debler, Foreign initial coin offering issuers beware: the Securities and Exchange Commission is watching, *Cornell Int'l LJ*, **51** (2018) 245–245. ⇒266

[20] J. Fiaidhi, A. Sabah, M. A. Ansari, Z. Ayaz, Bitcoin Price Prediction using ARIMA Model, 2020. ⇒270, 271, 277

[21] N. Gandal, H. Halaburda, Competition in the cryptocurrency market, 2014. ⇒266

[22] N. Gandal, H. Halaburda, Can we predict the winner in a market with network effects? Competition in cryptocurrency market, *Games*, **7,** 3 (2016) 16. ⇒266

[23] M. Ghorbani, M. Bahaghighat, Q. Xin, F. Özen, ConvLSTMConv network: a deep learning approach for sentiment analysis in cloud computing, *Journal of Cloud Computing*, **9,** 1 (2020) 1–12. ⇒267, 284

[24] P. Giudici, I. Abu-Hashish, What determines bitcoin exchange prices? A network VAR approach, *Finance Research Letters*, **28** (2019) 309–318. ⇒268

[25] A. Hajikarimi, M. Bahaghighat, Optimum Outlier Detection in Internet of Things Industries Using Autoencoder, in *Frontiers in Nature-Inspired Industrial Optimization*. Springer, 2022, pp. 77–92. ⇒268

[26] S. Hasani, M. Bahaghighat, M. Mirfatahia, The mediating effect of the brand on the relationship between social network marketing and consumer behavior, *Acta Technica Napocensis*, **60,** 2 (2019) 1–6. ⇒267

[27] G. Hileman, M. Rauchs, Global cryptocurrency benchmarking study, *Cambridge Centre for Alternative Finance*, **33** (2017) 33–113. ⇒266

[28] I. Kaastra, M. Boyd, Designing a neural network for forecasting financial and economic time series, *Neurocomputing*, **10,** 3 (1996) 215–236. ⇒267

[29] P. Katsiampa, Volatility estimation for Bitcoin: A comparison of GARCH models, *Economics Letters*, **158** (2017) 3–6. ⇒268

[30] M. Lischke, B. Fabian, Analyzing the bitcoin network: The first four years, *Future Internet*, **8,** 1 (2016) 7. ⇒266

[31] M. Mudassir, S. Bennbaia, D. Unal, M. Hammoudeh, Time-series forecasting of Bitcoin prices using high-dimensional features: a machine learning approach, *Neural Computing and Applications*, pp. 1–15, 2020. ⇒268

[32] K. Rathan, S. V. Sai, T. S. Manikanta, Crypto-currency price prediction using decision tree and regression techniques, in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*. IEEE, 2019, pp. 190–194. ⇒267

[33] F. Reid, M. Harrigan, An analysis of anonymity in the bitcoin system, in *Security and privacy in social networks*. Springer, 2013, pp. 197–223. ⇒266
[34] D. Ron, A. Shamir, Quantitative analysis of the full bitcoin transaction graph, in *International Conference on Financial Cryptography and Data Security*. Springer, 2013, pp. 6–24. ⇒266
[35] M. S. S. Sajadi, M. Babaie, M. Bahaghighat, Design and implementation of fuzzy supervisor controller on optimized DC machine driver, in *2018 8th Conference of AI & Robotics and 10th RoboCup Iranopen International Symposium (IRANOPEN)*. IEEE, 2018, pp. 26–31. ⇒268
[36] A. Shamseen, M. M. Zanjireh, M. Bahaghighat, Q. Xin, Developing a parallel classifier for mining in big data sets, *IIUM Engineering Journal*, **22,** 2 (2021) 119–134. ⇒268
[37] S. Siami-Namini, N. Tavakoli, A. S. Namin, A comparison of ARIMA and LSTM in forecasting time series, pp. 1394–1401, 2018. ⇒272, 273
[38] S. Trimborn, W. K. Härdle, CRIX an Index for blockchain based Currencies, *Journal of Empirical Finance*, **49** (2018) 107–222. ⇒266
[39] C. Trucíos, Forecasting Bitcoin risk measures: A robust approach, *International Journal of Forecasting*, **35,** 3 (2019) 836–847. ⇒268
[40] P. Vigna, M. J. Casey, *The age of cryptocurrency: how bitcoin and the blockchain are challenging the global economic order*. Macmillan, 2016. ⇒266
[41] H. White, Economic prediction using neural networks: The case of IBM daily stock returns, in *ICNN*, vol. 2, 1988, pp. 451–458. ⇒267
[42] L. H. White, The market for cryptocurrencies, *Cato J.* **35** (2015) 383. ⇒266
[43] D. Yermack, Is Bitcoin a real currency? An economic appraisal (No. w19747), *National Bureau of Economic Research*, **36,** 2 (2013) 843–850, ⇒266
[44] ∗∗∗, Historical OHLC price data includes volume, www.cryptodatadownload.com ⇒269, 274

Received: July 16, 2021 • Revised: October 25, 2021

Acta Univ. Sapientiae, Informatica **13,** 2 (2021) 288–302

DOI: 10.2478/ausi-2021-0013

Trading sparse, mean reverting portfolios using VAR(1) and LSTM prediction

Attila RÁCZ
Budapest University of Technology and Economics
Department of Networked Systems and Services
email: racz@hit.bme.hu

Norbert FOGARASI
Budapest University of Technology and Economics
Department of Networked Systems and Services
email: fogarasi@hit.bme.hu

Abstract. We investigated the predictability of mean reverting portfolios and the VAR(1) model in several aspects. First, we checked the dependency of the accuracy of VAR(1) model on different data types including the original data itself, the return of prices, the natural logarithm of stock and on the log return. Then we compared the accuracy of predictions of mean reverting portfolios coming from VAR(1) with different generative models such as VAR(1) and LSTM for both online and offline data. It was eventually shown that the LSTM predicts much better than the VAR(1) model. The conclusion is that the VAR(1) assumption works well in selecting the mean reverting portfolio, however, LSTM is a better choice for prediction. With the combined model a strategy with positive trading mean profit was successfully developed. We found that online LSTM outperforms all VAR(1) predictions and results in a positive expected profit when used in a simple trading algorithm.

1 Introduction

The usefulness of mean reverting portfolios was discussed several times before [2], [1], [3], [4]. Maximizing the predictability helps to create simple, effective

Computing Classification System 1998: G.1.6.
Mathematics Subject Classification 2010: 91B60, 91B84
Key words and phrases: portfolio selection, optimization, mean reversion, time-series prediction, LSTM

and robust trading strategies. Our previous work based on calibrating the VAR(1) generative model produced acceptable results on the S&P500 stock prices [10]. In the first part of this work we investigate how accurate the VAR(1) model is on different data models. Data models could mean the representation of the data, ie. the raw stock data itself or its transformation. Using a well performed data model we generate sparse portfolios with high predictability by searching the maximal generalized eigenvalue of the regression matrix. The second part of this work focuses to the prediction of the already created portfolio. The first strategy is to utilize the calibrated regression matrix of the VAR(1) model, the second is to train an LSTM neural network on the portfolio and use it for prediction [13], [11]. Both methods were tested for online and offline data. Prediction on online data means we use only the available real time data for future values while offline means we incorporate the previously predicted values to regress the next portfolio value. The latter makes it possible to predict accurately for a longer term. The structure of the paper is the following:

- In section 2, we briefly discuss the concept of VAR(1) model and mean reversion. Then we make a comparison of the accuracy of the VAR(1) model on the different data models.
- In section 3, we discuss the concept of LSTM neural network and how useful it is on mean reverting time series.
- In section 4, we discuss the details of the concept of online and offline prediction.
- In section 5, we compare the performance of the different techniques.
- In section 6, we make conclusions and recommendations.

2 VAR(1) model on derived time series

This section briefly explains the mean reverting processes and the modeling of the stock data with VAR(1). We call a stochastic process mean reverting when the value of the process oscillates around its average value. When the price is below its long-term mean it will likely increase rather than decrease and vice versa. This supports building a simple trading strategy and to estimate the trading range for the portfolio.

2.1 Ornstein-Uhlenbeck process

Mean reverting processes are formalized by the so-called Ornstein-Uhlenbeck process [8]. Let's denote the price of our portfolio by p_t at time t, and by s_t^i the price of the i^{th} stock at time t. Our mean reverting portfolio p_t is composed by the linear combination of s_t^i's. The stochastic differential equation that drives the Ornstein-Uhlenbeck process is

$$dp_t = \lambda(\mu - p_t)\,dt + \sigma dW_t \tag{1}$$

where W_t is a Wiener process, σ is a parameter proportional to the standard deviation of the Wiener process, λ is the speed of $\mathtt{mean\ reversion}$ and μ is the long-term mean of the process reverting to. The deterministic part of the stochastic differential equation (SDE) represents the property and that the magnitude of attraction to the long-term mean is proportional to the distance from the mean. The solution of stochastic differential equation is:

$$p(t) = p(0)\,e^{-\lambda t} + \mu\left(1 - e^{-\lambda t}\right) + \int_0^t \sigma e^{-\lambda(t-s)} dW(s) \tag{2}$$

The expected value of equation (2) is

$$\mathbb{E}[p(t)] = p(0)\,e^{-\lambda t} + \mu\left(1 - e^{-\lambda t}\right) \tag{3}$$

and the variance is

$$\mathbb{V}[p(t)] = \sigma^2 \int_0^t e^{-2\lambda(t-s)} ds = \sigma^2 \frac{1 - e^{-2\lambda t}}{2\lambda}. \tag{4}$$

Consequently in very long-term, the expectation converges to

$$\lim_{t\to\infty} \mathbb{E}[p(t)] = \mu, \tag{5}$$

while the variance is

$$\lim_{t\to\infty} \mathbb{V}[p(t)] = \frac{\sigma^2}{2\lambda}, \tag{6}$$

Note the variance is inversely proportional to the speed of mean reversion.

2.2 Asset dynamics and portfolio selection

2.2.1 Modeling asset dynamics with VAR(1)

In [2], [3] the concept of predictability was introduced in the following way:

$$\nu = \frac{\sigma_{t-1}^2}{\sigma_t^2} \tag{7}$$

where σ_t^2 is the variance of the time series. If the denominator is larger in (7), S_t will be pure noise as t goes to infinity therefore the time series is completely unpredictable, however while the nominator is larger as t going forward S_t will be perfectly predictable. Let the dynamics of the assets be modeled as discrete vector autoregressive process with parameter 1. Generally, VAR(p) means that the regression uses the last p values in the time series. As mentioned before $s_{i,t}$ denotes the price of the stock i at time t where $i = 1, \ldots, n$, where n is the size of the set of stocks. The most general model used in article 8 is the non-stationary VAR(1) model that contains a time independent constant scalar shift term to describe drift or ie. to ensure positivity of the elements for all t:

$$s_{t+1} = c + As_t + W_t, \tag{8}$$

where A is an n by n real matrix constant at some certain time period, c is a time independent real scalar constant, W_t represents the noise or error term of the model with zero mean value, some constant variance and uncorrelated across time. This can be rewritten in a concise VAR(1) notation by incorporating shift into the matrix of auto regression:

$$s'_{t+1} = A's'_t + W'_t \tag{9}$$

extending the notations

$$\begin{bmatrix} s'^1_{t+1} \\ \vdots \\ s'^n_{t+1} \\ 1 \end{bmatrix} = \begin{bmatrix} a_{1,1} & \cdots & a_{1,n} & c_1 \\ \vdots & \ddots & \vdots & \\ a_{n,1} & \cdots & a_{n,n} & c_n \\ 0 & \cdots & 0 & 1 \end{bmatrix} \begin{bmatrix} s'^1_t \\ \vdots \\ s'^n_t \\ 1 \end{bmatrix} + \begin{bmatrix} W^1_t \\ \vdots \\ W^n_t \\ 0 \end{bmatrix} \tag{10}$$

where A' refers to a $(n+1) \times (n+1)$ matrix in which the last column is filled with the constant shift c, the last row has zeros except the element of $(n+1)^{st}$ which should be strictly 1, x'_t a vector with $n+1$ elements strictly 1 at the $(n+1)^{st}$ element and W'_t still provides the noise as in the previous case except no noise for the $(n+1)^{st}$ element [7]. Henceforth, we ignore the prime sign in this article. The auto regression matrix in equation (8) can be approximated using least squares regression by

$$\hat{A} = \left(s_{t-1}^T s_{t-1}\right)^{-1} s_{t-1}^T s_t \tag{11}$$

Using this model a portfolio can be created with a linear combinations of the assets. To make this, let P be a real valued vector. This vector represents the

weights that relate to stocks. The time evolution of the value of our portfolio can be written as

$$Ps_t = PAs_{t-1} + PW_t \tag{12}$$

After using the definition of predictability (7) for the VAR(1) model we get:

$$\nu(P) = \frac{var\left(P^T A s_{t-1}\right)}{var\left(P^T s_{t-1}\right)} = \frac{E\left(P^T A s_{t-1} s_{t-1}^T A^T P\right)}{E\left(P^T s_{t-1} s_{t-1}^T P\right)} \tag{13}$$

As only s_t is stochastic, A and P can be factored out from the expectation calculation. So eventually we have the covariance matrix of the time series, which we denote by G. Maximizing predictability is eventually a generalized eigenvalue problem:

$$P_{opt} = argmax\left(\nu(P)\right) = argmax\left(\frac{P^T A G A^T P}{P^T G P}\right), \tag{14}$$

The argument of the $argmax$ operator is the so called Rayleigh quotient, where the above becomes the following:

$$AGA^T P = \lambda \tag{15}$$

Current scenario is to keep the number of constituents low, which is an additional constraint to the optimization. On the other hand, to hold the transaction cost as low as possible and also to keep the portfolio complexity low, only a low number of stocks will need to be enabled. The optimization problem now is the trade-off between the maximization of mean reversion speed and the minimization of the cardinality of stocks. Mathematically the equation (14) has an additional constraint

$$\begin{aligned} &P_{opt} = argmax\left(\nu(P)\right) = argmax\left(\frac{P^T A G A^T P}{P^T G P}\right), \\ &\text{subject to } \mathbf{Card}(P) \leq k \end{aligned} \tag{16}$$

The above optimization was performed in two steps. First a suboptimal solution is found by a greedy algorithm. In the next step Simulated Annealing (SA) was applied, as in [12], [5] and [4]. The starting point of the SA was the solution of the first step. To model the dynamics of the stock prices VAR(1) model with or without constant shift being used. However the calibrated regression matrix not applied for predict individual stock prices only to find it's maximal generalized eigenvalue, it is worth to investigate how accurate the

	Without constant shift	With constant shift
Normal data	0.0047619	0.08424908
Normal return	0.07875458	0.10769231
Log Data	0.0021978	0.23736264
Log return	0.19157509	0.29340659

Table 1: Relative frequency of the highest accuracy

regression model for different data types. Here, we will now discuss whether it is useful to apply the calibration on derived data such as the return, the natural logarithm or the logarithmic return of stock prices. The tests were performed on S&P500 data with many configurations. A point in this configuration space includes the width of the calibration window and the starting point. The particular time window range varied between 50 days to 400 days, the end points moved between $2016-01-01$ and $2021-01-01$. The calibration performed with and without the time independent shift for every data model. To be able to compare the accuracy, every regressed data other than normal data (i.e. normal return, log and log return) were transformed back to normal. Then we calculated the mean squared error and recorded which data model has the lowest error for a certain configuration. The results in relative frequency are summarized in Table 1. In this table we can see that higher accuracy is reachable when we incorporate the constant shift into the VAR(1) model. That also applies to a large extent of the cases the log or log return data model are the most accurate. In the future we use the log model to test and compare the effectiveness of different trading algorithms.

An example of regression with constant shift is in Figure 1. Here all the four prediction of data model (ie. the original data, simple return(diff), natural logarithm of the original and the log return) and values of the asset (Adobe) are represented. Here the regressed values can vary to a large extent compared to the real stock values. As mentioned previously the aim is not to predict individual stock prices but create a portfolio which has the highest predictability.

3 Predicting mean-reverting portfolio with LSTM

3.1 LSTM introduction

Long Short Term Memory networks (LSTM) are recurrent neural networks (RNN), designed to learn long-term dependencies by [6]. As an RNN, LSTMs have a repeating modules and each module has the similar structure with four,

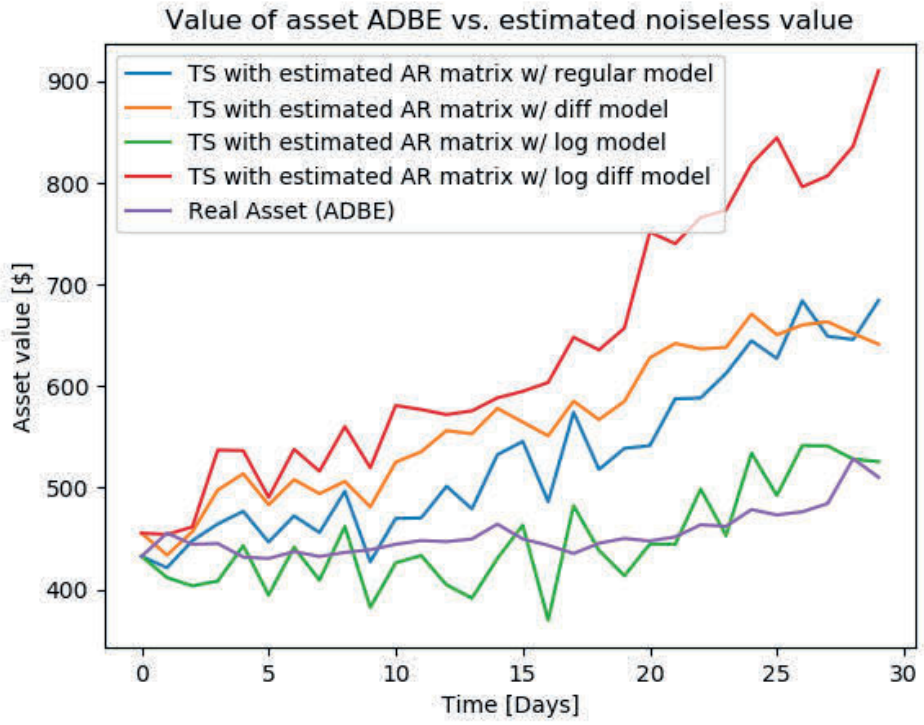


Figure 1: Result of prediction for every data model with constant shift on an example stock (Adobe).

interacting layers. The most important constituent for long-term learning is the state of the module or cell, which is represented by the top horizontal line. It interacts with other layers in the cell via pointwise multiplication ans addition and provides input for the next cell. The way how an LSTM cell constructed by gates makes it possible to remove or add information to the cell state. Since the sigmoid function outputs numbers between zero and one, it can provide weights to each components. A cell in an LSTM network contains three of these sigmoid gates. The first gate is the forget gate which is to decide what information have to be thrown away from the cell state. It takes the h_{t-1} hidden state from previous cell and x_t input, and the output multiplicated with the cell state C_{t-1}.

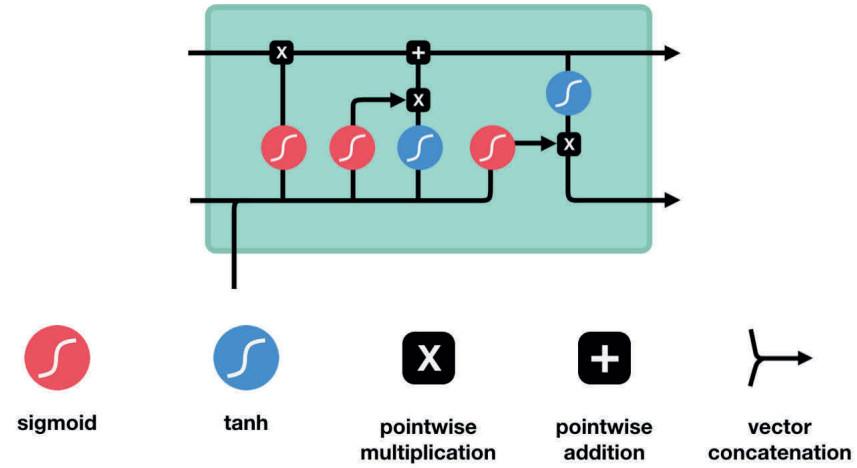


Figure 2: Schematic diagram of LSTM network. Source: [9]

$$o_{forget} = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{17}$$

where σ is the `sigmoid` function, W_f and b_f represents the network in forget gate. The following gate, the `input gate` is for to decide what information should be updated in the cell state which is consist of two layers with two different activation function.

$$f_{input} = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{18}$$

where W_i and b_i represent the neural network of the layer. The output of the input gate is multiplicated with the output of the `tanh` layer that creates a vector which will be used to update cell state values, C_t by pointwise adding the multiplicated result to the old cell state vector C_{t-1}.

$$f_C = \tanh(W_C[h_{t-1}, x_t] + b_C) \tag{19}$$

The final gate, the `output gate` decides what should hidden state values should be transferred to the next cell. The output will be the combination of the previous hidden state values, inputs of the actual cell and the updated cell state values.

$$C_t = o_{forget} * C_{t-1} + f_{input} * f_C \tag{20}$$

First the third sigmoid layer decides what parts of the cell state will be treated as output.

$$o_{output} = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{21}$$

The cell state C_t runs through the tanh layer, this makes the state values between -1 and 1, and multiply it by the output of the sigmoid gate o_{output}.

$$h_t = o_{output} \tanh(C_t) \tag{22}$$

.

The state variables h_t and C_t serve as input for the next module.

3.2 Predicting

The VAR(1) model was applied to fit an analytical model to the data model and from that a well predictable portfolio was created. The regression matrix can be used to predict the future values of the portfolio. However, the prediction is not based on the portfolio itself but on the constituents. Hence, the errors accumulated during the calibration against each stock make the predicted time series very noisy and inaccurate. It is much more reasonable to use the historical values of the portfolio itself to regress future prices. We constructed and trained LSTM recurrent neural network to predict future prices of our portfolio. We trained the network on the same time range as used for VAR(1) calibration. So we created the portfolio with the regression matrix and used as an input for the training. The number of LSTM layers is 4, batch size is 1, epochs is 100. We used 3 consecutive data to predict. In Figure 3 an example of the testing can be seen.

4 Online vs Offline prediction

The common property of the two methods, which are VAR(1) and LSTM is that they are only able regress one time step ahead. The online prediction is fully based on real data. As both VAR(1) and simple LSTM predict only t time step ahead, to regress a time interval we should use previously predicted data, as well.

Online prediction: Using only real data for prediction the next value can only be estimated when all real data are available.

Offline prediction: Using both real and predicted data, longer terms can be predicted by utilizing the previously estimated data as input for the regression.

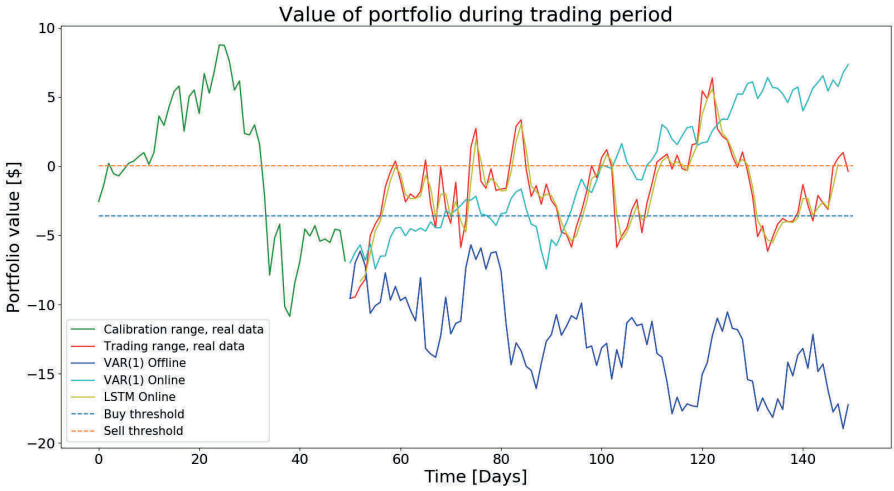


Figure 3: Predicting optimized portfolio. Green curve is the calibration range, blue VAR(1) prediction, red curve real value, yellow is the LSTM prediction.

4.1 Trading strategies

We can build trading strategies by combining the online and offline predictions with the regression methods. The definition of the trading range is essential. It can happen retrospectively or prospectively. As we incorporate the LSTM or VAR(1) regression values into the simple mean-reverting trading strategy the information they carry modifies the logic. As the selling or buying events are triggered by the value of the portfolio, the final decision is affected by the value of the prediction. As if the trader has cash in hand and the purchase event is triggered the real buying will be performed if the next estimated value is higher than the current. Otherwise at least one step will be waited out. Similar logic occurs when there is a portfolio in hand, see Figure 4.

Online strategies: The online strategy uses only the online prediction. Therefore the trading range can be estimated using the historical data.

Offline strategies: The offline trading strategy involves long-term price prediction regressing still only one time step ahead and incorporating predicted data for farther estimation. The method simply involves more predicted data successively as regression goes further ahead. This provides another method to

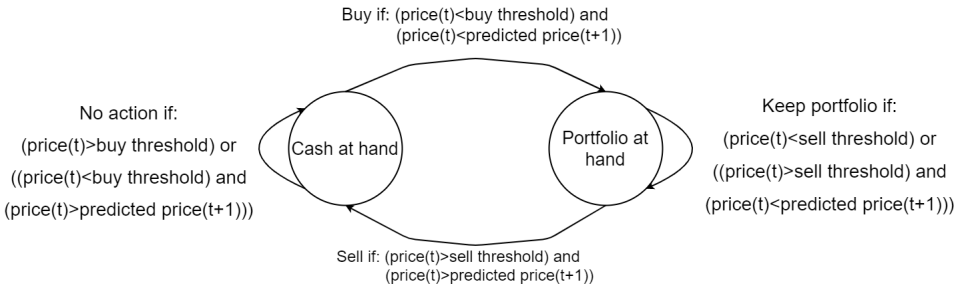


Figure 4: Schematic diagram of the general trading workflow.

estimate the trading range, which is utilized in this strategy both for VAR(1) and LSTM.

5 Performance test

The performance tests were carried out for all the 4 trading configurations, ie. combining online and offline prediction with VAR(1) and LSTM methods. The initial invested amount was \$10000. Each configuration was run with 5 different sparsities: $(3, 5, 8, 11, 14)$. The data was S&P500 daily close stock prices between 01/01/2016 and 12/31/2020. For a sparsity the trading was repeated 30 times with different calibration window lengths and positions.

The length of trading time was constrained to 100 days. This was necessary to be able to compare each trading cases, as we have limited length of data and the possible trading length could vary from few weeks to few years. In case when portfolio was at hand at the end of trading range because price did not hit sell threshold, the return was calculated by the actual value of portfolio subtracted the price at buying. During the test we see that at most 2 buy-sell events were performed. When 0 events was performed it was mostly due the trading range was not estimated precisely and the buy threshold was not hit. At this point we did not incorporate the trading fees as our work focused on the comparison of trading strategies detailed above.

In Table 2 we can see the mean gain and the standard deviation of returns for the used strategies.

On figures (5), (6) the distribution of the profits for the used strategies are visualized. We tested the strategies against different trading starting points and calibration window length $(50, 100, 150, 200)$. We applied a floor and ceiling functions for below $-\$100$ and above \$100 respectively only to make the

	3	5	8	11	13
Offline VAR	-3.31/39.52	2.38/20.04	0.30/25.42	1.03/18.53	-0.72/19.49
Offline LSTM	-2.88/30.52	1.08/23.18	-1.37/15.96	1.91/22.62	-0.69/19.38
Online VAR	4.73/28.15	7.86/40.79	3.97/27.32	4.97/31.06	5.88/35.79
Online LSTM	10.51/27.19	8.21/23.16	10.45/20.24	7.41/23.81	9.37/20.09

Table 2: The mean and standard deviation in terms of [$] of trade performance for trade methods and sparsity

useful part of distribution more detailed, this did not affected the calculation of the mean and standard deviation in Table 2.

It is clear from the table and from the figures that the most effective strategy is the online LSTM. It was able to consistently produce positive profits. Note that the offline data plots on Figures 5 and 6 very high peaks can be seen at 0 profit. These are due to inaccurate estimation of the trading range and no action was performed.

6 Conclusion and future works

We have presented how accurately we predict the S&P500 stocks with VAR(q) using different data models. We conclude that it is worth calibrating with log or log return data model. In the second part we have explained how the prediction of the sparse mean reverting portfolio, which was created by VAR(1), can be improved. The tests were performed on online and offline prediction. We modified the simple mean reverting trading logic by adding the predicted values to the trading decision process. We found that online LSTM outperforms all VAR(1) predictions and results in a positive expected profit when that is used in a simple trading algorithm in case of online prediction. One of the crucial things to increase profit is to estimate the trading range very precisely. One possible way is to create a Seq2seq LSTM neural network, that is able to predict more than one time step ahead. This can be the direction of future research.

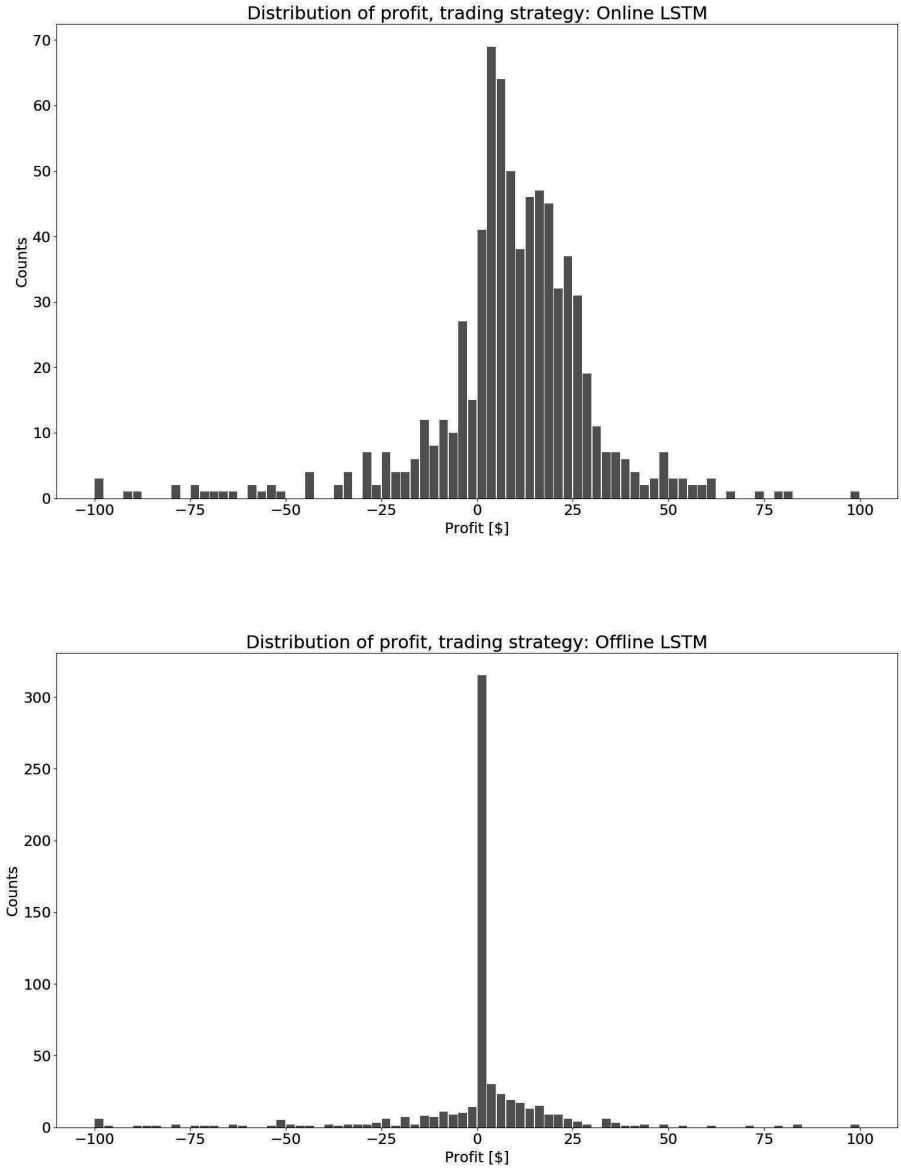


Figure 5: Profit histogram of LSTM prediction on online and offline data.

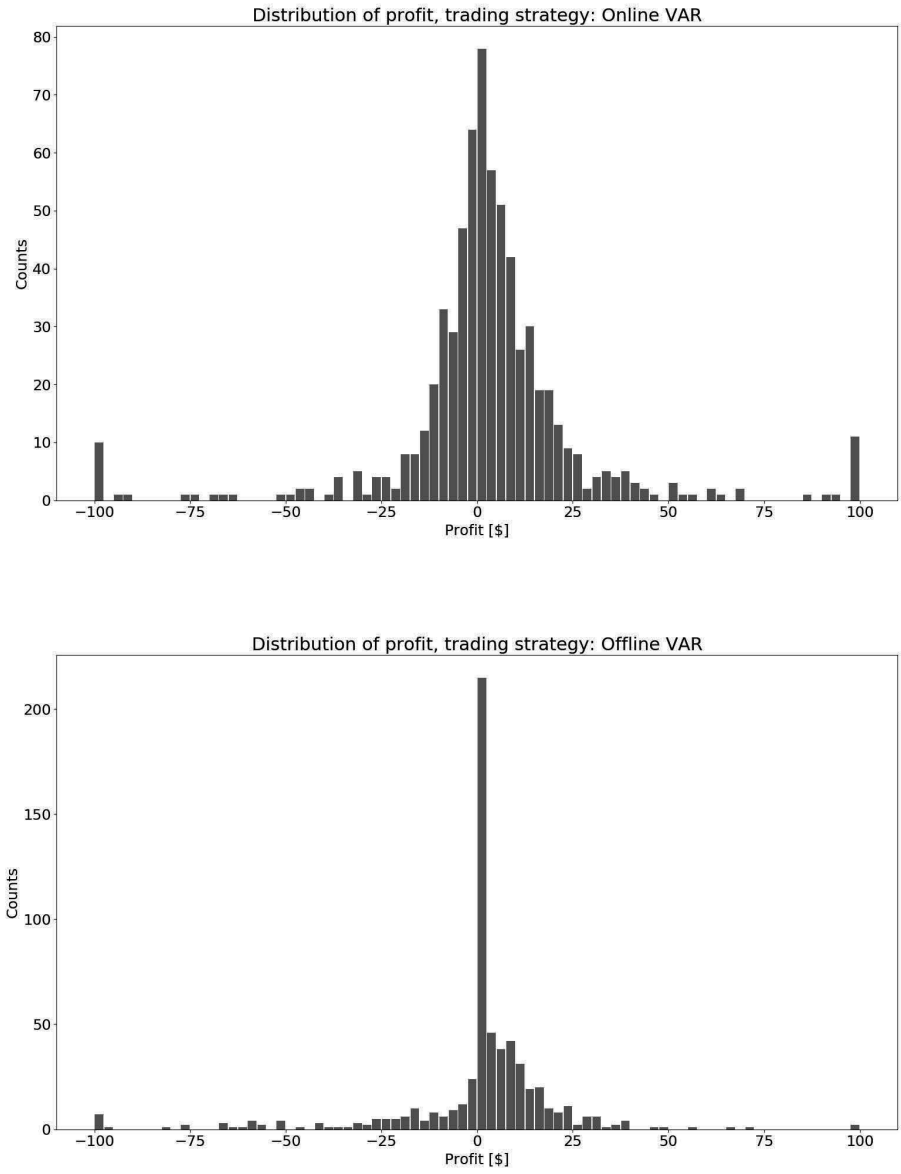


Figure 6: Profit histogram of VAR prediction on online and offline data.

References

[1] O. Banerjee, L. El Ghaoui, A. d'Aspremont, Model selection through sparse maximum likelihood estimation,*J. Mach. Learn. Res.*, **9** (2008) 485—516 ⇒288

[2] G. E. Box, G. C. Tiao, A canonical analysis of multiple time series *Biometrika*, **64** (1977) 355 ⇒288, 290

[3] A. d'Aspremont, Identifying small mean reverting portfolios, *Quant. Finance*, **11** (2011) 351–364 ⇒288, 290

[4] N. Fogarasi, J. Levendovszky, Sparse, mean reverting portfolio selection using simulated annealing,*Quant. Finance*, **11** (2011) 351–364 ⇒288, 292

[5] S. Geman, D. Geman, Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, *IEEE Trans. Patt. Anal. Mach. Intellig*, **6** (1984) 721–741 ⇒292

[6] S. Hochreiter, J. Schmidhube, Long Short-Term Memories, *Neural Computation*, **9** (1997) 1735–1780 ⇒293

[7] H. Lütkepohl, *New Introduction to Multiple Time Series Analysis*, Springer (1993) ⇒291

[8] L. S. Ornstein, G. E. Uhlenbeck, On the theory of the Brownian motion, *Phys. Rev*, **36** (1930) 823 ⇒290

[9] M. Phi, Illustrated Guide to LSTM's and GRU's: A step by step explanation, Sep 24, 2018. `https://tinyurl.com/aus2xtjx` ⇒295

[10] A. Rácz, N. Fogarasi, Improved sparse mean reverting portfolio selection using Simulated Annealing and Extreme Learning Machine, submitted to *Algorithmic Finance* ⇒289

[11] V.-D. Ta, C.-M. Liu, D. A. Tadesse, Portfolio optimization-based stock prediction using long-short term memory network in quantitative trading, *Applied Sciences*, **10** (2020) 437 ⇒289

[12] P. Salamon, P. Sibani, R. *Frost, facts, conjectures, and improvements for simulated annealing,* SIAM Monographs on Mathematical Modeling and Computation. Society for Industrial and Applied Mathematics, **36** (2002) ⇒292

[13] A. Yadav, C. K. Jha, A. Sharan, Optimizing LSTM for time series prediction in Indian stock market, *Procedia Computer Science*, **167** (2020) 2091–2100 ⇒289

Received: October 6, 2021 • Revised: November 3, 2021

Acta Univ. Sapientiae, Informatica **13,** 2 (2021) 303–323

DOI: 10.2478/ausi-2021-0014

Dual quaternion-based osculating circle algorithm for finding intersection curves

Vahide BULUT
Izmir Katip Celebi University,
Department of Engineering Sciences,
Cigli, Izmir, 35620, Turkey
email: vahide.bulut@ikcu.edu.tr

Abstract. The intersection of surfaces is a fundamental process in computational geometry and computer-aided design applications to build and interrogate complex shapes in the computer. This paper presents a novel and simple dual quaternion-based osculating circle DQOC algorithm to find the intersection curve of two regular surfaces based on the osculating circle concept and dual quaternions. Additionally, we expressed the natural equations of the intersection curve. We have also demonstrated the superiority of our method through numerical examples.

1 Introduction

Surface/surface intersections are widely used in computer-aided manufacturing (CAM) / computer-aided design (CAD), such as path planning, animations, and modeling some shapes. Numerical methods are generally preferred for the intersection of two surface cases.

The marching method provides an intersection curve's points sequences by utilizing the local differential geometry [8],[10]. First, the initial point must be determined to proceed through this method, and next, the point continues along the intersection curve by marching. The intersection curve's local

Computing Classification System 1998: I.3.3, I.3.5, J.2
Mathematics Subject Classification 2010: 65D17, 65D05, 53A04, 53A05
Key words and phrases: Surface intersection, Marching method,osculating circle, dual quaternion, intersection curve

geometric properties are used to compute the marching direction plus steps. The marching step is computed via the sequence of points that originated from the first point. Several possible solutions exist for marching directions, including moving in a tangential direction [2], [9] or traveling along a circle [1] or a parabola [12]. However, the most commonly used solution is step length, based on the curve's curvature.

Not only are dual quaternions vital because they perform the solution more robust and straightforward, but they also provide a compact, unambiguous, singularity - free rigid transform using minimal computations. Another positive feature of dual quaternions is that they are the most efficient and most compact form that can be utilized to represent rotation and translation. Additionally, they can solve a problem more rapidly while doing it in fewer steps, and they show the result more clearly. Also, fewer code lines are used for dual quaternions to practice [7].

In literature, some authors have used lengthy calculations to determine the step length and next intersection point. This study presents a new and applicable DQOC algorithm that employs fewer computational calculations to obtain the solution less complicated. Additionally, we obtain closer points to the intersection curve. On the other hand, we present the natural equations of the intersection curve. We also compare our algorithm with the method that Wu and Andrade used [15].

2 Preliminaries

2.1 Dual quaternions

Definition 1 *A dual number is written as*

$$A = a + \varepsilon \bar{a}$$

in which, a *and* $\bar{a}$ *are real numbers and* $\varepsilon^2 = 0$, $\varepsilon \neq 0$ [13], [16].

Definition 2 *An ordinary quaternion is defined as*

$$q = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$$

where $\mathbf{i}, \mathbf{j}, \mathbf{k}$ *are the standard orthonormal basis in* R^3, *providing* $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$, *and* a, b, c, d *are real numbers.*

Quaternions can present a rigid body's rotation according to an axis. Quaternions do not cause any singularity problem and provide the keyframe interpo-

lation better [5], [11], [6].

Definition 3 *A dual quaternion can be expressed by*

$$\hat{Q} = Q + \varepsilon Q^* \tag{1}$$

in which

$$Q = q_r + \vec{q} \qquad Q^* = q_r^* + \vec{q}^*$$

and $\varepsilon^2 = 0$.

Dual quaternions were first described by W. Kingdon Clifford in 1873 [3]. They are different from the real quaternions because they are utilized for both translation and rotation. A dual quaternion's four dual number terms can be interpreted as

$$\hat{Q} = \hat{q}_r + \hat{q}_x\mathbf{i} + \hat{q}_y\mathbf{j} + \hat{q}_z\mathbf{k} \tag{2}$$

or

$$\hat{Q} = s + x\mathbf{i} + y\mathbf{j} + z\mathbf{k} + \varepsilon\left(s_\varepsilon + x_\varepsilon\mathbf{i} + y_\varepsilon\mathbf{j} + z_\varepsilon\mathbf{k}\right).$$

Lemma 4 *If we take the two dual quaternions as*

$$\hat{Q}_1 = Q_1 + \varepsilon Q_1^*, \qquad \hat{Q}_2 = Q_2 + \varepsilon Q_2^*,$$

then, the dual quaternion multiplication can be given with

$$\hat{Q}_1\hat{Q}_2 = Q_1Q_2 + \varepsilon\left(Q_1Q_2^* + Q_2Q_1^*\right).$$

Lemma 5 *Dual conjugate* $\overline{\hat{Q}}$ *and dual quaternion norm can be defined as*

$$\overline{\hat{Q}} = Q - \varepsilon Q^*$$

and

$$\|\hat{Q}\| = \|Q\| + \varepsilon\frac{< Q, Q^* >}{\|Q\|}.$$

Definition 6 *A dual quaternion that satisfies the conditions* $\|\hat{Q}\| = 1$ *and* $< Q, Q^* >= 0$ *is called a unit dual quaternion.*

Lemma 7 *The inverse of a dual quaternion is*

$$\hat{Q}^{-1} = \frac{1}{Q} - \varepsilon \frac{Q^*}{Q^2}$$

in which, $Q \neq 0$.

Lemma 8 *A second conjugation operator for a dual quaternion is*

$$\overline{\hat{Q}}^* = (s, -x, -y, -z, -s_\varepsilon, x_\varepsilon, y_\varepsilon, z_\varepsilon).$$

Lemma 9 *Transformations can be represented by only one dual quaternion. Let* $\hat{Q}$ *and* $\hat{P}$ *be two transformation dual quaternions and* Q_v *be a position vector dual quaternion. Then, the combined transformation* C *can be applied to* Q_v *as*

$$\hat{Q}_v' = \hat{P}\left(\hat{Q}Q_v\overline{\hat{Q}}^*\right)\overline{\hat{P}}^* = \left(\hat{P}\hat{Q}\right)(Q_v)\left(\overline{\hat{Q}}^*\overline{\hat{P}}^*\right) \tag{3}$$

or

$$\hat{C} = \hat{P}\hat{Q} \Rightarrow \hat{Q}_v' = \hat{C}Q_v\overline{\hat{C}}^*.$$

Lemma 10 *Unit dual quaternions represent the three-dimensional 3D rotation with an angle* θ *and a unit axis* $\mathbf{n}$ *when the dual part* $Q^* = 0$.

$$\hat{Q}_r = \left[\cos\left(\tfrac{\theta}{2}\right), \mathbf{n}_x \sin\left(\tfrac{\theta}{2}\right), \mathbf{n}_y \sin\left(\tfrac{\theta}{2}\right), \mathbf{n}_z \sin\left(\tfrac{\theta}{2}\right)\right] [0, 0, 0, 0]$$

Lemma 11 *A pure translation can be expressed in terms of a dual quaternion by*

$$\hat{Q}_t = [1, 0, 0, 0]\left[0, \frac{t_x}{2}, \frac{t_y}{2}, \frac{t_z}{2}\right]$$

Lemma 12 *On the following, a single unit quaternion can be used to denote a rotation followed by a translation as* [7], [18]

$$\hat{Q} = \hat{Q}_t \times \hat{Q}_r. \tag{4}$$

3 Intersection curve

3.1 Geometric concepts of the intersection curve

First and second derivatives of the parametric curve $\boldsymbol{\alpha} = \boldsymbol{\alpha}(s)$ in R^3 in terms of arc-length parameter are given by

$$\boldsymbol{\alpha}'(s) = \mathbf{T}, \qquad \boldsymbol{\alpha}''(s) = \kappa\mathbf{N} \tag{5}$$

in which the vectors $\mathbf{T}$, $\mathbf{N}$ correspond to the components of the Frenet-Serret frame, and κ represent the curvature of this curve.

Definition 13 *Assume that the points* P, Q, *and* M *be on the same curve. If the points* P *and* M *tend to* Q, *the circle's limit that passes through all these points is called the osculating circle at* Q.

The osculating circle can then be used as an approximation of the intersection curve because the curve's osculating circle at a point has the same tangent and curvature as the curve at that point [4].

Suppose that the intersection curve $\boldsymbol{\alpha}(s)$ is given as of regular surfaces $\mathbf{S}_1(u, v)$ and $\mathbf{S}_2(r, s)$. The tangent vector of this curve can be written using the surfaces' unit normal vectors $\mathbf{n}_1$ and $\mathbf{n}_2$ at the point P with following equation [17].

$$\mathbf{T} = \frac{\mathbf{n}_1 \times \mathbf{n}_2}{\|\mathbf{n}_1 \times \mathbf{n}_2\|}. \tag{6}$$

In this paper, the intersection curve is obtained by the transversal intersection of two regular surfaces.

3.2 Marching algorithm

Finding the starting points of the intersection curves plays a vital role in tracing methods. Analytically, we need to compute the below equation to determine these two parametric surfaces' intersection.

$$\mathbf{S}_1(u, v) = \mathbf{S}_2(r, s). \tag{7}$$

In the case of not intersecting these surfaces, the minimum distance is computed between two surfaces to determine the starting points. For instance, if the two surfaces exist in the same bounding box, we can compute a starting point using the bounding box method.

For every next point, we can find the step vector at each point by using the marching algorithm and the marching process can be performed in parametric or Cartesian space. In Cartesian spaces, the marching method uses a fixed-step. However, with our algorithm, we consider the curve's local geometry instead of a fixed step [14].

This paper utilizes the osculating circle to compute the length of each step. The reason is that the intersection curve's osculating circle $\boldsymbol{\alpha}(s)$ can best approximate the curve at the same point [15].

4 Proposed method

Our proposed method focused on the osculating circle of the intersection curve via using dual quaternions:

- We need to find initial points.
- We can compute the center and the osculating circle's radius by using these initial points.
- We obtain the next intersection point using dual quaternions.

In our method, we begin with determining the starting points by applying the steps as follows:

Step 1: Determine the point P to be a starting point using the equation (7).

Step 2: Obtain the point Q. It is accepted that the distance between the points P and Q is $\frac{\mathsf{L}}{(\mathsf{k}.10)}$. The initial value of k is taken as 2. Here, L indicates the step length.

Step 3: Increase the value of k by one, and continue by Step 2 if the point Q becomes a singular point. On the other hand, if the singular case remains the same, the value of k can be increased sequentially till 10.

Step 4: Continue with step 1 if the point Q can not be found because of the singularity.

Next, to find the osculating circle's center and radius, we utilized the method in [15]. We can compute the osculating circle's center with the following linear system of equations.

$$\begin{cases} \mathsf{C}\mathbf{u} = \mathsf{P}\mathbf{u} \\ \mathsf{C}\mathbf{v} = \mathsf{Q}\mathbf{v} \\ \mathsf{C}\mathbf{w} = \mathsf{Q}\mathbf{w} \end{cases} \tag{8}$$

in which C is the osculating circle's center point at Q. This center point C is obtained via the intersection of three planes that have the normal vectors $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{w}$. Also, the distance between C and Q indicates the osculating circle's radius.

The step length can be obtained right after the osculating circle approximation. Wu and Andrade first computed the normal vector to the circumference plane. After the transformations of translation and rotation, they moved the osculating circle to the XOY plane by placing the center to $\mathsf{O} = (0, 0, 0)$. Consequently, they obtained the transformed points and then determined whether

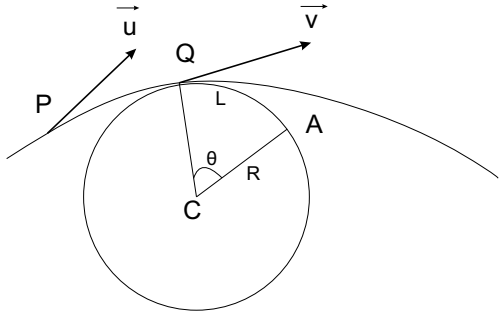


Figure 1: The circular step is obtained by the proposed method.

the orientation of the arc $\left(\overset{\frown}{PQ}\right)$ was clockwise or counterclockwise. Next, they found point A' after some computations and applied the inverse transformations to point A' to ascertain the next intersection point A [15]. For this process, they used very long calculations to find the next intersection point A.

In our new method, we determined θ as an increment L as shown in Fig. 1. We can compute the next intersection point A using only the equation (3) below as

$$\begin{aligned} A = \hat{Q}Q\overline{\hat{Q}}^{*} \quad &= [1,\, 0,\, 0,\, 0] + \varepsilon[0,\, c_1 + (q_1 - c_1)\cos\theta + (c_2 - q_2)\sin\theta, \\ &\quad c_2 + (q_2 - c_2)\cos\theta + (q_1 - c_1)\sin\theta,\, q_3]. \end{aligned} \tag{9}$$

Using the equation (9), we translate the osculating circle's center C to the origin O, implement the rotation, and then re-translate C to its initial position and find the next intersection point applying only unit dual quaternions. The proposed dual quaternion based osculating circle DQOC algorithm is given in Algorithm 1.

5 Natural equation of the intersection curve

We can obtain the natural equation of the intersection curve obtained by the transversal intersection of two regular surfaces, using the Darboux frames of these surfaces.

Theorem 14 *Let two surfaces be given as* $\mathbf{S}_1(u, v)$ *and* $\mathbf{S}_2(r, s)$*, and the intersection curve of these surfaces be* $\boldsymbol{\alpha}(s)$ *parametrized with arc length parameter.*

We can expand series of the intersection curve $\boldsymbol{\alpha}(s)$ *at* $s = 0$ *as*

$$\boldsymbol{\alpha}(s) = \frac{s}{1!}\frac{d\boldsymbol{\alpha}}{ds} + \frac{s^2}{2!}\frac{d^2\boldsymbol{\alpha}}{ds^2} + \frac{s^3}{3!}\frac{d^3\boldsymbol{\alpha}}{ds^3} + ... \tag{10}$$

We can write the natural equation of the intersection curve $\boldsymbol{\alpha}(s)$ *at* $s = 0$ *in terms of the Darboux frame coefficients of the surfaces* $\mathbf{S}_1$ *and* $\mathbf{S}_2$ *with the following equation.*

$$\begin{aligned}
\boldsymbol{\alpha}(s) \quad &= \frac{s}{1!}t + \frac{s^2}{2!}\frac{1}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|}\left[-a\left(\tau_{g_1} - \tau_{g_2}\right)\mathbf{n}\right. \\
&\left. -k_{n_2}\mathbf{g}_1 + \tau_{g_1} b\mathbf{n}_1 + k_{n_1}\mathbf{g}_2 - \tau_{g_2} c\mathbf{n}_2\right] \\
&= +\frac{s^3}{3!}\frac{1}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|}\left[-a'\left(\tau_{g_1} - \tau_{g_2}\right) - a\left(\tau'_{g_1} - \tau'_{g_2}\right)\right. \\
&\left. +\frac{1}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|}a^2\left(\tau_{g_1} - \tau_{g_2}\right)^2 + k_{n_2}\mathbf{g}_1\right. \\
&\left. -\tau_{g_1} b k_{n_1} - k_{n_1} k_{g_2} - \tau_{g_2} k_{n_2}\right]\mathbf{t} \\
&+\frac{s^3}{3!}\frac{1}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|}\left[\frac{1}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|}a\left(\tau_{g_1} - \tau_{g_2}\right)k_{n_2}\right. \\
&\left. -k'_{n_2} - \tau^2_{g_1} b\right]\mathbf{g}_1 \\
&+\frac{s^3}{3!}\frac{1}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|}\left[-\frac{1}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|}a\left(\tau_{g_1} - \tau_{g_2}\right)\tau_{g_1} b\right. \\
&\left. -k_{n_2}\tau_{g_1} + \tau'_{g_1} b + \tau_{g_1} b'\right]\mathbf{n}_1 \\
&+\frac{s^3}{3!}\frac{1}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|}\left[-\frac{1}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|}a\left(\tau_{g_1} - \tau_{g_2}\right)k_{n_1} + k'_{n_1} + \tau^2_{g_2}\right]\mathbf{n}_2 \\
&+\frac{s^3}{3!}\frac{1}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|}\left[+\frac{1}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|}a\left(\tau_{g_1} - \tau_{g_2}\right)\tau_{g_2} c\right. \\
&\left. +k_{n_1}\tau_{g_2} - \tau'_{g_2} c - \tau_{g_2} c'\right]\mathbf{n}_2 \\
&+...
\end{aligned} \tag{11}$$

Proof. Since the intersection curve will take place on both surfaces, Darboux frames of this curve on the surfaces $\mathbf{S}_1$ and $\mathbf{S}_2$ can be taken as $\{\mathbf{t}_1, \mathbf{g}_1, \mathbf{n}_1\}$ and $\{\mathbf{t}_2, \mathbf{g}_2, \mathbf{n}_2\}$. These frames are right-handed orthogonal frames associated with each point of the intersection curve $\boldsymbol{\alpha}(s)$, where $\mathbf{t}_1$ and $\mathbf{t}_2$ are the unit tangent vectors of $\boldsymbol{\alpha}(s)$ at the surfaces $\mathbf{S}_1$ and $\mathbf{S}_2$; $\mathbf{n}_1$ and $\mathbf{n}_2$ are the unit normal vectors of the surfaces $\mathbf{S}_1$ and $\mathbf{S}_2$; $\mathbf{g}_1$ and $\mathbf{g}_2$ are the geodesic normals of the surfaces $\mathbf{S}_1$ and $\mathbf{S}_2$, respectively. On the other hand, we can write the unit tangent vector of the intersection curve using the equation (6) as

$$\frac{d\boldsymbol{\alpha}}{ds} = \frac{\mathbf{n}_1 \wedge \mathbf{n}_2}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|} = \mathbf{t} = \mathbf{t}_1 = \mathbf{t}_2. \tag{12}$$

Next, we can get the second derivative of the above equation as follows:

$$\begin{aligned}
\frac{d^2\boldsymbol{\alpha}}{ds^2} &= \frac{1}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|}\left[\mathbf{n}_1' \wedge \mathbf{n}_2 + \mathbf{n}_1 \wedge \mathbf{n}_2'\right] \\
&= \frac{1}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|}\left[(-k_{n_1}\mathbf{t} - \tau_{g_1}\mathbf{g}_1) \wedge \mathbf{n}_2 + \mathbf{n}_1 \wedge (-k_{n_2}\mathbf{t} - \tau_{g_2}\mathbf{g}_2)\right] \\
&= \frac{1}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|}\left[k_{n_1}\mathbf{n}_2 - \tau_{g_1}\left((\mathbf{n}_1\mathbf{n}_2)\,\mathbf{t} - (\mathbf{t}\mathbf{n}_2)\,\mathbf{n}_1\right)\right. \\
&\left. -k_{n_2}\mathbf{g}_1 - \tau_{g_2}\left((\mathbf{n}_1\mathbf{t})\,\mathbf{n}_2 - (\mathbf{n}_1\mathbf{n}_2)\,\mathbf{t}\right)\right] \\
&= \frac{1}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|}\left[-a\,(\tau_{g_1} - \tau_{g_2})\,\mathbf{t}\right. \\
&\left. -k_{n_2}\mathbf{g}_1 + \tau_{g_1} b\mathbf{n}_1 + k_{n_1}\mathbf{g}_2 - \tau_{g_2} c\mathbf{n}_2\right]
\end{aligned} \tag{13}$$

where, $\mathbf{n}_1\mathbf{n}_2 = a$, $\mathbf{t}\mathbf{n}_2 = b$ and $\mathbf{n}_1\mathbf{t} = c$. Also, we can write the third derivative of the equation (12) with below equation as

$$
\begin{aligned}
\frac{d^3\boldsymbol{\alpha}}{ds^3} &= \frac{1}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|}[-a'(\tau_{g_1}-\tau_{g_2})\mathbf{t} \\
&-a\left(\tau'_{g_1}-\tau'_{g_2}\right)\mathbf{t}-a(\tau_{g_1}-\tau_{g_2})\mathbf{t}'-k'_{n_2}\mathbf{g}_1-k_{n_2}\mathbf{g}'_1 \\
&+\tau'_{g_1}b\mathbf{n}_1+\tau_{g_1}b'\mathbf{n}_1+\tau_{g_1}b\mathbf{n}'_1+k'_{n_1}\mathbf{g}_2 \\
&+k_{n_1}\mathbf{g}'_2-\tau'_{g_2}c\mathbf{n}_2-\tau_{g_2}c'\mathbf{n}_2-\tau_{g_2}c\mathbf{n}'_2] \\
&= \frac{1}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|}\left[-a'(\tau_{g_1}-\tau_{g_2})-a\left(\tau'_{g_1}-\tau'_{g_2}\right)\right. \\
&+\frac{1}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|}a^2(\tau_{g_1}-\tau_{g_2})^2+k_{n_2}\mathbf{g}_1-\tau_{g_1}bk_{n_1} \\
&\left.-k_{n_1}k_{g_2}-\tau_{g_2}k_{n_2}\right]\mathbf{t} \\
&+\frac{1}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|}\left[\frac{1}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|}a(\tau_{g_1}-\tau_{g_2})k_{n_2}-k'_{n_2}-\tau^2_{g_1}b\right]\mathbf{g}_1 \\
&+\frac{1}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|}\left[-\frac{1}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|}a(\tau_{g_1}-\tau_{g_2})\tau_{g_1}b\right. \\
&\left.-k_{n_2}\tau_{g_1}+\tau'_{g_1}b+\tau_{g_1}b'\right]\mathbf{n}_1 \\
&+\frac{1}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|}\left[-\frac{1}{\|n_1 \wedge n_2\|}a(\tau_{g_1}-\tau_{g_2})k_{n_1}+k'_{n_1}+\tau^2_{g_2}\right]\mathbf{g}_2 \\
&+\frac{1}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|}\left[+\frac{1}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|}a(\tau_{g_1}-\tau_{g_2})\tau_{g_2}c\right. \\
&\left.+k_{n_1}\tau_{g_2}-\tau'_{g_2}c-\tau_{g_2}c'\right]\mathbf{n}_2.
\end{aligned}
\tag{14}
$$

If we continue taking this derivative process as above, we can find the n^{th} derivative with following equation.

$$
\frac{d^n\boldsymbol{\alpha}}{ds^n} = u_n\mathbf{t}+v_n\mathbf{g}_1+w_n\mathbf{n}_1+v^*_n\mathbf{g}_2+w^*_n\mathbf{n}_2 \tag{15}
$$

where, u_n, v_n, w_n, v_n^* and w_n^* are the known functions of sequential derivatives of k_{n_i}, k_{g_i} and τ_{g_i}, $i = 1, 2$. Taking derivative of (15), we can get the equation below.

$$\begin{aligned}
\frac{d^{n+1}\boldsymbol{\alpha}}{ds^{n+1}} &= \frac{du_n}{ds}\mathbf{t} + u_n\mathbf{t}' + \frac{dv_n}{ds}\mathbf{g}_1 + v_n\mathbf{g}_1' + \frac{dw_n}{ds}\mathbf{n}_1 \\
&+ w_n\mathbf{n}_1' + \frac{dv_n^*}{ds}\mathbf{g}_2 + v_n^*\mathbf{g}_2' + \frac{dw_n^*}{ds}\mathbf{n}_2 + w_n^*\mathbf{n}_2' \\
&= \left[\frac{du_n}{ds} - k_{g_1}v_n - k_{n_1}w_n - k_{g_2}v_n^* - k_{n_2}w_n^*\right. \\
&\left. - \frac{u_n}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|}a\left(\tau_{g_1} - \tau_{g_2}\right)\right]\mathbf{t} \\
&+ \left(\frac{dv_n}{ds} - \tau_{g_1}w_n - \frac{u_n}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|}ak_{n_2}\right)\mathbf{g}_1 \\
&+ \left(\frac{dw_n}{ds} + \tau_{g_1}v_n + \frac{u_n}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|}\tau_{g_1}b\right)\mathbf{n}_1 \\
&+ \left(\frac{dv_n^*}{ds} - \tau_{g_2}w_n^* + \frac{u_n}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|}k_{n_1}\right)\mathbf{g}_2 \\
&+ \left(\frac{dw_n^*}{ds} + \tau_{g_2}v_n^* - \frac{u_n}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|}\tau_{g_2}c\right)\mathbf{n}_2.
\end{aligned} \tag{16}$$

Thus, we can obtain the recurrence formulas as

$$
\begin{aligned}
u_{n+1} &= \frac{du_n}{ds} - k_{g_1} v_n - k_{n_1} w_n - k_{g_2} v_n^* - k_{n_2} w_n^* \\
&\quad - \frac{u_n}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|} a \left(\tau_{g_1} - \tau_{g_2}\right), \\
v_{n+1} &= \frac{dv_n}{ds} - \tau_{g_1} w_n - \frac{u_n}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|} a k_{n_2}, \\
w_{n+1} &= \frac{dw_n}{ds} + \tau_{g_1} v_n + \frac{u_n}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|} \tau_{g_1} b, \\
v_{n+1}^* &= \frac{dv_n^*}{ds} - \tau_{g_2} w_n^* + \frac{u_n}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|} k_{n_1}, \\
w_{n+1}^* &= \frac{dw_n^*}{ds} + \tau_{g_2} v_n^* - \frac{u_n}{\|\mathbf{n}_1 \wedge \mathbf{n}_2\|} \tau_{g_2} c.
\end{aligned}
\tag{17}
$$

Finally, using the equations (12), (13), (14), (15), (16) and (17) the natural equation of the intersection curve given in (11) can be written. □

6 Experimental results

This study compares our proposed dual quaternion-based osculating circle DQOC algorithm with Wu and Andrade's [15] method. To determine the intersection curve points, we need to check the parallelism between the tangent vectors $\mathbf{u}$ and $\mathbf{v}$ at the points P and Q, and so on. If $\mathbf{u}$ and $\mathbf{v}$ are parallel, the curvature radius goes to infinity. Therefore, the best choice for step vector is the tangent vector at Q. We used the value $\cos\theta$, which is the angle between the tangent vectors of consecutive intersection points. If $\cos\theta \geq 1 - \varepsilon$ or $\cos\theta \leq -1 + \varepsilon$, we can assume that the vectors $\mathbf{u}$ and $\mathbf{v}$ are parallel. If, $\cos\theta \longrightarrow 1$ or $\cos\theta \longrightarrow -1$, then radius of the osculating circle $\longrightarrow \infty$. Because of this situation, we defined a parallel threshold value called ε above. In the following applications, we obtained closer points to the intersection curve than their method did. On the other hand, the number of next intersection points on the tangential direction of the point Q can change according to the threshold value. If the threshold value grows steadily, then, the number of next intersection points may increase on the tangential direction of the point Q. If the threshold value is taken as very small, next intersection points are

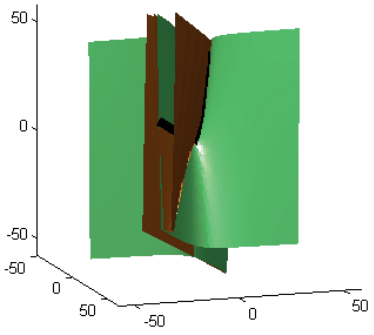
Algorithm 1 Dual Quaternion based Osculating Circle DQOC Algorithm

Input: Two surfaces, S_i $(i = 1, 2)$
Output: Points on the intersection curve

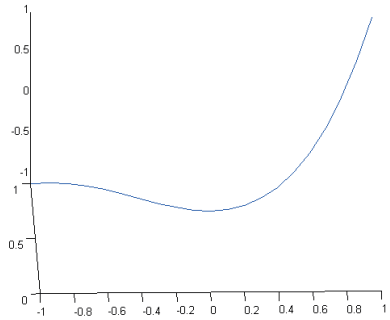
global Points, pointA, pointP, pointQ, pointM, center
global Tangent vectors, pointPtan, pointQtan
global Normal vector of plane, normal; radius of osculating circle, R
global Transformation matrices and their elements, lambda, V, TMatrix, RMatrix, pointPTrans, pointQTrans, directionValue, beta, dist, L;
▷ Find the initial points of the intersection of two surfaces S_i $(i = 1, 2)$
count ← 1;
tValue1 ← 0.0;
tStartValue ← 0.0;
tValue1 ← tStartValue + count ∗ 0.005;
findInitialPointsPQM(tValue1);
while count < 2 **do**
    count ← count + 1;
    parallel ← isParallel(pointPtan, pointQtan);
    if parallel **then**
        pointA ← getNewPointAlongTangent(pointQ, L);
    else
        findCenterOfOsculatingCircle();
        R ← getRadiusOfOsculator(pointQ, center);
        normal ← getNormalVector(center, pointQ, pointP);
        lambda ← getLambda(normal);
        V ← getV(normal);
        TMatrix ← getTMatrix(center); ▷ Obtain translation matrix
        RMatrix ← getRMatrix(normal); ▷ Obtain rotation matrix
        findTransformedPoints();
        directionValue ← getDirectionValue(pointPTrans, pointQTrans);
        beta ← getBeta(directionValue);
        pointA ← getPointA(R, beta, TMatrix, RMatrix);
        dist ← getDistanceToCurve(pointA);
    end if
    pointAtan ← getTangentOfPoint(pointA);
    updatePoints();
end while

going to be far from the intersection curve. First, we present intersection of two surfaces and intersection curves of these surfaces as seen from the figures (6) and (6) for example 1, figures (6(a)) and (6) for example 2, and figures (10(a)) and (10) for example 3. Next, we compared our method with Wu and Andrade's [15] method for different parallel threshold values as seen from the figures (3), (4) and (5) for example 1, figures (7), (8) and (9) for example 2, and figures (11), (12) and (13) for example 3.

Example 1: Let two surfaces be given as $\mathbf{S}_1 : xy - z = 0$ and $\mathbf{S}_2 : z = x^3 + xy - 2z = 0$. Then, the intersection curve of these two surfaces can be expressed as $\boldsymbol{\alpha}(t) = (t, t^2, t^3)$.

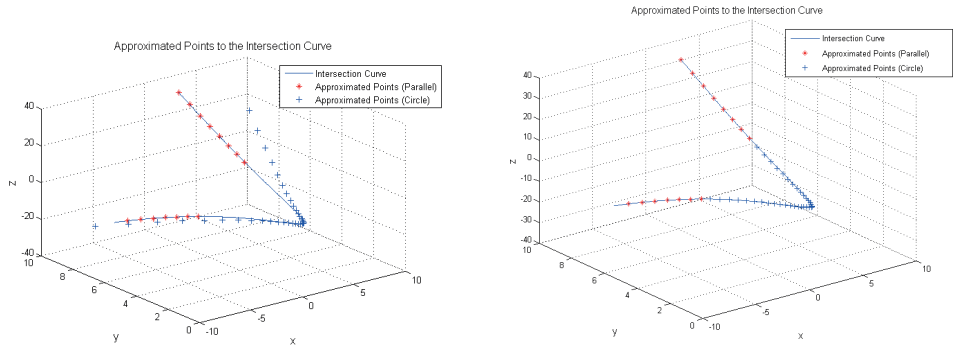


(a) The intersection of surfaces $\mathbf{S}_1 : xy - z = 0$ $\mathbf{S}_2 : z = x^3 + xy - 2z = 0$



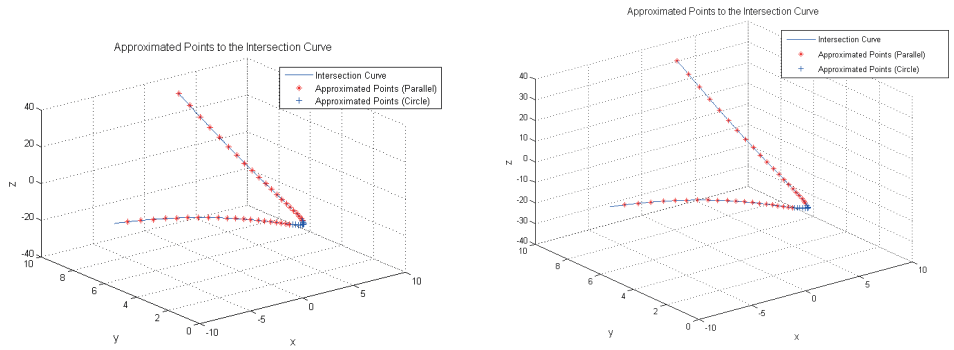
(b) The intersection curve $\boldsymbol{\alpha}(t) = (t, t^2, t^3)$

Figure 2: Intersection curve of two given surfaces.



(a) (b)

Figure 4: The figure in (a) shows the approximated points found by the method in ([15]), and the figure (b) shows the approximated points found by the proposed method based on the threshold value 10^{-9}.

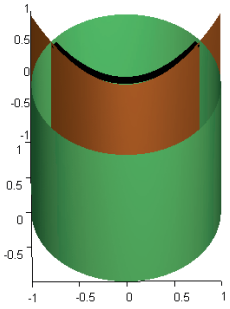


(a) (b)

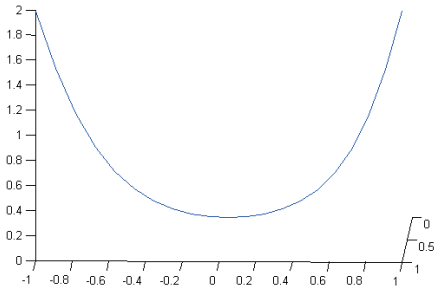
Figure 3: The figure in (a) shows the approximated points found by the method in ([15]), and the figure (b) shows the approximated points found by our method based on the threshold value 10^{-7}.

Figure 5: Comparison of distances to the intersection curve.

Example 2: The surfaces are given as $\mathbf{S}_1 : x^2 + y^2 = z$ and $\mathbf{S}_2 : y = x^2$. We can define the intersection curve with $\boldsymbol{\alpha}(t) = \left(t,\, t^2,\, t^2 + t^4\right)$.

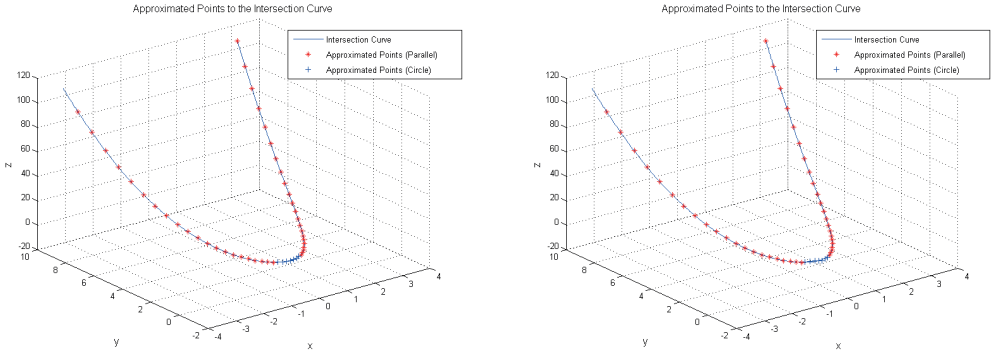


(a) The intersection of surfaces $\mathbf{S}_1 : x^2 + y^2 = z$ and $\mathbf{S}_2 : y = x^2$.



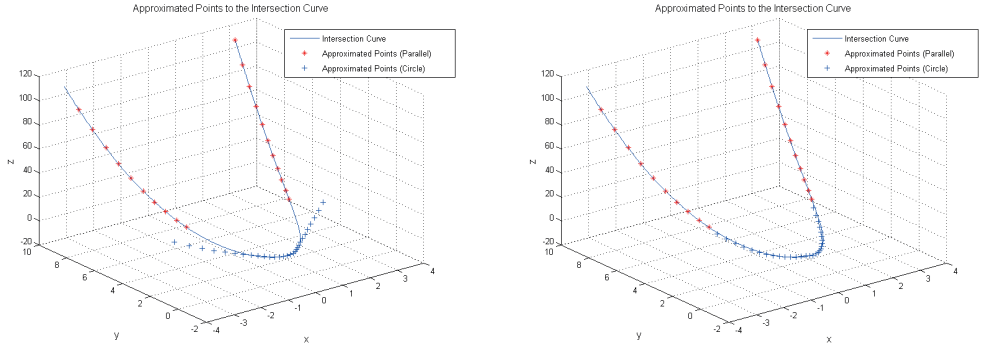
(b) The intersection curve $\boldsymbol{\alpha}(t) = \left(t,\, t^2,\, t^2 + t^4\right)$.

Figure 6: Intersection curve of two given surfaces.



(a) (b)

Figure 7: The figure in (a) shows the approximated points found by the method in ([15]), and the figure (b) shows the approximated points found by our method based on the threshold value 10^{-7}.



(a) (b)

Figure 8: The figure in (a) shows the approximated points found by the method in ([15]), and the figure (b) shows the approximated points found by the proposed method based on the threshold value 10^{-9}.

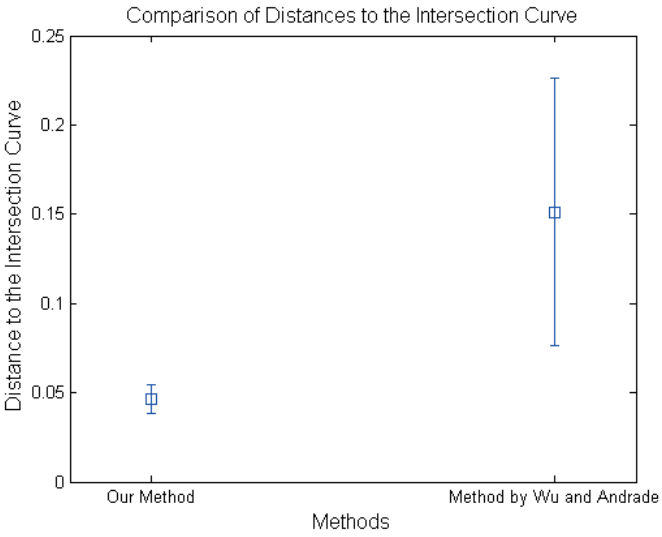
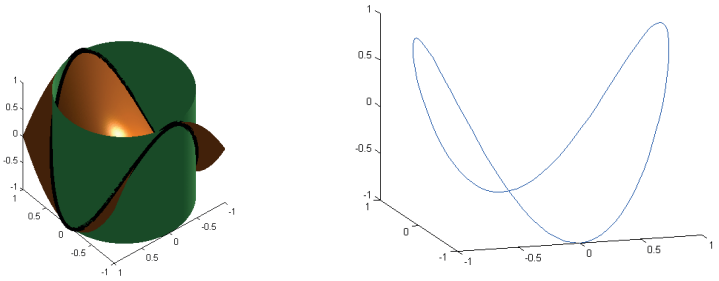


Figure 9: Comparision of distances to the intesection curve.

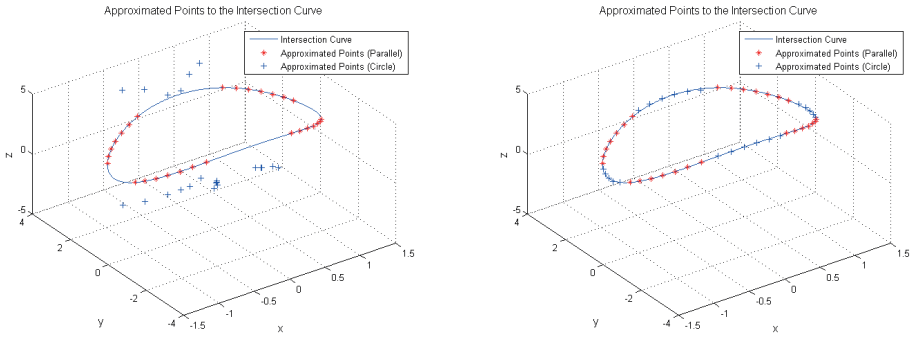
Example 3: Let's take two surfaces as $\mathbf{S}_1 : x^2 + y^2 = 1$ and $\mathbf{S}_2 : z = x^2 - y^2$. We can find the intersection curve as $\boldsymbol{\alpha}(t) = (\cos t, \sin t, \cos 2t,)$.



(a) The intersection of surfaces $\mathbf{S}_1 : x^2 + y^2 = 1$ and $\mathbf{S}_2 : z = x^2 - y^2$.

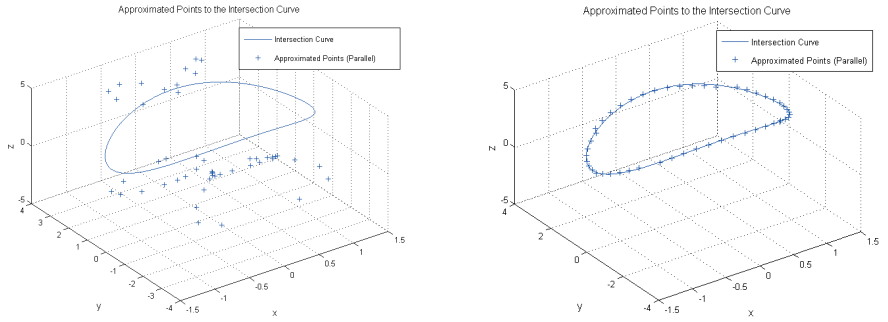
(b) The intersection curve $\boldsymbol{\alpha}(t) = (\cos t, \sin t, \cos 2t,)$.

Figure 10: Intersection curve of two given surfaces.



(a) (b)

Figure 11: The figure on the left shows the approximated points found by the method in ([15]), and the one on the right shows the approximated points found by our method based on the threshold value 10^{-7}.



(a) (b)

Figure 12: The figure in (a) shows the approximated points found by the method in ([15]), and the figure (b) shows the approximated points found by the proposed method based on the threshold value 10^{-9}.

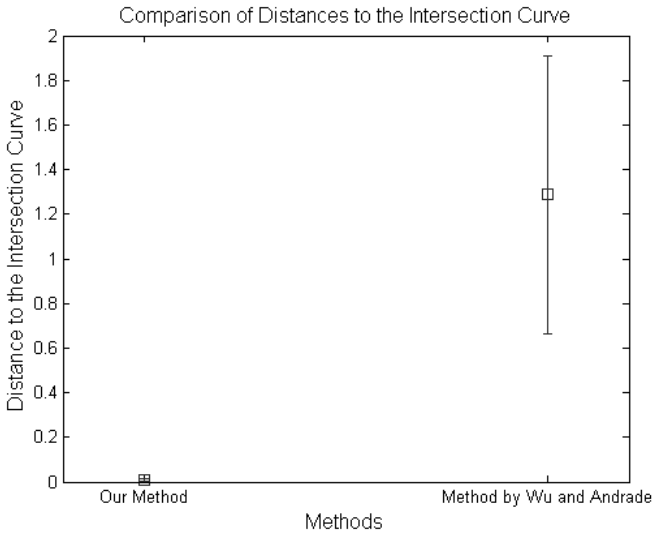


Figure 13: Comparision of distances to the intesection curve.

7 Conclusion

In this study, we presented a novel dual quaternion-based osculating circle DQOC algorithm for finding the intersection curve points of regular two surfaces based on the differential geometric properties of the curve and the use of dual quaternions. We obtained closer points to the intersection curve when we compare our method with Wu and Andrade's [15] method. Our proposed method is more efficient and accurate when we compare our method with Wu and Andrade's method for different parallel threshold values. Also, we used fewer calculations for this process. Moreover, we gave the natural equations of the intersection curve.

References

[1] C. Asteasu, Intersection of arbitrary surfaces. *Computer Aided Design*, **20** (1988) 533–538. ⇒304

[2] R. E. Barnhill, S. N. Kersey, A marching method for parametric surface/surface intersection. *Computer Aided Geometric Design*, **7** (1990) 257–280. ⇒304

[3] W. Clifford, Preliminary sketch of bi-quaternions. *Proc. London Math. Soc.*, **4,** 1 (1873) 381–395. ⇒305

[4] M. Do Carmo, *Differential geometry of curves and surfaces.* Prentice-Hall, New Jersey, 1976. ⇒307
[5] W. R. Hamilton, On quaternions; or a new system of imaginaries in algebra.v*Phil. Mag. J.* **25** (1844) 489–495. ⇒305
[6] W. R. Hamilton, *Elements of quaternions, Vol. I and II.* Chelsea Publishing Company, New York, 1869. ⇒305
[7] B. Kenwright A beginners guide to dual-quaternions: What they are, how they work and how to use them for 3d character hierarchies. *In The 20th International Conference on Computer Graphics, Visualization and Computer Vision,* 2012. ⇒304, 306
[8] R. E. Lynch, C. L. Bajaj, C. M. Hoffman, J. E. H. Hopcroft, Tracing surface intersections. *Computer Aided Geometric Design*, **5** (1988) 285–307. ⇒303
[9] M. E. Mortenson, *Geometric Modeling.* Wiley, USA, 1 st ed. 1985. ⇒304
[10] N. M. Patrikalakis, Surface-to-surface intersections. *Computer Graphics and Applications, IEEE*, **13,** 1 (1993) 89–95. ⇒303
[11] E. Salamin, Application of quaternions to computation with rotations. *Technical Report, Stanford Univ*, 1979. ⇒305
[12] Tz. E. Stoyanov, Marching along surface/surface intersection curves with an adaptive step length. *Computer Aided Geometric Design*, **9** (1992) 485–489. ⇒304
[13] G. R. Veldkamp On the use of dual numbers, vectors, and matrices in instantaneous spatial kinematics. *Mech. Mach. Theory*, **11** (1976) 141–156. ⇒304
[14] M. Ventura, C. G. Soares, Surface intersection in geometric modeling of ship hulls. *Journal of Marine Science and Technology*, **17,** 1 (2012) 114–124. ⇒307
[15] S. T. Wu, L. Andrade, Marching along a regular surface/surface intersection with circular steps. *Computer Aided Geometric Design*, **16,** 4 (1999) 249–268. ⇒304, 307, 308, 309, 314, 316, 317, 319, 321, 322
[16] A. T. Yang, *Application of Quaternion Algebra and Dual numbers to the Analysis of Spatial Mechanisms.* PhD thesis, Columbia University, 1963. ⇒304
[17] X. Ye, T. Maekawa Differential geometry of intersection curves of two surfaces. *Computer Aided Geometric Design*, **16** (1999) 767–788. ⇒307
[18] J. Zara, L. Kavan, S. Collins, C. O'Sullivan, Skinning with dual quaternions. *In Proceedgings of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games,* 2007. ⇒306

Received: November 3, 2021 • Revised: November 13, 2021

Acta Univ. Sapientiae, Informatica **13,** 2 (2021) 324–334

DOI: 10.2478/ausi-2021-0015

Vertex stress related parameters for certain Kneser graphs

Johan KOK
Independent Mathematics Researcher, City of Tshwane, South Africa & Visiting Faculty at CHRIST (Deemed to be a University), Bangalore, India
email: jacotype@gmail.com & johan.kok@christuniversity.in

Abstract. This paper presents results for some vertex stress related parameters in respect of specific subfamilies of Kneser graphs. Kneser graphs for which $\mathtt{diam}(\mathsf{KG}(n,k)) = 2$ and $k \geq 2$ are considered. The note establishes the foundation for researching similar results for Kneser graphs for which $\mathtt{diam}(\mathsf{KG}(n,k)) \geq 3$. In addition some important vertex stress related properties are stated. Finally some results for specific bipartite Kneser graphs i.e. $\mathsf{BK}(n,1)$, $n \geq 3$ will be presented. In the conclusion some worthy research avenues are proposed.

1 Introduction

It is assumed that the reader has good working knowledge of set theory. For the general notation, notions and important introductory results in set theory, see [3]. For the general notation, notions and important introductory results in graph theory, see [2, 4].

Only non-trivial, finite, undirected and connected simple graphs are considered. Let X_i, $i = 1, 2, 3, \ldots, \binom{n}{k}$, $k \geq 1$ be the k-element subsets of the set,

Computing Classification System 1998: G.2.2
Mathematics Subject Classification 2010: 05C12, 05C30, 05C69
Key words and phrases: vertex stress, diameter, distance, Kneser graph.

$\{1, 2, 3, \ldots, n\}$. A Kneser graph denoted by $KG(n, k)$, $n, k \in \mathbb{N}$ is a graph with vertex set,

$$V(KG(n, k)) = \{v_i : v_i \mapsto X_i\}$$

and the edge set,

$$E(KG(n, k)) = \{v_i v_j : X_i \cap X_j = \emptyset\}.$$

Without any relation between n and k the following subfamilies of Kneser graphs follow easily.

1. For $k > n$ the Kneser graph has an empty vertex set implying that the edge set is empty. Hence, for $n \in \mathbb{N}$ the empty graph is obtained.
2. For $k = 1$ the Kneser graph $KG(n, k) \cong K_n \ \forall\ n$.
3. For $k = n$ the Kneser graph is always a trivial graph (i.e. K_1).
4. For $\frac{n}{2} < k < n$ the vertex set is non-empty. However, the edge set is empty so for the permissible values of k and n the corresponding null graphs $\mathfrak{N}_t$, $t = \binom{n}{k}$ are obtained.
5. For n even and $k = \frac{n}{2}$ a corresponding matching graph is obtained.
6. For n even and $2 \leq k \leq \frac{n-2}{2}$ and; for n odd and $2 \leq k \leq \lfloor \frac{n}{2} \rfloor$ the subfamily of non-trivial, connected and non-complete Kneser graphs is obtained.

Note that the author draws a distinction between an empty graph G, (both $V(G) = \emptyset$ and $E(G) = \emptyset$) and a null graph H, ($V(H) \neq \emptyset$ and $E(H) = \emptyset$). This distinction is not common in the literature. Conventionally, the 0-element subset is not considered. However, from set theory it is known that the empty-set is indeed a subset of any set. Therefore, it seems proper to state that $KG(n, 0)$ is a trivial graph say, $v_1 \mapsto \emptyset$.

It follows directly from the structure of a Kneser graph that the order of a Kneser graph is given by $\nu(KG(n, k)) = \binom{n}{k}$. Furthermore, because vertex adjacency as it relates to a k-element subset is defined without loss of generality, a Kneser graph is inherently a degree regular graph. The number of neighbors of any vertex v_i is given by, $\deg(v_i) = \binom{n-k}{k}$. From the aforesaid it follows that the number of edges is given by, $\varepsilon(KG(n, k)) = \frac{1}{2} \times \frac{n!}{k!k!(n-2k)!}$.

Example: $KG(5, 2)$: Let $V(KG(5, 2))$ be define as: $v_1 \mapsto \{1, 2\}$, $v_2 \mapsto \{1, 3\}$, $v_3 \mapsto \{1, 4\}$, $v_4 \mapsto \{1, 5\}$, $v_5 \mapsto \{2, 3\}$, $v_6 \mapsto \{2, 4\}$, $v_7 \mapsto \{2, 5\}$, $v_8 \mapsto \{3, 4\}$, $v_9 \mapsto \{3, 5\}$, $v_{10} \mapsto \{4, 5\}$. See figure 1 as illustration. It is known that, $KG(5, 2) \cong$ *Petersen graph.*

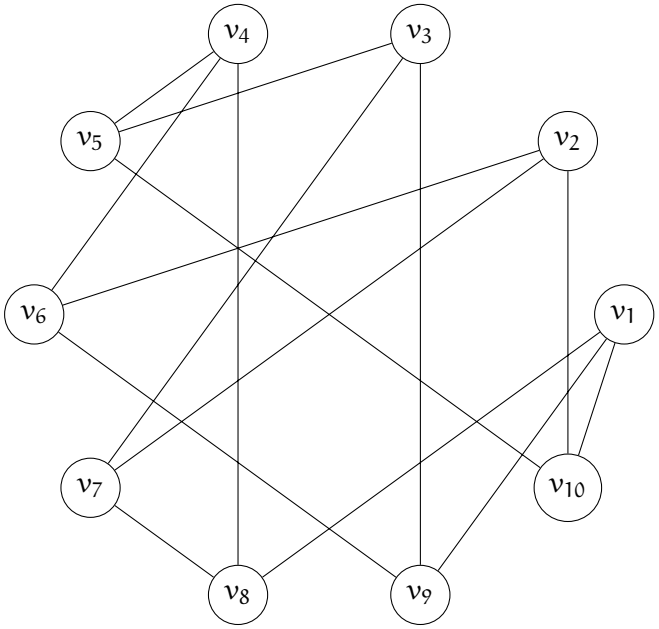


Figure 1: $KG(5,2)$ of order 10.

2 Total induced vertex stress, total vertex stress and vertex stress

The vertex stress of vertex $v \in V(G)$ is the number of times v is contained as an internal vertex in all shortest paths between all pairs of distinct vertices in $V(G)\backslash\{v\}$. Formally stated, $\mathcal{S}_G(v) = \sum_{u \neq w \neq v \neq u} \sigma(v)$ with $\sigma(v)$ the number of shortest paths between vertices u, w which contain v as an internal vertex. Such a shortest uw-path is also called a uw-distance path. See [8, 9]. The *total vertex stress* of G is given by $\mathcal{S}(G) = \sum_{v \in V(G)} \mathcal{S}_G(v)$, [5]. From [10, 11] we recall the definition of total induced vertex stress denoted by, $\mathfrak{s}_G(v_i)$, $v_i \in V(G)$.

Definition 1 [11] *Let* $V(G) = \{v_i : 1 \leq i \leq n\}$. *For the ordered vertex pair* (v_i, v_j) *let there be* $k_G(i,j)$ *distinct shortest paths of length* $l_G(i,j)$ *from* v_i *to* v_j. *Then,* $\mathfrak{s}_G(v_i) = \sum_{j=1, j\neq i}^{n} k_G(i,j)(\ell_G(i,j) - 1)$.

The notion of vertex stress finds application in research related to *centrality* in graphs. In dynamical graph theory the parameter assists to identify vertices

which are more prone to system failure. The nodes within road networks are a good example. A more subtle example is identifying the possibility of a high number of step-through certain steps in an algorithm. Such high number step-through may lead to excessive memory requirements. Highly congested airports can be pre-empted by determining the vertex stress of airports serving as the nodes of a flight route network.

The families no. 1 to 5 have the vertex stress related parameters equal to 0. The only interesting family of Kneser graphs to consider with regards to vertex stress and related parameters is family no. 6. Thus only Kneser graphs within the range $2 \leq k < \frac{n}{2}$ will be studied. From [12] we have that $\mathtt{diam}(KG(n,k)) = 1+\lceil \frac{k-1}{n-2k} \rceil$. Hence, for $2 \leq k \leq \frac{n+1}{3}$ a corresponding Kneser graph has $\mathtt{diam}(KG(n,k)) = 2$. This section seeks to find results for $k \geq 2$ subject to, $n \geq 3k-1$.

Case 1: Let $k = 2$, then $n \geq 5$. For ease of reasoning the following convention for 2-subsets of the set $\{1,2,3,\ldots,n\}$ will be used. The vertices are defined as: $v_1 \mapsto \{1,2\}$, $v_2 \mapsto \{1,3\},\ldots,v_{n-1} \mapsto \{1,n\}$, $v_n \mapsto \{2,3\}$, $v_{n+1} \mapsto \{2,4\},\ldots,v_{2n-3} \mapsto \{2,n\}$, $\ldots$, $v_{\binom{n}{2}} \mapsto \{n-1,n\}$.

Remark. As stated before, since vertex adjacency is defined without loss of generality (for brevity, the *wlg*-principle) all results in respect of vertex v_1 are (immediately) valid for all $v_i \in V(KG(n,k))$. Hence, for ease of reasoning the results for v_1 will be determined and then generalized. Such generalization is axiomatically valid and requires no further proof.

Recall that the set of vertices adjacent to vertex v_i is called the open neighborhood of v_i and it is denoted by $N(v_i)$ (or $N_G(v_i)$ if reference to G is important). The closed neighborhood of vertex v_i is defined as, $N[v_i] = N(v_i) \cup \{v_i\}$ (or $N_G[v_i]$ if reference to G is important).

Proposition 2 *For a Kneser graph* $KG(n,2)$, $n \geq 5$ *the total induced vertex stress of* v_1 *is given by*

$$\mathfrak{s}_{KG(n,2)}(v_1) = \left[\binom{n}{2} - \left(\binom{n-2}{2}+1\right)\right] \times \binom{n-3}{2}.$$

Proof. Clearly, $N(v_1) = \{v_i : 1,2 \notin v_i\}$. It is known that $|N(v_1)| = \binom{n-2}{2}$. Because $\mathtt{diam}(KG(n,k)) = 2$ there does not exist any shortest 3-path in $KG(n,2)$. Hence, there only exist shortest 2-paths from v_1 to the remaining $\binom{n}{2}-\left(\binom{n-2}{2}+1\right)$ vertices of $KG(n,2)$. Without loss of generality consider vertex $v_2 \mapsto \{1,3\}$. Since $N(v_2) = \{v_j : 1,3 \notin v_j\}$ it follows easily that $|N(v_1) \cap N(v_2)| = \binom{n-3}{2}$. The aforesaid is true because the elements $1,2,3$ are (must be) excluded

as elements of the 2-element subsets in $N(v_1) \cap N(v_2)$. Thus, the *partial total vertex stress* induced by vertex v_1 along all shortest v_1v_2-paths is settled. By the *wlg*-principle the principle of immediate induction is valid. Therefore,

$$\mathfrak{s}_{KG(n,2)}(v_1) = \left[\binom{n}{2} - \left(\binom{n-2}{2} + 1\right)\right] \times \binom{n-3}{2}. \qquad \square$$

The generalized corollaries follow immediately.

Corollary 3 *For a Kneser graph* $KG(n,2)$, $n \geq 5$ *the total induced vertex stress of* $v_i \in V(KG(n,2))$ *is given by,*

$$\mathfrak{s}_{KG(n,2)}(v_i) = \left[\binom{n}{2} - \left(\binom{n-2}{2} + 1\right)\right] \times \binom{n-3}{2}.$$

Corollary 4 *For a Kneser graph* $KG(n,2)$, $n \geq 5$ *the total vertex stress is given by*

$$\mathcal{S}(KG(n,2)) = \tfrac{1}{2}\binom{n}{2}\left[\binom{n}{2} - \left(\binom{n-2}{2} + 1\right)\right] \times \binom{n-3}{2}.$$

Proof. The result follows from Definition 1 read together with the proof of Proposition 2 and Corollary 3. $\square$

Corollary 5 *For a Kneser graph* $KG(n,2)$, $n \geq 5$ *the vertex stress is given by*

$$\mathcal{S}_{KG(n,2)}(v_i) = \tfrac{1}{2}\left[\binom{n}{2} - \left(\binom{n-2}{2} + 1\right)\right] \times \binom{n-3}{2}.$$

Proof. It is known that the Kneser graphs $KG(n,2)$ are distance regular, see [1]. By Theorem 3.6 in [9] the Kneser graphs $KG(n,2)$ are stress regular as well. Thus, the result of Corollary 4 must simply be divided by $\binom{n}{2}$. $\square$

Note that since $2 \leq k \leq \frac{n+1}{3}$ it follows that $n \geq 3k_1 - 1$ for a $k_1 \in \mathbb{N}\backslash\{1\}$ to ensure that $\mathtt{diam}(KG(n,k_1)) = 2$. This observation enables immediate generalizations. The vertices which may be used for reasoning of proof are:

$$v_1 \mapsto \{1,2,3,\ldots,k_1-1,k_1\},\ v_2 \mapsto \{1,2,3,\ldots,k_1-1,k_1+1\}$$
$$\text{and } v_i \in N(v_1) \cap N(v_2).$$

Because the reasoning of proof is similar to that found in Proposition 2 and Corollaries 3 to 5 and the fact that the *wlg*-principle applies throughout in all Kneser graph embodiments, no further proofs are presented.

Theorem 6 *For a Kneser graph* $KG(n,k_1)$, $k_1 \in \mathbb{N}\backslash\{1,2\}$, $n \geq 3k_1 - 1$ *the total induced vertex stress of* $v_i \in V(KG(n,k_1))$ *is given by*

$$\mathfrak{s}_{KG(n,k_1)}(v_i) = \left[\binom{n}{k_1} - \left(\binom{n-k_1}{k_1} + 1\right)\right] \times \binom{n-(k_1+1)}{k_1}.$$

Corollary 7 *For a Kneser graph* $KG(n,k_1)$, $k_1 \in \mathbb{N}\backslash\{1,2\}$, $n \geq 3k_1 - 1$ *the total vertex stress is given by*

$$\mathcal{S}(KG(n,k_1)) = \tfrac{1}{2}\binom{n}{k_1}\left[\binom{n}{k_1} - \left(\binom{n-k_1}{k_1} + 1\right)\right] \times \binom{n-(k_1+1)}{k_1}.$$

Corollary 8 *For a Kneser graph* $KG(n,k_1)$, $k_1 \in \mathbb{N}\backslash\{1,2\}$, $n \geq 3k_1 - 1$ *the vertex stress is given by,*

$$\mathcal{S}_{KG(n,k_1)}(v_i) = \tfrac{1}{2}\left[\binom{n}{k_1} - \left(\binom{n-k_1}{k_1} + 1\right)\right] \times \binom{n-(k_1+1)}{k_1}.$$

2.1 Vertex stress related properties of $KG(n,2)$

Recall some results from [9]. A graph G for which $\mathcal{S}_G(v_i) = \mathcal{S}_G(v_j)$ for all distinct pairs $v_i, v_j \in V(G)$ is said to be stress regular.

Theorem 9 [9] *Every distance regular graph is stress regular.*

Corollary 10 [9] *Every strongly regular graph is stress regular.*

Corollary 11 [9] *Every distance transitive graph is stress regular.*

Since it is known that the family of Kneser graphs $KG(n,2)$ are distance regular graphs it follows from Theorem 9 that the Kneser graphs $KG(n,2)$ are stress regular. Furthermore, it is known from [1] that every distance regular graph G with $diam(G) = 2$, is strongly regular. The aforesaid read together with Corollary 10 permit the next corollary without further proof.

Corollary 12 *Kneser graphs* $KG(n,k_1)$, $k_1 \in \mathbb{N}\backslash\{1,2\}$, $n \geq 3k_1 - 1$ *are stress regular.*

In fact, a general result (without further proof) is permitted from the knowledge that all Kneser graphs $KG(n,k)$, $n \geq k$ are vertex transitive.

Theorem 13 *All Kneser graphs* $KG(n,k)$, $n \geq k$ *are stress regular.*

2.2 Stress balanced graphs

Definition 14 *A graph* G *is said to be stress-balanced if and only if*

$$\sum_{v_t \in N[v_i]} \mathcal{S}_G(v_t) = \sum_{v_l \in N[v_j]} \mathcal{S}_G(v_l)$$

for all pairs of distinct vertices $v_i, v_j \in V(G)$.

The value $\gamma(v_i) = \sum_{v_t \in N[v_i]} \mathcal{S}_G(v_t)$ is called the *vertex stress index* of the vertex v_i. A star graph (for brevity, a star) is a tree which has a central vertex v_0 with $m \geq 0$ pendent vertices (or leafs) attached to v_0. The star is denoted by $S_{1,m}$ and the leafs are labeled, v_i, $i = 1, 2, 3, \ldots, m$. It is straightforward to verify that the respective vertex stress are, $\mathcal{S}_{S_{1,m}}(v_i) = 0$, $1 \leq i \leq m$ and $\mathcal{S}_{S_{1,m}}(v_0) = \frac{m(m-1)}{2}$. Since, $\sum_{v_j \in N_{S_{1,m}}[v_0]} \mathcal{S}_{S_{1,m}}(v_j) = \frac{m(m-1)}{2} + m \times 0 = \frac{m(m-1)}{2}$ and $\sum_{v_j \in N_{S_{1,m}}[v_i]} \mathcal{S}_{S_{1,m}}(v_j) = 0 + \frac{m(m-1)}{2} = \frac{m(m-1)}{2}$, $1 \leq i \leq m$ a star is stress-balanced. A star shows that, despite not being degree regular or stress regular, a star is stress-balanced. Figure 2 depicts another example. The graph $G = C_4 + v_1v_3$ is not degree regular and has $\mathcal{S}_G(v_1) = \mathcal{S}_G(v_3) = 1$ and $\mathcal{S}_G(v_2) = \mathcal{S}_G(v_4) = 0$. So G is not stress regular but it is stress-balanced.

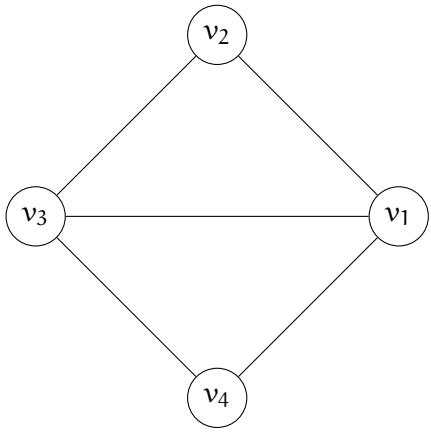


Figure 2: $G = C_4 + v_1v_3$.

Lemma 15 *A graph G which is degree regular (or regular for brevity) and stress regular is stress-balanced.*

Proof. The result is a direct consequence of Definition 14. □

We present the main result of this subsection.

Theorem 16 *All Kneser graphs $KG(n, k)$, $n \geq k$ are stress-balanced.*

Proof. Since all Kneser graphs $KG(n, k)$, $n \geq k$ are degree regular and stress regular (see Theorem 13), read together with Definition 14 and Lemma 15, settles the result. □

3 On bipartite Kneser graphs, $BK(n,k)$

Without loss of generality let $n \geq 3$ and let $1 \leq k \leq \lceil \frac{n}{2} \rceil - 1$. Let X_i, $i = 1,2,3,\ldots,\binom{n}{k}$ be the k-element subsets of the set, $\{1,2,3,\ldots,n\}$. Let Y_i, $i = 1,2,3,\ldots,\binom{n}{k}$ be the $(n-k)$-element subsets of the set, $\{1,2,3,\ldots,n\}$. Let $V_1 = \{v_i : v_i \mapsto X_i\}$ and $V_2 = \{u_i : u_i \mapsto Y_i\}$. A connected bipartite Kneser graph denoted by $BK(n,k)$ is a graph with vertex set,

$$V(BG(n,k)) = V_1 \cup V_2$$

and the edge set,

$$E(BG(n,k)) = \{v_i u_j : X_i \subset Y_j\}.$$

From the definition it is axiomatically true (or from applying the *wlg*-principle) that $BK(n,k)$ is degree regular with $\deg(v_i) = \deg(u_j) = \binom{n-k}{k}$. Equally straightforward is that $BK(n,k)$ is of order $2 \times \binom{n}{k}$. In fact, $|V_1| = |V_2| = \binom{n}{k}$.

Theorem 17 *A bipartite Kneser graph* $BK(n,k)$ *has* $\mathrm{diam}(BK(n,k)) \geq 3$.

Proof. Since $\binom{n-k}{k} < \binom{n}{k}$ it follows that $N(v_i) \subset V_2$ and similarly, $N(u_j) \subset V_1$. Hence, there exists at least one shortest $v_i u_j$-path (or shortest $u_j v_i$-path) of distance greater or equal to 3. □

Similar to the notion of stress regularity it is said that a graph G is *induced vertex stress regular* (for brevity, IVS-regular) if and only if $\mathfrak{s}_G(v_i) = \mathfrak{s}_G(v_j)$ for all distinct pairs $v_i, v_j \in V(G)$.

Theorem 18 *An* IVS*-regular graph* G *is stress regular.*

Proof. The result follows from the fact that for any vertex v_i the vertex stress is given by, $\mathcal{S}_G(v_i) = \frac{\mathfrak{s}_G(v_i)}{2}$. □

Corollary 19 *An* IVS*-regular graph* G *with a singular adjacency regime for all vertices is stress-balanced.*

Proof. The result follows from Theorem 18 and the fact that $\deg_G(v_i) = \deg_G(v_j)$ for all distinct pairs $v_i, v_j \in V(G)$. □

Theorem 18 and Corollary 19 read together with the definition of $BK(n,k)$ imply that bipartite Kneser graphs are stress-balanced.

3.1 Specific results for $BK(n,1)$, $n \geq 3$

Theorem 20 *Bipartite Kneser graphs,* $BK(n,1)$, $n \geq 3$ *are stress regular.*

Proof. Proposition 3.4 in [7] convinces that a graph $BK(n,1)$ is distance-transitive. Also, *distance-transitive* $\Rightarrow$ *distance regular*. Therefore, the latter read together with Theorem 9 (Theorem 3.6 in [9]) settles the fact that bipartite Kneser graphs $BK(n,1)$, $n \geq 3$ are stress regular. □

Theorem 21 *A bipartite Kneser graph,* $BK(n,1)$, $n \geq 3$ *has,*

$$diam(BK(n,1)) = 3.$$

Proof. Let $V_1(BK(3,1)) = \{v_i \mapsto \{i\} : i = 1,2,3$ and $V_2(BK(n,1)) = \{u_1 \mapsto \{1,2\}, u_2 \mapsto \{1,3\}, u_3 \mapsto \{2,3\}\}$. From the definition of $BK(n,1)$ it follows immediately that $BK(3,1) \cong C_6$. Hence, $diam(BK(3,1)) = 3$. For $B(4,1)$ each vertex $u_i \in V_2(B(3,1))$ becomes $u_i \cup \{4\}$ and exactly two vertices are added. Therefore, $V_1(BK(4,1)) = V_1(BK(3,1)) \cup \{4\}$ and $V_2(BK(4,1)) = \{u_i \cup \{4\} : u_i \in V_2(BK(3,1))\} \cup \{1,2,3\}$. After adding the edges in accordance with the adjacency definition it follows easily that, $diam(BK(4,1)) = 3$. Obviously the vertex changes and the addition of exactly two new vertices remain consistent as n progresses consecutively through $5, 6, 7 \ldots$.

Assume the result holds for $BK(n,1)$, $5 \leq n \leq k$. By similar reasoning to show the result for the progression from $n = 3$ to $n = 4$, it follows by immediate induction that the results holds for the progression from $n = k$ to $n = k+1$. Thus,

$$BK(n,1),\ n \geq 3 \text{ has } diam(BK(n,1)) = 3.$$ □

Proposition 22 *A vertex* $v_i \in V_1(BK(n,1))$ *(or* $u_j \in V_2(BK(n,1))$*) has:*

$$s_{BK(n,1)}(v_i) = 3 \times deg_{BK(n,1)}(v_i)(deg_{BK(n,1)}(v_i) - 1).$$

Proof. It follows from Theorem 21 that a vertex $v_i \in V_1(BK(n,1))$ (or $u_j \in V_2(BK(n,1))$) has exactly,

$$deg_{BK(n,1)}(v_i)(deg_{BK(n,1)}(v_i) - 1) \text{ shortest 2-paths}$$

and exactly,

$$deg_{BK(n,1)}(v_i)(deg_{BK(n,1)}(v_i) - 1) \text{ shortest 3-paths.}$$

Obviously v_1 has $deg_{BK(n,1)}(v_i)$ shortests 1-paths (or edges). Applying Definition 1 settles the result. □

Corollary 23 *(a) A vertex* $v_i \in V_1(BK(n,1))$ *(or* $u_j \in V_2(BK(n,1))$*) has:*

$$\mathcal{S}_{BK(n,1)}(v_i) = \tfrac{3(n-1)(n-2)}{2}.$$

(b) $BK(n,1)$, $n \geq 3$ *has,* $\mathcal{S}(BK(n,1)) = 3n(n-1)(n-2)$.
(c) $BK(n,1)$, $n \geq 3$ *has,* $\mathfrak{s}(BK(n,1)) = 6n(n-1)(n-2)$.

Proof. Trivial from the appropriate definitions. □

4 Conclusion

Author is of the view that an extension of this paper through a study of Kneser graphs with diameter greater than 2 is a worthy endeavor. To ensure that say, $\mathtt{diam}(KG(n,k)) = 3$ it follows that for $k \geq 3$, $n \geq \frac{5k-1}{2}$. The number of vertices (k-subsets) of $\binom{\geq \frac{5k-1}{2}}{k}$ becomes large very rapidly. More so for $\mathtt{diam}(KG(n,k)) = \ell$, $\ell \geq 4$. It is suggested that shortest path algorithms (existing or newly developed) or experimental mathematics through simulation programs be utilized in support of further research.

The new notion of stress-balanced graphs has only been briefly introduced. It is suggested to be an interesting concept with a wide scope for further research. A graph G which is not stress-balanced will have at least one vertex say, v_i such that $\gamma(v_i) = \mathtt{max}\{\gamma(v_j) : \gamma(v_j) = \sum_{v_t \in N[v_j]} \mathcal{S}_G(v_t)\}$. A closed neighborhood $N[v_i]$ which yields such maximum is called a *stress district* of G. Similarly a closed neighborhood $N[v_j]$ which yields $\mathtt{min}\{\gamma(v_k) : \gamma(v_k) = \sum_{v_t \in N[v_k]} \mathcal{S}_G(v_t)\}$ is called a *stress suburb* of G. Studying stress districts and stress suburbs remains open.

Various studies of other families of graphs which are constructed from the subsets of a set together with a well-defined adjacency regime have been published. We refer to this as the study of graphs from sets. A specific and perhaps less known family called set-graphs can be read in [6]. Hence, various research projects under the theme *Vertex stress related parameters for graphs from sets* remain open.

Acknowledgment

The author would like to thank the anonymous referee for his(her) constructive comments, which helped to improve the elegance and the content enhancement (bipartite Kneser graphs) of this paper.

References

[1] N. Biggs, *Algebraic Graph Theory 2nd Edition*, Cambridge University Press, Cambridge, 1993, ISBN-13: 978-0521458979. ⇒328, 329

[2] J.A. Bondy, U.S.R. Murty, *Graph Theory with Applications*, Macmillan Press, London, 1976. ⇒324

[3] R.C. Freiwald, *An Introduction to Set Theory and Topology*, Washington State University, Saint. Louis, (2014), http://doi.org/10.7936/K7D798QH ⇒324

[4] F. Harary, *Graph Theory*, Addison-Wesley, Reading MA, 1969. ⇒324

[5] J. Kok, J. Shiny, V. Ajitha, Total vertex stress alteration in cycle related graphs, *Matematichki Bilten*, **44,** 2 (2020) 149–162. http://doi.org/10.37560/matbil2020149k ⇒326

[6] J. Kok, K. P. Chithra, N.K. Sudev, C. Susanth, A study on set-graphs, *International Journal of Computer Applications*, **118,** 7, (2015) 1–5. http://doi.org/10.5120/20754-3173 ⇒333

[7] S. M. Mirafzal, A. Zafari, Some algebraic properties of bipartite Kneser graphs, *arXiv: 1804.04570V1*, (2018) 1–11, to appear in *Ars Combinatoria*. ⇒332

[8] A. Shimbel, Structural parameters of communication networks, *The Bulletin of Mathematical Biophysics*, **15,** 4 (1953) 501–507. ⇒326

[9] J. Shiny, V. Ajitha, Stress regular graphs, *Malaya Journal of Matematik*, **8,** 3 (2020) 1152–1154. http://doi.org/10.26637/MJM0803/0072 ⇒326, 328, 329, 332

[10] J. Shiny, Induced stress of some graph operations, *Malaya Journal of Matematik*, **9**(1), (2021), 259-261. https://doi.org/10.26637/MJM0901/0043 ⇒326

[11] J. Shiny, J. Kok, V. Ajitha, Total induced vertex stress in barbell-like graphs, *Journal of the Indonesian Mathematical Society*, **27**(2), (2021), 150-157. ⇒326

[12] M. Valencia-Pabon and J-C. Vera, On the diameter of Kneser graphs, *Discrete Mathematics*, **305** (1-3), (2005), 383-385. ⇒327

Received: November 14, 2021 • Revised: November 27, 2021

Acta Univ. Sapientiae, Informatica **13,** 2 (2021) 335–360

DOI: 10.2478/ausi-2021-0016

Reproducibility in the technical debt domain

Kristóf SZABADOS
Eötvös Loránd University,
Budapest, Hungary
email: SzabadosKristf@gmail.com

Izabella Ingrid FARKAS
Eötvös Loránd University,
Budapest, Hungary
email: Ingrid.Farkas@inf.elte.hu

Attila KOVÁCS
Eötvös Loránd University,
Budapest, Hungary
email: Attila.Kovacs@inf.elte.hu

Abstract.

Context: It is crucial to understand how reproducible the measurement results in the scientific publications are, as reproducibility is one of the cornerstones of engineering.

Objective: The goal of this study is to investigate the scientific publications presented at the premier technical debt conferences by understanding how reproducible the reported findings are.

Method: We conducted a systematic literature review of 135 unique papers published at the "International Workshop on Managing Technical Debt" and the "International Conference on Managing Technical Debt", the premier scientific conference series on technical debt.

Results: Only 44 of the investigated 135 papers presented numerical evidence and only 5 papers listed the tools, the availability of the tools, and the version of the tools used. For the rest of the papers additional information would have been needed for the potential reproducibility. One of the published papers even referred to a pornographic site as a source of a toolset for empirical research.

Computing Classification System 1998: D.2.3, D.2.4, D.2.5, D.2.7, D.2.8, D.2.9,
Mathematics Subject Classification 2010: 68N30
Key words and phrases: technical debt, reproducibility, literature review, software quality, research reporting quality

Conclusions: The field of technical debt research might have a reproducibility crisis as only approx. 32% of the papers published at the most prestigious workshop/conference series of the field present measurement results, of which only approx. 11% (4% of all papers investigated) identify the tools used, to make reproduction possible. Our findings might even point out to a bias in the field: researchers of this domain pursue Technical Debts in order to improve the quality of software products, assuming that the software products used for measurement are flawless and no longer possible to improve upon, which is a contradiction in and of itself.

1 Introduction

Reproducibility is one of the cornerstones of engineering science. Results, that can not be consistently reproduced (within the same boundary conditions and field-specific accepted divergence) by independent or groups of researchers, using different measurement tools, can not be reliably built upon to extend scientific knowledge. Such scientific publications can even erode the general public's trust in scientific claims.

Reproducing results with different measurement instruments was already shown to be hard in the technical debt domain research. Results of different tools claiming to measure technical debt might not be statistically correlated ([48]), might even use different terms and metrics ([10]). Even when using the same tool, the exact version used for measurements might matter.

On one hand, researchers and tool vendors are actively working on identifying new ways and methods for software product quality improvement, for example by identifying constructs that might lead to faults ([58]). Detecting such new constructs may increase the technical debt related measures reported by tools in the future.

On the other hand, the reported data are frequently overestimated ([52]). In case of junior developers it could even be up to 2 - 20 times ([40]). The fault proneness of the found issues (how likely they would be to lead to bugs) might also be unclear ([39], [44]). Improving the theory and precision of the detection algorithms might lead to tools reporting different (maybe drastically smaller) technical debt related measures in the future.

Also, these tools sometimes evolve even with backwards incompatible changes, which can make community extensions obsolete or unable working together with newer versions ([11]). Research including such extensions might produce misleading output.

In this paper we overview the articles published at the domain's premier workshop series "Workshop on Managing Technical Debt" ([14], [63], [64], [65], [66], [67], [68]) and conference series "International Conference on Technical Debt" ([69], [70], [71]), [72]) in order to see how many of them include the measured values:

- The name of all of the tools used for the measurements.
- The public availability of those tools.
- Exact versions of the applied tools.

These information would be indispensable for independent researchers to be able to reproduce the presented results. Please note that we did not require the input data to be publicly available.

We have found that of the 135 articles published in the proceedings of these conferences, only 44 presented any numerical measurement results. Of these 44 articles, only 5 contained all of the information about the used tools that might be needed to reproduce the results. We also found an article, published in 2018 ([8]), referencing a website as a source of "a toolset developed to support empirical software engineering research", which was by 2021 showing pornographic content.

The paper is organised as follows. In Section 2 we present some earlier work related to the subject. Section 3 presents our research methodology. Section 4 presents our findings. Section 5 deals with the validity of our results. Finally, Section 6 summarises our findings and Section 7 offers ideas for further research.

2 Related work

The term *debt*, in a technical context, was first used in 1992 by Cunningham [20] in his experience report to illustrate the difference between a theoretical waterfall development[51] and incremental growth (enabled by object oriented programming at the time). "The traditional waterfall development cycle has endeavored to avoid programming catastrophe by working out a program in detail before programming begins", "we recognize this amounts to preserving the concept of payment up-front and in-full". In incremental growth development can start before all of the requirements are collected, all architecture is completely designed and all details are fixed, but comes with a risk: "Shipping first time code is like going into debt. A little debt speeds development so long as it is paid back promptly with a rewrite. Objects make the cost

of this transaction tolerable". Stating that incremental growth "leads to the most appropriate product in the shortest possible time" and concluding that "The modularity offered by objects and the practice of consolidation make the alternative, incremental growth, both feasible and desirable in the competitive financial software market".

Since 1992 the technical debt research domain has greatly expanded. The first scientific workshop titled "Workshop on Managing Technical Debt" ([14]) was organized already in 2010. Khomyakov et al. [36] found 603 papers published between 2011 and 2017 related to the automated measurement of technical debt. According to Lenarduzzi et al. [38] in 2018 alone (between March 2018 to December 2018) 384 unique papers were published related to technical debt research. By the end of 2021 already 7 workshops have been organized in the series "Workshop on Managing Technical Debt" ([14], [63], [64], [65], [66], [67], [68]) and 4 conferences in the conference series "International Conference on Technical Debt" ([69], [70], [71], [72]).

While the domain expanded with papers reporting valuable results, some of the papers pointed out to some challenges. Barta et al. ([11]) presented how backward incompatible changes were introduced during the evolution of SonarQube's API. They investigated 66 plug-ins listed as community plug-ins on github[1]. They found that 22 were by the time of their investigation obsolete, 5 of the 23 most recently updated plug-ins (at that time) were already incompatible to the point of not working with the newest SonarQube version. Parodi et al. [48] showed that even different static analysis tools (Sonar, FindBugs) claim to measure technical debt, their outputs are not statistically correlated. They conclude that "Static code analysis tools must be thoroughly studied in order to evaluate if they represent meaningful proxies for Technical Debt" Paris et al. [10] studied numerous tools (both commercial and research prototypes) that claimed to measure technical debt. They found among others that "different tools adopt different terms, metrics, and ways to identify and measure technical debt". They also observed fragmentation of communities on discussion forums on the internet around tools: different tools being popular on different web forums. Saarimki et al. [52] investigated the accuracy of the fixing time estimations for technical debt items by SonarQube on 15 projects with 65 student developers. They found that the estimations were only correct for 2 projects, for 9 projects they were overestimated. Lenarduzzi et al. [40] investigated (among others) how long it takes for junior developers to refactor technical debt items detected by SonarQube. They found that the time

[1] `https://github.com/SonarQubeCommunity` (last accessed 2021.10.05)

estimated by SonarQube was always at least 2 times higher than the actual fixing time, in some cases 20 times higher. Marcilio et al. [44] studied the realistic use of SonarQube on 421 976 issues from 246 projects. They found that only approx. 13% of the issues reported by SonarQube were resolved. They conjectured that "just a subset of the checkers reveal real design and coding flaws, and this might artificially increase the technical debt of the systems". Lenarduzzi et al. [39] conducted an empirical study on 21 open-source projects to understand how fault-prone SonarQube's violations are. They found that of the 202 violations defined for Java only 26 have a low fault-proneness, "violations classified as "bug" does not seem to be the root cause of faults", warning that the current way of calculating technical debt is incorrect as several non-fault prone rules are counted as fault-prone, while some other rules should be considered fault-prone. Concluding that "companies should carefully consider which rules they really need to apply".

Even if only looking at the release notes of SonarQube (the tool used most often in the reviewed articles), the tool obviously goes through changes that might impact the measures it reports for a given source code:

- On one hand, the number of rules supported for Java increased from 400+ in version 7.3[2] (released on August 13, 2018) to 550+ by version 8.0[3] (released on October 16th, 2019). More rules, finding more instances of quality issues, potentially lead to higher technical debt values reported for the same source codes.

- On the other hand, version 8.2[4] describes its Java related changes as "Ground-up rewrite brings more accurate analysis, with fewer false positives". More precise detection reporting fewer false positives, potentially leads to lower technical debt values reported for the same source codes.

Other software products used for checking code quality are also not immune to software quality issues, that might affect their measured and reported values. We have already reported some issues to the producers of such products. For example:

- The functionality of PMD, checking if a public function is commented or not, might not correctly detect in some cases that the function has a comment, falsely reporting that the function is undocumented [5].

[2] `https://www.sonarqube.org/sonarqube-7-3/` (last accessed 2021.10.05)
[3] `https://www.sonarqube.org/sonarqube-8-0/` (last accessed 2021.10.05)
[4] `https://www.sonarqube.org/sonarqube-8-2/` (last accessed 2021.10.05)
[5] `https://github.com/pmd/pmd-eclipse-plugin/issues/138` (last accessed 2021.10.05)

- We have found a situation, where the correctness of the code was not evaluated in enough depth in a sub-tool, that lead to false-positive finding reported in the main tool and complexities in the issue reporting procedure. In a particular case[6] it turned out, that:

  1. The root of the problem is not in the given product, but in a different product they use to offer their functionalities.
  2. The developers/maintainers don't have the resources needed to help in handling the issue, or to help report the issue (with additional information) to the correct source.
  3. We, as users, are not well equipped to identify which sub-product embedded in/used by this main-product is the root of the problem. Also, not well equipped to provide the developers of that sub-product with additional information on how their product is called/integrated.

In general, at the time of writing this article there are:

- 729 open bugs for the "Static Analyzer" component of Clang[7]
- 8 066 for Clang itself[8].
- At least 10 000 for GCC[9]
- 519 open issues for PMD[10]
- 386 open issues for SpotBugs[11]
- SonarSource also has an active community[12] with many topic related to shortcomings of some products (like SonarQube itself).

[6] `https://github.com/Ericsson/codechecker/issues/3098` (last accessed 2021.10.05)

[7] `https://bugs.llvm.org/buglist.cgi?bug_status=__open__&component=Static%20Analyzer&limit=0&order=priority%2Cbug_severity&product=clang&query_format=advanced` (last accessed 2021.10.05, needs registered user)

[8] `https://bugs.llvm.org/buglist.cgi?bug_status=__open__&limit=0&no_redirect=1&order=priority%2Cbug_severity&product=clang&query_format=specific` (last accessed 2021.10.05, needs registered user)

[9] `https://gcc.gnu.org/bugzilla/buglist.cgi?bug_status=__open__&limit=0&no_redirect=1&order=priority%2Cbug_severity&product=gcc&query_format=specific` (last accessed 2021.10.05, needs registered user)

[10] `https://github.com/pmd/pmd/issues` (last accessed 2021.10.05)

[11] `https://github.com/spotbugs/spotbugs/issues` (last accessed 2021.10.05)

[12] `https://community.sonarsource.com/tags` (last accessed 2021.10.05)

3 Methodology

In order to understand how reproducible the results published in the technical debt domain research are, we conducted a systematic literature review. In this section, we describe the goal of our research, report our search strategy and paper evaluation method.

3.1 Goal of our research

The goal of the research presented in this paper was to investigate the existing body of knowledge in software engineering to understand how reproducible the academic papers published in the technical debt domain research are.

3.2 Search strategy

The search strategy involves the bibliographic sources, the definition of the inclusion and exclusion criteria.

`Bibliographic sources:` To get a good view of the field of technical debt research, we decided to use the domain's premier workshop series "Workshop on Managing Technical Debt" ([14], [63], [64], [65], [66], [67], [68]) and conference series "International Conference on Technical Debt" ([69], [70], [71]), [72]) as sources. We used the ACM Digital Library[13] and IEEEXplore Digital Library[14] to reach the articles that were published as part of the proceedings of these conferences.

`Inclusion criteria:`

- Papers presenting measured values in relation to technical debt (numerical values that might be reproduced and compared to in a replication study).

`Exclusion criteria:`

- Papers not presenting a measurement result (example: [3], [13]).
- Papers using surveys as input for measurements (example: [12]).

Please note, that we did include papers that performed their measurements on projects that were anonymized in the paper. We believe that the field needs to have information from the industry, which might come with such limitations.

[13] `https://www.acm.org/` (last accessed 2021.10.05)
[14] `https://ieeexplore.ieee.org` (last accessed 2021.10.05)

`Search process:` The search was conducted in March 2021.

Based on the described process, we retrieved a total of 122 papers for the review.

3.3 Paper evaluation

To decide if a result presented in a paper might be reproducible, we tried to identify what tools were described as being used to create the result, and checked how well they were identified:

- Is the name of the tool represented?
- Is there a direct link provided, from where the tool could be accessed (either for free or commercially)?
- Is the version of the tool identified?

During the evaluation of the papers, we only considered those tools, for each paper, that were presented as being used for creating the published result. Tools mentioned in relation to other works or in comparison with the used tools, were not counted as used in the paper.

Please note, that this method comes with a limitation: as we did not actually try to reproduce the published result, we could not detect if the researchers had to use some undisclosed tools to get to the published result.

4 Results

In this section we present our findings. In a number of investigated papers it is not clear what tools the researchers were using. Even if the tool's name resembles a well known tool we could not precisely cite the tool, as it is not clear if they meant the same tool and if so which version of it. For this reason in our short assessments of all investigated papers we list the tool names found in a paper, in apostrophes first. This is done to provide a common presentation format and to indicate that the name might not refer to the tool readers could automatically associate it with.

4.1 Details for 2010 ([14])

According to the Software Engineering Institute of Carnegie Mellon University [17] the results of the first workshop organized for the topic of managing technical debt was published as [14]. This was a single paper, that only introduced the idea and importance of holding international workshops on managing technical debt, but in itself did not disclose measurements.

Table 1: The general result of our Systematic Literature Review

Venue	Nr. of articles	Presents measured results	Fully identifies used tools
MTD 2010 ([14])	1	0	0
MTD 2011 ([63])	9	4	0
MTD 2012 ([64])	11	3	0
MTD 2013a ([65])	11	2	0
MTD 2013b ([27])	1	0	0
MTD 2014 ([66])	9	4	1
MTD 2015 ([67])	12	7	0
MTD 2016 ([68])	7	3	1
TechDebt 2018 ([69])	22	7	0
TechDebt 2019 ([70])	24	3	0
TechDebt 2020 ([71])	15	4	1
TechDebt 2021 ([72])	13	7	2

4.2 Details for: "MTD 2011: Proceedings of the 2nd Workshop on Managing Technical Debt" ([63])

Out of the 9 papers published as part of the 2nd workshop on Managing Technical Debt we have found 4 that disclosed some measurement results. None of which might be reproducible.

The papers that might have reproducibility problems:

- "An Empirical Model of Technical Debt and Interest" ([47]) mentions that they use the "Software Analysis Toolkit" of the "Software Improvement Group", but there is no reference to the tool or a version number disclosed.
- "From Assessment to Reduction: How Cutter Consortium Helps Rein in Millions of Dollars in Technical Debt" ([33]) mentions that "Cutter's technical debt assessment is an automated analysis of code deficits", but the availability or version of the tool is not disclosed. The only referenced document does not seem to be available any longer.
- "An Extraction Method to Collect Data on Defects and Effort Evolution in a Constantly Modified System" ([34]) mentions querying the "Mantis Tool" and contains a reference to it, but does not disclose the version used.

- "Prioritizing Design Debt Investment Opportunities" ([62]) discloses the metrics they measured and the methods they used, but not the tool.

4.3 Details for: "MTD 2012: Proceedings of the Third International Workshop on Managing Technical Debt" ([64])

Out of the 11 papers published as part of the third International Workshop on Managing Technical Debt we have found 3 that disclosed some measurement results, none of which might be reproducible.

The papers that might have reproducibility problems:

- "Estimating the Size, Cost, and Types of Technical Debt" ([21]) uses the "Application Intelligence Platform" of "CAST", without disclosing the version used.
- "Investigating the Impact of Code Smells Debt on Quality Code Evaluation" ([30]) uses several tools:
  - "iPlasma": the referenced web page is not available.
  - "Eclipse Metrics": version used is not disclosed, although according to the referenced page the tool was not updated since 2013.
  - "Google CodePro Analytix": the referenced webpage is not available.
- "What Is the Value of Your Software?" ([24]) discloses taking the data from the "software analysis warehouse" of the "Software Improvement Group", without disclosing a reference to the data, when it was accessed, with which tool version the data was measured.

4.4 Details for: "MTD 2013: Proceedings of the 4th International Workshop on Managing Technical Debt" ([65])

Out of the 11 papers published as part of the 4th International Workshop on Managing Technical Debt we have found 2 that disclosed some measurement results, none of which might be reproducible.

The papers that might have reproducibility problems:

- "Exploring Software Supply Chains From a Technical Debt Perspective" ([45]) uses the tools "Sonar", "Understand" and "Cytoscape". Although their web pages are referenced and their last access time is presented, it is not clear which version of these tools was used for the measurement

(that might be different from the version available on the web page at the last time it was accessed).

- "Generating Precise Dependencies for Large Software" ([61]) uses a tool written by the authors for measurements. There is no link to the tool, not even it's name is disclosed.

4.5 Details for: "MTD 2013: Proceedings of the fifth International Workshop on Managing Technical Debt" ([27])

According to the Software Engineering Institute of Carnegie Mellon University ([17]) the summary of the fifth workshop organized for the topic of managing technical debt was published as [27].

The workshop does not seem to have had a proceedings published. Although the Software Engineering Institute of Carnegie Mellon University ([17]) makes the slides of some of the presentations of the workshops available, we excluded them as not being published articles.

This was a single paper, that summarized the discussions of the workshop, but in itself did not disclose measurements.

4.6 Details for: "MTD 2014: Proceedings of the 2014 Sixth International Workshop on Managing Technical Debt" ([66])

Out of the 9 papers published as part of the sixth International Workshop on Managing Technical Debt we have found 4 that disclosed some measurement results, only 1 also disclosing the version of the tool used.

The papers that might have reproducibility problems:

- "A Framework for Estimating Interest on Technical Debt by Monitoring Developer Activity Related to Code Comprehension" ([55]) uses the tools "Blaze" and "Understand". The version of neither of them is disclosed, also the reference of Blaze points to a paper, not directly to the tool.
- "Are the Methods in Your Data Access Objects (DAOs) in the Right Place?A Preliminary Study" ([9]) uses the tool "calculadora-de-daos". The link of the tool, is either corrupted or points to a source code repository no longer available.
- "The Correspondence between Software Quality Models and Technical Debt Estimation Approaches" ([35]) uses the tools "PMD", "findbugs", "SonarQube", "Understand", "inFusion" all of which identified by their websites only, without disclosing their versions used. In the case of inFusion the website does not seem to be available.

The paper fully identifying the tool used:

- "Explicating, Understanding and Managing Technical Debt from Self-Driving Miniature Car Projects" ([43]) uses the tools "SonarQube", "Sonar-Runner" and "SonarWay" disclosing the version of these tools used for their measurements.

4.7 Details for: "2015 IEEE 7th International Workshop on Managing Technical Debt (MTD 2015)" ([67])

Out of the 12 papers published as part of the 7th International Workshop on Managing Technical Debt we have found 7 that disclosed some measurement results, none of which might be reproducible.

The papers that might have reproducibility problems:

- "A Contextualized Vocabulary Model for Identifying Technical Debt on Code Comments" ([23]) uses the tool "eXcomment" developed by the authors. The tool is only identified by a web link, that no longer works.

- "Detecting and Quantifying Different Types of Self-Admitted Technical Debt" ([42]) uses "jDeodorant" for information extraction and an unnamed tool supporting manual classification. The version of jDeodorant is not disclosed and the reference points to an academic paper, not directly a tool.

- "Estimating the Breaking Point for Technical Debt" ([16]) uses the tools "JCaliper" and "SEAgle". Both of them developed by the authors, neither of them has their version disclosed. SEAgle is identified by a reference to an academic paper, while jCaliper is not identified at all.

- "Technical Debt of Standardized Test Software" ([57]) uses "Titan" for measurement, only identified by a reference to an academic paper.

- "Towards a Prioritization of Code Debt: A CodeSmell Intensity Index" ([31]) uses the tool "JCodeOdor" developed by the authors. The tool is only identified by a reference to an academic article, without a direct reference to the tool, or its version used.

- "Towards an Open-Source Tool for Measuring and Visualizing the Interest of Technical Debt" ([28]) uses the tool "MIND (ManagIng techNical Debt)" built as a plug-in of "SonarQube". Both tools are identified by a link to their web pages, neither of them has their version disclosed.

- "Validating and Prioritizing Quality Rules for Managing Technical Debt: An Industrial Case Study" ([29]) uses the tool "Technical Debt Analyzer" built by the authors as a plug-in for "SonarQube". SonarQube is only identified by a link to its web page, Technical Debt Analyzer is not identified in any way.

4.8 Details for: "2016 IEEE 8th International Workshop on Managing Technical Debt (MTD 2016)" ([68])

Out of the 7 papers published as part of the 8th International Workshop on Managing Technical Debt we have found 3 that disclosed some measurement results, only 1 also disclosing the version of the tool used.

The papers that might have reproducibility problems:

- "Practical Technical Debt Discovery by Matching Patterns in Assessment Graph" ([54]) does not identify the tools used for measurements.
- "The Perception of Technical Debt in the Embedded Systems Domain: An Industrial Case Study" ([7]) does not identify the tool used for measuring maintainability (used as a proxy for technical debt).

The paper fully identifying the tool used:

- "Technical Debt Indexes provided by tools:a preliminary discussion" ([32]) uses the tools "CAST", "inFusion", "Sonargraph", "SonarQube" and "Structure101". Each identified by their respective web pages and the exact version of the tools used.

4.9 Details for: "TechDebt '18: Proceedings of the 2018 International Conference on Technical Debt" ([69])

Out of the 22 papers published as part of the 2018 International Conference on Technical Debt we have found 7 that disclosed some measurement results, none of which might be reproducible.

The papers that might have reproducibility problems:

- "A Framework for Managing Interest in Technical Debt: An Industrial Validation" ([8]) uses the tools "SonarQube", "Percerons Client" and "Breaking Point Calculator".
  - For SonarQube only the documentation is referenced, with a link, not the used version of the tool or the tool itself.

  - Breaking Point Calculator was developed by the authors, is not referenced and does not seem to be available for the community.
  - "Percerons Client" is identified by a link to, what seems to be a website for pornographic movies[15] (see figure 1).

- "An exploratory study on the influence of developers in technical debt" ([4]) uses "SonarQube" and the author's own scripts. SonarQube is only identified by a reference to an academic article, without a direct reference to the tool, or its version used. The author's own scripts are not available.
- "Design Debt Prioritization: A Design Best Practice-Based Approach" ([50]) uses "MUSE" a tool developed by the authors, identified by a reference to an academic article, without a direct reference to the tool, or its version used.
- "Evaluating Domain-Specific Metric Thresholds: An Empirical Study" ([46]) uses the tools "CK Tool" and "TDTool". "CK Tool" is identified with a link to its source code repository, without revealing the version used. TDTool is identified by a reference to an academic article, without a direct reference to the tool, or its version used.
- "From Lasagna to Spaghetti, a Decision Model to Manage Defect Debt" ([2]) uses a tool developed by the authors for measurements, but does not identify the tool or its availability.
- "Prioritize Technical Debt in Large-Scale Systems Using CodeScene" ([60]) uses "CodeScene". The tool itself is only indirectly referenced in a reference of a data set, that is stored on a web page similar to the tool's name. The version used is not identified.
- "The developer's dilemma: factors affecting the decision to repay code debt" ([6]) uses "SonarQube", which is only identified by a reference to the tool's website, without disclosing the version used for measurements.

4.10 Details for: "TechDebt '19: Proceedings of the Second International Conference on Technical Debt" ([70])

Out of the 24 papers published as part of the 2019 International Conference on Technical Debt we have found 3 that disclosed some measurement results, none of which might be reproducible.

[15] Already on 2021.05.18, last checked on 2021.10.05

Figure 1: The pornographic site referenced by Ampatzoglou et al. ([8]) as the source of "a toolset developed to support empirical software engineering research". (last accessed 2021.10.05)

The papers that might have reproducibility problems:

- "Investigating on the Impact of Software Clones on Technical Debt" ([41]) uses "NiCad Clone Detector" and "SonarQube", neither of them being identified in the article.
- "On the Diffuseness of Code Technical Debt in Java Projects of the Apache Ecosystem" ([53]) uses "SonarQube", identified only by a reference to the tool's website, without disclosing the version used for measurements.
- "The Delta Maintainability Model: Measuring Maintainability of Fine-Grained Code Changes" ([25]) proposes and uses a new maintainability model "Delta Maintainability Model" (DMM) for fine-grained measure-

ments of code changes, without making the tool implemented and used for measurements available (or at least it is not referenced in the article).

4.11 Details for: "TechDebt '20: Proceedings of the 3rd International Conference on Technical Debt" ([71])

Out of the 15 papers published as part of the 2020 International Conference on Technical Debt we have found 4 that disclosed some measurement results, only 1 also disclosing the version of the tool used.

The papers that might have reproducibility problems:

- "Detecting Bad Smells with Machine Learning Algorithms: An Empirical Study" ([19]) uses the tools "JDeodorant", "JSpirit", "PMD", "DECOR", "Organic". While PMD is identified with a link to it's open source repository, the other tools are identified by references to academic papers. None of the tools used has it's version disclosed.

- "Towards Microservice Smells Detection" ([49]) uses "Arcan", a tool developed by the authors, identified by a reference to 2 academic articles, without a direct reference to the tool, or its version used.

- "The Hidden Cost of Backward Compatibility: When Deprecation Turns into Technical Debt - an Experience Report" ([56]) uses a unidentified tool developed by the authors of the article.

The paper fully identifying the tool used:

- "An Empirical Study on Self-Fixed Technical Debt" ([59]) uses the tool "SonarQube", disclosing also the version used for their measurements.

4.12 Details for: "TechDebt '21: Proceedings of the 4th International Conference on Technical Debt" ([72])

Out of the 13 papers published as part of the 2021 International Conference on Technical Debt we have found 7 that disclosed some measurement results, only 2 also disclosing the version of all of the tools used.

The papers that might have reproducibility problems:

- "Assessing Smart Contracts Security Technical Debts" ([1]) uses several tools:

  - "Slither", "SmartCheck", "Securify" and "Mythril" are identified by reference to academic articles, without direct references to the tools, or their versions disclosed.
  - "Manticore", "Solhint" and "Ethlint" are identified by links to their GitHub repositories, without disclosing the used version.
  - "Sfuzz" is referenced by an academic article, without disclosing the used version. It is also disclosed, that a web interface was used, not a local installation.

- "Business-Driven Technical Debt Prioritization: An Industrial Case Study" ([22]) uses an undisclosed tool.

- "Predicting Relative Thresholds for Object Oriented Metrics" ([5]) uses a tool developed by the authors for measurements, but does not identify the tool or its availability.

- "Worst Smells and Their Worst Reasons" ([26]) uses "SonarCloud", identified by a reference to the tools website, without disclosing the version used for measurements.

- "Impact of Opportunistic Reuse Practices to Technical Debt" ([15]) uses "SonarQube" and "Arcan". SonarQube is identified by a link to its website and the version used. Arcan is identified by a reference to an academic article, without a direct reference to the tool or its version disclosed.

The paper fully identifying the tool used:

- "Carrot and Stick approaches revisited when managing Technical Debt in an educational context" ([18]) uses the tool "SonarQube", disclosing also the version used for their measurements. (They also use "PostgreSQL" and "MySQL server" without any identification, "Tablon" identified by a link to a web page without dislosing the version used. These tools are not used to measure technical debt.)

- "Experiences on Managing Technical Debt with Code Smells and AntiPatterns" ([37]) uses the tools "CodeMR" and "IntelliJ IDEA code inspection tool". CodeMR is dientified with a link to its website and the version used is disclosed. The "IntelliJ IDEA code inspection tool" also has the used version disclosed.

5 Threats to validity

While we believe that our results are valid in their described context, we have to point out that this is a limited context.

We did not try to reproduce the results published in the papers, and we did not include in our reproducibility criteria the public availability of the projects used as input to the tools. This way, it was not possible to detect potential data manipulation or required tools not being mentioned.

We only reviewed the papers published in the workshop and conference series we believed to be the most prestigious in the field. It might very well be the case that reviewing all publications in the field would find a different number and ratio of reproducible papers.

6 Summary

In this paper we presented our work and findings on the reproducibility of research results in the technical debt research domain, published at the most prestigious workshop and conference series.

We have found that of the 135 papers published in the proceedings of these conferences, only 44 presented any numerical measurements. Of these 44 papers, only 5 contained the information about the used tools that might be needed to reproduce the result.

This might indicate a bias in the field. Researchers working to improve the quality of software products might not be aware that the software products they use for measurements could also need quality improvements.

One of the papers we investigated ([8]) referenced a website showing pornographic content. This situation is problematic for both the publisher of the proceedings (that now indirectly advertises pornographic content) and the field in general (in 2 years after publication, entire toolset can move or disappear).

We also found articles where the measurement tool was not available (either was never made public, or their repository already was deleted).

Our final observation is that the field should use more stringent practices when reviewing and handling research publications to ensure that the published results can be reproduced/replicated (and so safely built upon) by others.

We believe this paper is a step forward to the technical debt researching community. Improving the reproducibility of papers published in this field by

raising the scientific standard could enable more scrutiny towards the published results, making them safer to build on.

We would recommend authors to follow (and reviewers to check) the following guidelines to make their papers more reproducible:

- Disclose in the paper the tools used:
  - Hardware used.
  - Software stack used (operating system etc..)
  - The names, configuration, precise version identification for all of the tools used and a link to a web page from where the tool can be obtained (even if commercially).
  - Governance completeness. When using cloud based tools or ones shared with other groups, if the team does not have complete and exclusive governance, the tool might be updated and setting changed even during measurements, without the team being notified of such actions[16].
  - Any other special circumstances that could affect the numbers reported. For example:
    - According to the download site[17] of SonarQube the detection of injection flows for Java is only available in the commercial editions of the software.
    - Before version 6.6 of SonarQube[18] the built-in profiles could be changed. Potentially leading to researchers miss-reporting what setting were used.
- Make a direct statement that all of the tools used are listed. No other scripts, methods, transformations were needed to produce the reported results.
- If the input was open-source data, identify it. If the input data is the property of a company, include information that might reasonably be needed for other researchers to contact that company and request access to the data.

[16]Please note, that determining this property might require care. For example CAST promotes its solution as being SaaS subscription, but actually the measurements are done locally, only the results are uploaded into the cloud. https://www.castsoftware.com/products/highlight/pricing (last accessed 2021.10.05)

[17]https://www.sonarqube.org/downloads/ (last accessed 2021.10.05)

[18]https://blog.sonarsource.com/sonarqube-6.5-in-screenshots (last accessed 2021.10.05)

7 Further work

There are several ways to extend the research presented in this paper.

We have only reviewed articles published at the most prestigious workshop and conference series. The whole field of technical debt research is bigger, containing papers published at other venues and journals, that should be reviewed.

We have limited resources, so we did not try to reproduce the results in the papers we have found to be potentially reproducible. It would be possible to get a deeper understanding of the situation by doing the actual reproducibility check.

It would also be interesting to investigate how cloud-based technical debt tools impact research and work efforts. On the one side, we can expect cloud-based solutions to spread in both work and research environments, as they free people from the mundane task of allocating and managing hardware resources, installing and managing software tools, making sure these tools are up to date and configured for security, etc. On the other side, the automated updating of the platform and software resources can cause problems for research reproducibility as these changes might happen independently from the researchers at any point in time (even during experiments).

Acknowledgment

The third author was supported by the Project no. TKP2020-NKA-06 (Application domain specific highly reliable IT solutions) with the support from the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Program funding scheme.

References

[1] S. Ahmadjee, C. Mera-Gómez, R. Bahsoon, Assessing Smart Contracts Security Technical Debts, *2021 IEEE/ACM International Conference on Technical Debt (TechDebt)*, 2021, pp. 6-15. ⇒350

[2] A. Aldaeej, C. Seaman, From Lasagna to Spaghetti: A Decision Model to Manage Defect Debt, *2018 IEEE/ACM International Conference on Technical Debt (TechDebt)*, 2018, pp. 67-71. ⇒348

[3] R. Alfayez, W. Alwehaibi, R. Winn, E. Venson, B. Boehm, A systematic literature review of technical debt prioritization, *In Proceedings of the 3rd International Conference on Technical Debt (TechDebt '20)*, 2020, pp. 1–10. ⇒341

[4] R. R. Alfayez, P. Behnamghader, K. Srisopha, B. Boehm, An Exploratory Study on the Influence of Developers in Technical Debt, *2018 IEEE/ACM International Conference on Technical Debt (TechDebt)*, 2018, pp. 1-10. ⇒348

[5] S. Alhusain, Predicting Relative Thresholds for Object Oriented Metrics, *2021 IEEE/ACM International Conference on Technical Debt (TechDebt)*, 2021, pp. 55-63. ⇒351

[6] T. Amanatidis, N. Mittas, A. Chatzigeorgiou, A. Ampatzoglou, L. Angelis, The Developer's Dilemma: Factors Affecting the Decision to Repay Code Debt, *2018 IEEE/ACM International Conference on Technical Debt (TechDebt)*, 2018, pp. 62-66. ⇒348

[7] A. Ampatzoglou, A. Ampatzoglou, A. Chatzigeorgiou, P. Avgeriou, P. Abrahamsson, A. Martini, U. Zdun, K. Systa, The Perception of Technical Debt in the Embedded Systems Domain: An Industrial Case Study, *2016 IEEE 8th International Workshop on Managing Technical Debt (MTD)*, 2016, pp. 9-16. ⇒347

[8] A. Ampatzoglou, A. Michailidis, C. Sarikyriakidis, A. Ampatzoglou, A. Chatzigeorgiou, P. Avgeriou, A Framework for Managing Interest in Technical Debt: An Industrial Validation, *2018 IEEE/ACM International Conference on Technical Debt (TechDebt)*, 2018, pp. 115-124. ⇒337, 347, 349, 352

[9] M. F. Aniche, G. A. Oliva, M. A. Gerosa, Are the Methods in Your Data Access Objects (DAOs) in the Right Place? A Preliminary Study, *Sixth International Workshop on Managing Technical Debt (MTD '14)*, 2014, pp. 47-50. ⇒345

[10] P. C. Avgeriou, D. Taibi, A. Ampatzoglou, A. F. Fontana, T. Besker, A. Chatzigeorgiou, V. Lenarduzzi, A. Martini, A. Moschou, I. Pigazzini, N. Saarimäki, D. Sas, S. Toledo, A. Tsintzira. An Overview and Comparison of Technical Debt Measurement Tools, in *IEEE Software*, vol. 38, no. 3, pp. 61-71, May-June 2021. ⇒336, 338

[11] B. Barta, G. Manz, I. Siket, R. Ferenc, Challenges of SonarQube Plug-In Maintenance, *IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, 2019, pp. 574-578. ⇒336, 338

[12] T. Besker, A. Martini, J. Bosch, Technical debt cripples software developer productivity: a longitudinal study on developers' daily software development work, *In Proceedings of the 2018 IEEE/ACM International Conference on Technical Debt (TechDebt '18)*, 2018, pp. 105–114. ⇒341

[13] J. Bohnet, J. Döllner, Monitoring code quality and development activity by software maps, *In Proceedings of the 2nd Workshop on Managing Technical Debt (MTD '11)*, ACM, 2011, New York, NY, USA, 9–16. ⇒341

[14] N. Brown, Y.g Cai, Y. Guo, R. Kazman, M. Kim, P. Kruchten, E. Lim, A. MacCormack, R. L. Nord, I. Ozkaya, R. Sangwan, C. Seaman, K. Sullivan, N. Zazworka, Managing technical debt in software-reliant systems, *In Proceedings of the FSE/SDP workshop on Future of software engineering research (FoSER '10)*, 2010, pp. 47–52. ⇒337, 338, 341, 342, 343

[15] R. Capilla, T. Mikkonen, C. Carrillo, F. A. Fontana, I. Pigazzini, V. Lenarduzzi, Impact of Opportunistic Reuse Practices to Technical Debt, *2021 IEEE/ACM International Conference on Technical Debt (TechDebt)*, 2021, pp. 16-25. ⇒351

[16] A. Chatzigeorgiou, A. Ampatzoglou, A. Ampatzoglou, T. Amanatidis, Estimating the breaking point for technical debt, *2015 IEEE 7th International Workshop on Managing Technical Debt (MTD)*, 2015, pp. 53-56. ⇒346

[17] Carnegie Mellon University, Software Engineering Institute, Managing Technical Debt Workshops, previous editions, 2010, https://resources.sei.cmu.edu/news-events/events/techdebt/past.cfm, last visited: 2021.10.05. ⇒342, 345

[18] Y. Crespo, A. Gonzalez-Escribano, M. Piattini, Carrot and Stick approaches revisited when managing Technical Debt in an educational context, *2021 IEEE/ACM International Conference on Technical Debt (TechDebt)*, 2021, pp. 99-108. ⇒351

[19] D. Cruz, A. Santana, E. Figueiredo, Detecting bad smells with machine learning algorithms: an empirical study, *In Proceedings of the 3rd International Conference on Technical Debt (TechDebt '20)*, 2020, ACM, New York, NY, USA, 31–40. ⇒350

[20] W. Cunningham, The WyCash Portfolio Management System, *In Addendum to the proceedings on Object-oriented programming systems, languages, and applications (Addendum) (OOPSLA '92)*. ACM, NY, USA, 1992, 29–30. ⇒337

[21] B. Curtis, J. Sappidi, A. Szynkarski, Estimating the size, cost, and types of Technical Debt, *Third International Workshop on Managing Technical Debt (MTD'12)*, 2012, pp. 49-53. ⇒344

[22] R. R. de Almeida, R. do Nascimento Ribeiro, C. Treude, U. Kulesza, Business-Driven Technical Debt Prioritization: An Industrial Case Study, *2021 IEEE/ACM International Conference on Technical Debt (TechDebt)*, 2021, pp. 74-83. ⇒351

[23] M. A. de Freitas Farias, M. G. de Mendonça Neto, A. B. d. Silva, R. O. Spínola, A Contextualized Vocabulary Model for identifying technical debt on code comments, *2015 IEEE 7th International Workshop on Managing Technical Debt (MTD)*, 2015, pp. 25-32. ⇒346

[24] J. de Groot, A. Nugroho, T. Bäck, J. Visser, What is the value of your software?, *Third International Workshop on Managing Technical Debt (MTD'12)*, 2012, pp. 37-44. ⇒344

[25] M. di Biase, A. Rastogi, M. Bruntink, A. van Deursen, The Delta Maintainability Model: Measuring Maintainability of Fine-Grained Code Changes, *2019 IEEE/ACM International Conference on Technical Debt (TechDebt)*, 2019, pp. 113-122. ⇒349

[26] D. Falessi, R. Kazman, Worst Smells and Their Worst Reasons, *2021 IEEE/ACM International Conference on Technical Debt (TechDebt)*, 2021, pp. 45-54. ⇒351

[27] D. Falessi, P. Kruchten, R. L. Nord, I. Ozkaya, Technical debt at the crossroads of research and practice: report on the fifth international workshop on managing technical debt, 2014, *SIGSOFT Softw. Eng. Notes 39, 2*, 31–33. ⇒343, 345

[28] D. Falessi, A. Reichel, Towards an open-source tool for measuring and visualizing the interest of technical debt, *2015 IEEE 7th International Workshop on Managing Technical Debt (MTD)*, 2015, pp. 1-8. ⇒346

[29] D. Falessi, A. Voegele, Validating and prioritizing quality rules for managing technical debt: An industrial case study, *2015 IEEE 7th International Workshop on Managing Technical Debt (MTD)*, 2015, pp. 41-48. ⇒347

[30] F. A. Fontana, V. Ferme, S. Spinelli, Investigating the impact of code smells debt on quality code evaluation, *Third International Workshop on Managing Technical Debt (MTD'12)*, 2012, pp. 15-22. ⇒344

[31] F. A. Fontana, V. Ferme, M. Zanoni, R. Roveda, Towards a prioritization of code debt: A code smell Intensity Index, *2015 IEEE 7th International Workshop on Managing Technical Debt (MTD)*, 2015, pp. 16-24. ⇒346

[32] F. A. Fontana, R. Roveda, M. Zanoni, Technical Debt Indexes Provided by Tools: A Preliminary Discussion, *2016 IEEE 8th International Workshop on Managing Technical Debt (MTD)*, 2016, pp. 28-31. ⇒347

[33] I. Gat, J. D. Heintz, From assessment to reduction: how cutter consortium helps rein in millions of dollars in technical debt, *In Proceedings of the 2nd Workshop on Managing Technical Debt (MTD '11)*, 2011, Waikiki, Honolulu, HI, USA, 24–26. ⇒343

[34] R. Gomes, C. Siebra, G. Tonin, A. Cavalcanti, Fabio Q.B. da Silva, Andre L.M. Santos, R. Marques, An extraction method to collect data on defects and effort evolution in a constantly modified system, *In Proceedings of the 2nd Workshop on Managing Technical Debt (MTD '11)*, 2011, Waikiki, Honolulu, HI, USA, 27–30. ⇒343

[35] I. Griffith, D. Reimanis, C. Izurieta, Z. Codabux, A. Deo, B. Williams, The Correspondence Between Software Quality Models and Technical Debt Estimation Approaches, *Sixth International Workshop on Managing Technical Debt (MTD '14)*, 2014, pp. 19-26. ⇒345

[36] I. Khomyakov, Z. Makhmutov, R. Mirgalimova, A. Sillitti, 2019, Automated Measurement of Technical Debt: A Systematic Literature Review, *In Proceedings of the 21st International Conference on Enterprise Information Systems - Volume 2(ICEIS)*, 2019, pages 95-106. ⇒338

[37] J. R. Lahti, A. -P. Tuovinen, T. Mikkonen, Experiences on Managing Technical Debt with Code Smells and AntiPatterns, *2021 IEEE/ACM International Conference on Technical Debt (TechDebt)*, 2021, pp. 36-44. ⇒351

[38] V. Lenarduzzi, T. Besker, D. Taibi, A. Martini, F. A. Fontana, Technical Debt Prioritization: State of the Art. A Systematic Literature Review, *arXiv:1904.12538*. ⇒338

[39] V. Lenarduzzi, F. Lomio, H. Huttunen, D. Taibi, Are SonarQube Rules Inducing Bugs?, *IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, 2020, pp. 501-511. ⇒336, 339

[40] V. Lenarduzzi, V. Mandić, A. Katin, D. Taibi, How long do Junior Developers take to Remove Technical Debt Items?, *In Proceedings of the 14th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement(ESEM '20)*, 2020, Article 30, 1–6. ⇒336, 338

[41] A. Lerina, L. Nardi, Investigating on the Impact of Software Clones on Technical Debt, *2019 IEEE/ACM International Conference on Technical Debt (TechDebt)*, 2019, pp. 108-112. ⇒349

[42] E. d. S. Maldonado, E. Shihab, Detecting and quantifying different types of self-admitted technical Debt, *2015 IEEE 7th International Workshop on Managing Technical Debt (MTD)*, 2015, pp. 9-15. ⇒346

[43] M. A. Al Mamun, C. Berger, J. Hansson, Explicating, Understanding, and Managing Technical Debt from Self-Driving Miniature Car Projects, *Sixth International Workshop on Managing Technical Debt (MTD '14)*, 2014, pp. 11-18. ⇒346

[44] D. Marcilio, R. Bonifácio, E. Monteiro, E. Canedo, W. Luz, G. Pinto, Are Static Analysis Violations Really Fixed? A Closer Look at Realistic Usage of SonarQube, *IEEE/ACM 27th International Conference on Program Comprehension (ICPC)*, 2019, pp. 209-219. ⇒336, 339

[45] J. Y. Monteith, J. D. McGregor, Exploring software supply chains from a technical debt perspective, *4th International Workshop on Managing Technical Debt (MTD '13)*, 2013, pp. 32-38. ⇒344

[46] A. Mori, G. Vale, M. Viggiato, J. Oliveira, E. Figueiredo, E. Cirilo, P. Jamshidi, C. Kastner, Evaluating Domain-Specific Metric Thresholds: An Empirical Study, *2018 IEEE/ACM International Conference on Technical Debt (TechDebt)*, 2018, pp. 41-50. ⇒348

[47] A. Nugroho, J. Visser, T. Kuipers, An empirical model of technical debt and interest, *In Proceedings of the 2nd Workshop on Managing Technical Debt (MTD '11)*, 2011, Waikiki, Honolulu, HI, USA, 1–8. ⇒343

[48] E. Parodi, S. Matalonga, D. Macchi, M. Solari, Comparing technical debt in student exercises using test driven development, test last and ad hoc programming, *2016 XLII Latin American Computing Conference (CLEI)*, 2016, pp. 1-10. ⇒336, 338

[49] I. Pigazzini, F. A. Fontana, V. Lenarduzzi, D. Taibi, Towards microservice smells detection, *In Proceedings of the 3rd International Conference on Technical Debt (TechDebt '20)*, 2020, ACM, New York, NY, USA, 92–97. ⇒350

[50] R. Plösch, J. Bräuer, M. Saft and C. Körner, Design Debt Prioritization: A Design Best Practice-Based Approach, *2018 IEEE/ACM International Conference on Technical Debt (TechDebt)*, 2018, pp. 95-104. ⇒348

[51] W. W. Royce, Managing the development of large software systems, *Proceedings of IEEE WESCON*, 1970, pp. 1–9. ⇒337

[52] N. Saarimäki, M. T. Baldassarre, V. Lenarduzzi and S. Romano, On the Accuracy of SonarQube Technical Debt Remediation Time, *45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 2019, pp. 317-324. ⇒336, 338

[53] V. Lenarduzzi, N. Saarimäki, D. Taibi, On the Diffuseness of Code Technical Debt in Java Projects of the Apache Ecosystem, *2019 IEEE/ACM International Conference on Technical Debt (TechDebt)*, 2019, pp. 98-107. ⇒349

[54] A. Shapochka, B. Omelayenko, Practical Technical Debt Discovery by Matching Patterns in Assessment Graph, *2016 IEEE 8th International Workshop on Managing Technical Debt (MTD)*, 2016, pp. 32-35. ⇒347

[55] V. Singh, W. Snipes, N. A. Kraft, A Framework for Estimating Interest on Technical Debt by Monitoring Developer Activity Related to Code Comprehension, *Sixth International Workshop on Managing Technical Debt (MTD '14)*, 2014, pp. 27-30. ⇒345

[56] A. Sundelin, J. Gonzalez-Huerta, K. Wnuk, The hidden cost of backward compatibility: when deprecation turns into technical debt - an experience report, *In Proceedings of the 3rd International Conference on Technical Debt (TechDebt '20)*, 2020, ACM, New York, NY, USA, 67–76. ⇒350

[57] K. Szabados and A. Kovács, Technical debt of standardized test software, *2015 IEEE 7th International Workshop on Managing Technical Debt (MTD)*, 2015, pp. 57-60. ⇒346

[58] R. Szalay, Á. Sinkovics, Z. Porkoláb, The Role of Implicit Conversions in Erroneous Function Argument Swapping in C++, *20th International Working Conference on Source Code Analysis and Manipulation (SCAM)*, 2020, pp. 203-214. ⇒336

[59] J. Tan, D. Feitosa, P. Avgeriou, An empirical study on self-fixed technical debt, *In Proceedings of the 3rd International Conference on Technical Debt (TechDebt '20)*, 2020, ACM, New York, NY, USA, 11–20. ⇒350

[60] A. Tornhill, Prioritize Technical Debt in Large-Scale Systems Using CodeScene, *2018 IEEE/ACM International Conference on Technical Debt (TechDebt)*, 2018, pp. 59-60. ⇒348

[61] P. Wang, J. Yang, L. Tan, R. Kroeger, J. David Morgenthaler, Generating precise dependencies for large software, *4th International Workshop on Managing Technical Debt (MTD '13)*, 2013, pp. 47-50. ⇒345

[62] N. Zazworka, C. Seaman, F. Shull, Prioritizing design debt investment opportunities, *In Proceedings of the 2nd Workshop on Managing Technical Debt (MTD '11)*, 2011, pp. 39–42. ⇒344

[63] ∗ ∗ ∗, *MTD '11: Proceedings of the 2nd Workshop on Managing Technical Debt*, 2011, ACM, New York, NY, USA. ⇒337, 338, 341, 343

[64] ∗ ∗ ∗, *MTD '12: Proceedings of the Third International Workshop on Managing Technical Debt*, 2012, IEEE Press ⇒337, 338, 341, 343, 344

[65] ∗ ∗ ∗, *MTD '13: Proceedings of the 4th International Workshop on Managing Technical Debt*, 2013, IEEE Press. ⇒337, 338, 341, 343, 344

[66] ∗ ∗ ∗, *MTD '14: Proceedings of the 2014 Sixth International Workshop on Managing Technical Debt*, 2014, IEEE Computer Society. ⇒ 337, 338, 341, 343, 345

[67] ∗ ∗ ∗, *2015 IEEE 7th International Workshop on Managing Technical Debt (MTD)*, 2015, IEEE Computer Society. ⇒337, 338, 341, 343, 346

[68] ∗ ∗ ∗, *2016 IEEE 8th International Workshop on Managing Technical Debt (MTD)*, 2016, IEEE Computer Society. ⇒337, 338, 341, 343, 347
[69] ∗∗∗, *2018 IEEE/ACM International Conference on Technical Debt (TechDebt)*, 2018, IEEE Computer Society. ⇒337, 338, 341, 343, 347
[70] ∗∗∗, *2019 IEEE/ACM International Conference on Technical Debt (TechDebt)*, 2019, IEEE Computer Society. ⇒337, 338, 341, 343, 348
[71] ∗∗∗, *TechDebt '20: Proceedings of the 3rd International Conference on Technical Debt*, 2020, ACM, New York, NY, USA. ⇒337, 338, 341, 343, 350
[72] ∗∗∗, *2021 IEEE/ACM International Conference on Technical Debt (TechDebt)*, 2021, IEEE Computer Society. ⇒337, 338, 341, 343, 350

Received: October 28, 2021 • Revised: December 14, 2021

Acta Univ. Sapientiae, Informatica **13,** 2 (2021) 361–372

DOI: 10.2478/ausi-2021-0017

DP-solver: automating dynamic programming

Zoltan KATAI
Sapientia Hungarian University of Transylvania
Department of Mathematics and Informatics
Tirgu Mures, Romania
email: katai_zoltan@ms.sapientia.ro

Attila ELEKES
Sapientia Hungarian University of Transylvania
email: attilaelekes97@gmail.com

Abstract. Dynamic programming (DP) is a widely used optimization method with several applications in various fields of science. The DP problem solving process can be divided in two phases: mathematical part and programming part. There are a number of researchers for whom the mathematical part is available, but they are not familiar with computer programming. In this paper we present a software tool that automates the programming part of DP and allows users to solve problems based only on their mathematical approach. The application builds up the "d-graph model" of the problem to be solved and applies the "d-variant" of the corresponding single source shortest path algorithm. In addition, we report experimental results regarding the efficiency of the tool relative to the Matlab implementation.

1 Introduction

Dynamic programming (DP) is an optimization method used in numerous fields of science. It was proposed by Richard Bellman in his book published

Computing Classification System 1998: F.2.2
Mathematics Subject Classification 2010: 68R15
Key words and phrases: dynamic programming, d-graphs, software tool

in 1957 [1]. Since that, this strategy has become an often used problem solving method in applied mathematics, computer sciences, artificial intelligence, bioinformatics, macroeconomics, etc. The solving process of these kinds of problems can be divided into two major steps. Firstly, a functional equation is established, which describes the problem solving process in a recursive way by implementing the principle of optimality. Secondly, a computer program is implemented, which processes the recursive formula in a bottom-up way. In this paper we will refer to these two phases as mathematical and programming parts of DP.

Due to the diversity of applications of this optimization method, the programming part of DP may cause difficulties for researchers who are not familiar with the programming languages or scripts. In this paper we present a software tool that automates the programming part of DP and allows users to solve problems based only on their mathematical approach.

2 Literature review

As mentioned above, in the last more than half century DP has been applied in several fields of science. More recently, for example, it was applied for solving the k-Color Shortest Path Problem that arises in the field of transmission networks design [3]. The authors of this study compared their DP approach with two previously published methods and they found that the DP algorithm vastly outperformed previous approaches. The authors of paper [18] present a survey of financial applications under a specific semimartingale result of Markov chains and two of the described strategies apply DP approach. Besides, other recent studies apply DP in the field of genomics and bioinformatics [15], for developing modern rainwater harvesting systems [17], for optimizing the synchronization and reducing gear-shifting time in mechanical transmissions [13], and even for solving the School Bus Routing Problem [19].

As DP became more widespread, software tools for solving (more or less automatically) certain DP problem families began to appear. For example, the Stochastic Dynamic Programming application (published in 1995) was a commercial software for solving general stochastic and deterministic optimization problems [14]. This tool (running on PCs using DOS operating system) was able (i) to accommodate user-specified conditions and functions of stage, state and decision; (ii) to minimize or maximize the optimal value function; (iii) to solve finite and infinite time-horizon problems, etc. Another software tool for solving DP problems is DP2PN2Solver [4]. This application uses specialized

Petri nets, named Bellman nets, to solve DP problems. In addition, it should also be mentioned the tool named MDPToolbox, which has been developed since 2004 [2]. MDPToolbox is a set of functions for solving Markov-decision Problems in various platforms like Matlab, Scilab, R and GNU Octave.

De Moor [16] also emphasizes the need for DP solvers. After stating that in contrast with other techniques (such as linear programming, where there exists a single generic program that solves all instances) DP is usually regarded as a design technique where each application is designed as an individual program, he argues that it would be much preferable if dynamic programming could be understood as a software component. The component presented by De Moor is suitable for a large class of applications in which the decision process is a sequential scan of the input sequence. With respect to the programming part he introduces a C++ and Haskell program and concludes that the simplicity offered by lazy functional programming is preferable. Another useful tool is VisuAlgo [24] which is a website for visualizing different algorithms including DP strategies. This tool was developed mostly for educational purposes and also uses a programming language (JavaScript) instead of mathematical functional equations.

Relatively recent studies were realized in this field by Kátai [5, 6, 7, 8, 10, 12]. In [5], he proposed the concept of $\mathfrak{d}$-graphs for modelling DP problems. The software tool presented in [9] was based on this approach. Later, Katai refined the model and introduced the notion of generalized $\mathfrak{d}$-graphs [5]. Katai and Fulop [11] compared Petri nets (used by the above referred DP2PN2Solver) and $\mathfrak{d}$-graphs as tools for intermediate representation of certain DP problems. They emphasize two advantages of $\mathfrak{d}$-graphs: (i) $\mathfrak{d}$-graphs can handle problems with cyclic target function; (ii) while the usage of DP2PN2-Solver requires the knowledge of gDPS language, $\mathfrak{d}$-graphs can be built based on the functional equation. In [12], Katai defines the so-called generalized deterministic Markov decision processes where each decision may result in more than one next state. The author also proposed a combined $\mathfrak{d}$-graph algorithm for finding optimal policies for the previously mentioned Markov processes.

From this enumeration the probably most generally known and commonly used software cannot be missed. The Matlab is an available commercial software, which is able to perform numeric computations and to visualize data structures and models. There are also a lot of available plugins for solving various problems, including DP. The programming language used by Matlab is also relatively simple and has a high abstraction level.

In this paper we present a simplified implementation of the approach Katai proposed [10, 12], and more importantly, we present a software tool that imple-

ments this strategy and also visualizes the DP solving process. In addition, we report experimental results regarding the efficiency of the tool we developed (compared with the Matlab solutions).

3 Mathematical part of the DP solving process

The core of DP is the principle of optimality, which states that the optimal solution for the initial problem is based on the optimal solutions of its subproblems. This principle is expressed by a recursive formula (target function). Subproblems are solved in bottom-up order (starting from the level of the trivial ones), and the optimal solution of the current subproblem is computed (based on the recursive formula) from the optimal solutions of previously solved (and stored) simpler subproblems. The solving process is a sequence of optimal decisions (min or max) that results in the optimal solution of the original problem. To illustrate this approach we present the strategy on a sample problem.

Two Person Game: Let us consider a sequence of natural numbers stored in array $a_i, i = 0, \ldots, n-1$ (where n is even). In every step the current player takes a number from the beginning or the end of the sequence. The question is, what is the highest score which can be collected by the beginner? We assume that both players are taking the optimal decision.

Assuming that cell c_{ij} of array c $(i, j = 0, \ldots, n-1)$ stores the optimal value associated to sub-sequence $a_i \ldots a_j$, the functional equation for this problem is the following:

$$c_{ij} = \begin{cases} 0 & j \leq i \\ \max\{a_i + \min\{c_{i+1,j-1}, c_{i+2,j}\}, & \\ \qquad a_j + \min\{c_{i+1,j-1}, c_{i,j-2}\}\} & i < j \end{cases}$$

In this problem the sub-problems are represented by the even length subsequences of the original sequence. More precisely, cell $c_{i,j}$ stores the maximum score that can be collected by the first player if they play on sub-sequence $a_i \ldots a_j$. The solving process implies the computation of the highest scores which the first player can collect from all 2, 4, ..., n length sub-sequences. Function max reflects that player 1 takes optimal decisions. Function min reflects that player 2 also takes optimal decisions.

4 The idea behind the software

As mentioned above, Kátai [10] proposed generalized d-graphs as intermediate representations to solve DP problems based on the optimal path algorithms in weighted graphs. There are three main single-source shortest path algorithms in weighted digraph:

- If the graph is cycle free, the optimal paths can be found based on the topological order of the vertices (Viterbi algorithm);
- If the graph contains circles and all arcs have positive weights, then we use the Dijkstra algorithm;
- If the graph contains negative weighted arcs, but does not contains "negative cycles" (the sum of the weights of the arcs of the cycle is negative), the problem is solved by the Bellman-Ford algorithm.

Katai [7, 10] emphasizes that, on the one hand, these algorithms implement DP strategies and, on the other hand, the DP problem solving process can be reduced to these strategies. Katai suggests the following approach (for details see [10]): (i) the DP problem to be solved is modelled with a d-graph where p-vertices represent the sub-problems and d-vertices represent the possible choices defined by the target function; (ii) the optimal decision sequence is found by searching for the optimal d-path between the source vertex, representing the original problem, and a virtual sink vertex, representing the trivial sub-problems. In the second step the d-variant of the corresponding optimal path algorithm is applied (d-Viterbi, d-Dijkstra, d-BellmanFord).

5 DP-solver

DP-solver is a software tool for solving DP problems based only on the mathematical functional equation. To automate the programming part of the DP solving process it takes advantage of the relationship between dynamic programming and optimal paths algorithms. The tool is easy to use in the sense that it does not presume to learn any programming language and the input format is very similar to the mathematical syntax. Another functionality is the visualization of the solving process. This feature uses 1D and 2D arrays, where the cells represent the sub-problems. Thus, DP-solver is both a scientific and educational tool.

The application was implemented using the JavaFX language and its core is a Java package, named exp4j [20], which is able to interpret mathematical

expressions. From developer point of view, exp4j is an easy to use package which knows the fundamental mathematical operators and functions and it is able to work with variables and general formulas and to detect syntactic and runtime errors. Since it is easy to define more operators and functions, the package could comfortably be adapted to the current software requirements.

As mentioned above, the main functionality of the software is to solve DP problems based on the mathematical equation. Accordingly, the user's role includes the following steps: (i) to introduce the formula and its variables, and to ask the tool (ii) to evaluate the formula and (iii) to visualize the solving process. The application also enables users to save, reload and edit previously created DP models. In visualization mode the software works similarly to a media player.

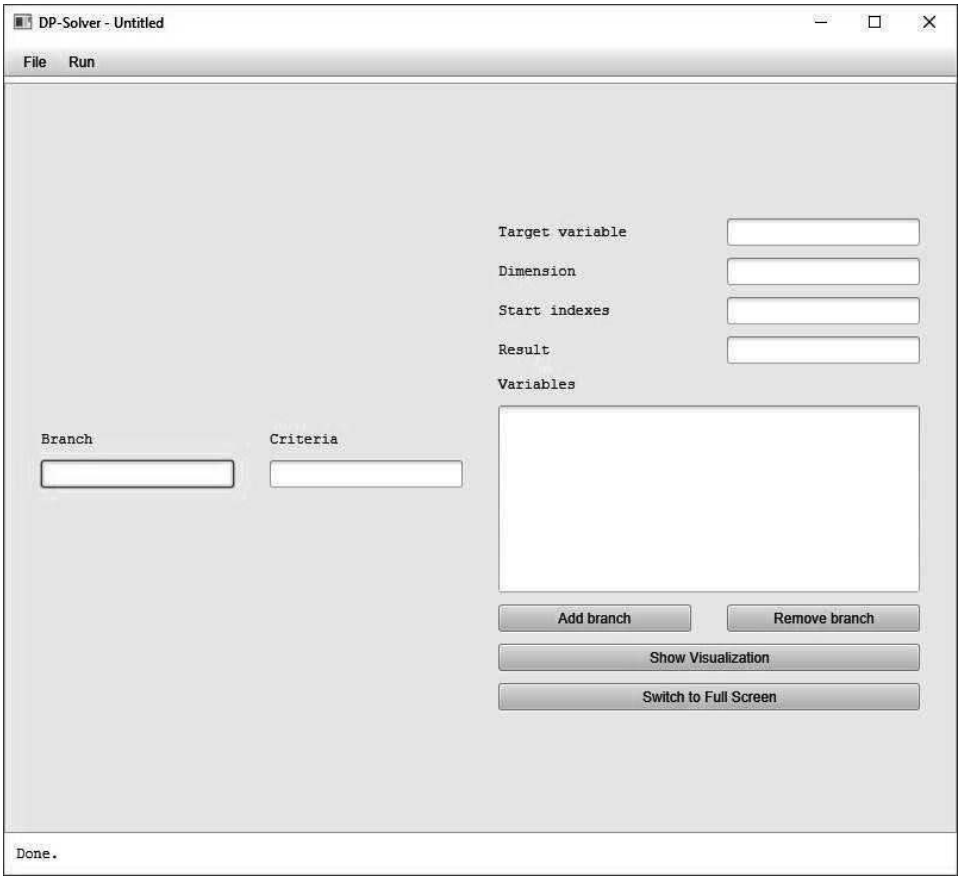


Figure 1: The main window of DP-solver

6 The graphical interface

Figure 1 shows the graphical interface of DP-solver. In the input fields labeled with "Branch" and "Criteria" the user has to introduce the formula, branch-by-branch. Attached to each branch, the corresponding criterion has to be also introduced.

In the "Target variable" field the user has to specify the array which stores the optimal values corresponding to the optimal solutions of the subproblems. The dimension of the problem (dimension of the target variable) and the indices of the cell that represents the optimal value of the original problem ("Start indices") are also required to be specified. In the "Variables" field the input variables (which are included in the formula) have to be defined. Before the evaluation starts the software checks all input fields for syntax errors. The optimal value attached to the optimal solution is displayed in the field named "Result".

The status bar (on the bottom of the window) informs the user about the state of the problem solving process. For example, the user is informed about the currently performed task and error messages also appears here.

Figure 2: Input and output of the Two Person Game problem

Figure 2 illustrates the usage of DP-solver for the above presented Two Person Game problem for a 10-length number sequence. As it could be noticed, the recursive formula has two branches, the target variable is a 10x10 bi-dimensional array and cell c_{09} represents the original problem. If the input sequence stored in array a is {19, 2, 4, 16, 3, 15, 4, 14, 17, 1} then the optimal value which is going to appear (after pushing the "Run" button) in the "Result" field is 65.

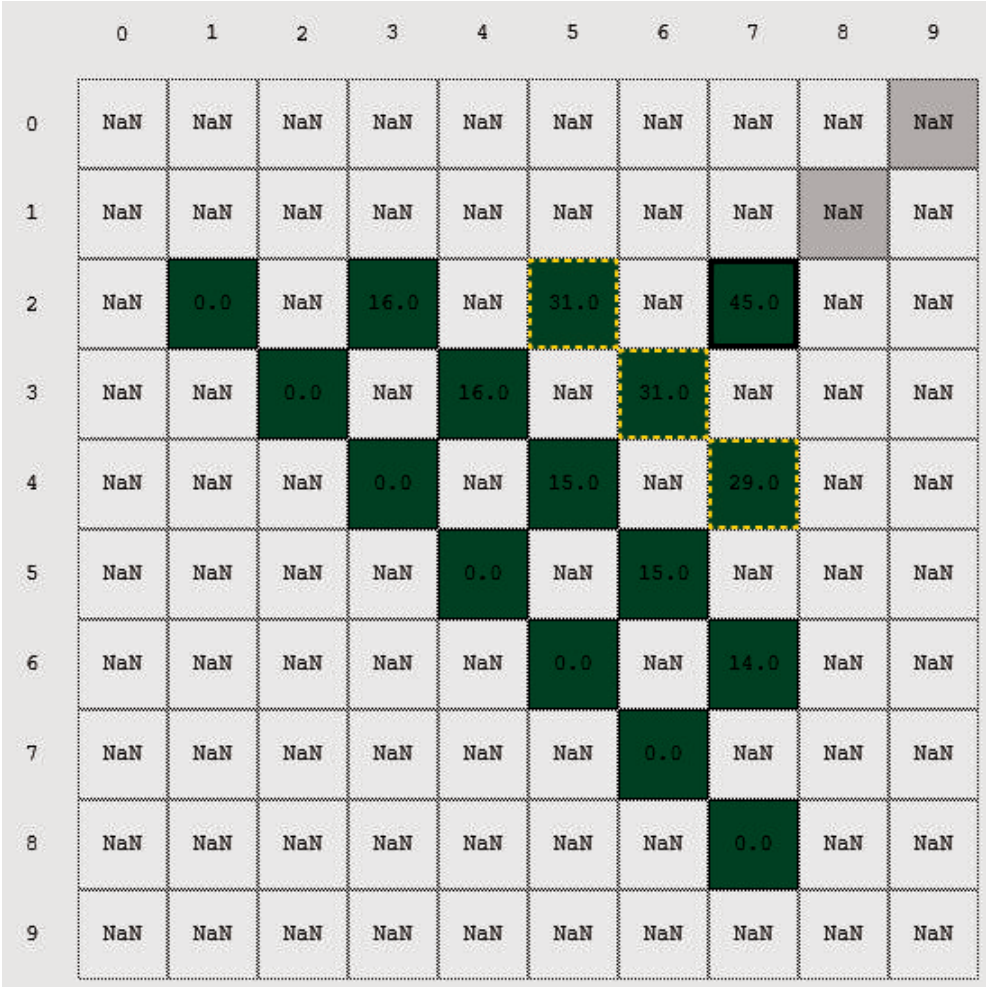
After the result appears, the solving process can be visualized step by step. Since sub-problems correspond to the even length sub-sequences, only every second diagonal of the target variable is filled (green cells containing non-NaN values) (see Fig. 3). The diagonal below the main diagonal stores the optimal

	0	1	2	3	4	5	6	7	8	9
0	NaN	19.0	NaN	23.0	NaN	37.0	NaN	52.0	NaN	65.0
1	0.0	NaN	4.0	NaN	18.0	NaN	33.0	NaN	50.0	NaN
2	NaN	0.0	NaN	16.0	NaN	31.0	NaN	45.0	NaN	46.0
3	NaN	NaN	0.0	NaN	16.0	NaN	31.0	NaN	46.0	NaN
4	NaN	NaN	NaN	0.0	NaN	15.0	NaN	29.0	NaN	30.0
5	NaN	NaN	NaN	NaN	0.0	NaN	15.0	NaN	31.0	NaN
6	NaN	NaN	NaN	NaN	NaN	0.0	NaN	14.0	NaN	21.0
7	NaN	NaN	NaN	NaN	NaN	NaN	0.0	NaN	17.0	NaN
8	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0	NaN	17.0
9	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0	NaN

Figure 3: The completed visualization process (Two Person Game problem)

values for the trivial sub-problems (corresponding to the 0-length sequences). The optimal values of "non-trivial cells" are computed from diagonal to diagonal. Lastly, cell c_{09} gets the value 65.

The screenshot from Figure 4 captures the solving process in an intermediate step. By default every cell has a white background color and dotted border. The default value of each cell is NaN until the result of the corresponding sub-problem is computed. Accessed cells get a light gray background color (the solving process reached them, but their values are not computed yet). The



	0	1	2	3	4	5	6	7	8	9
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	NaN	0.0	NaN	16.0	NaN	31.0	NaN	45.0	NaN	NaN
3	NaN	NaN	0.0	NaN	16.0	NaN	31.0	NaN	NaN	NaN
4	NaN	NaN	NaN	0.0	NaN	15.0	NaN	29.0	NaN	NaN
5	NaN	NaN	NaN	NaN	0.0	NaN	15.0	NaN	NaN	NaN
6	NaN	NaN	NaN	NaN	NaN	0.0	NaN	14.0	NaN	NaN
7	NaN	NaN	NaN	NaN	NaN	NaN	0.0	NaN	NaN	NaN
8	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0	NaN	NaN
9	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Figure 4: The visualization process in progress (Two Person Game problem)

current cell is marked with black thick solid border and dark gray background color. The cells representing the direct descendants of the current subproblem (the value of the current cell is computed based on these cells' values) are highlighted with a yellow dashed border. When a subproblem is solved, the related cell turns its border to solid and its background color to green. The buttons for controlling the visualization process are in the bottom left corner of the main window.

7 Performance

To find out our software's performance, we compared DP-Solver with Matlab®. We have chosen Matlab®, because it is one of the most commonly used, well known and widely available programming software used for mathematical problem solving. In our tests we used the Matlab® R2017b version. The laptop we used has the following configuration: Intel® Celeron® N2940 CPU (quad core, up to 2.23 GHz), 4 GB DDR3L memory and Samsung 860 EVO SSD.

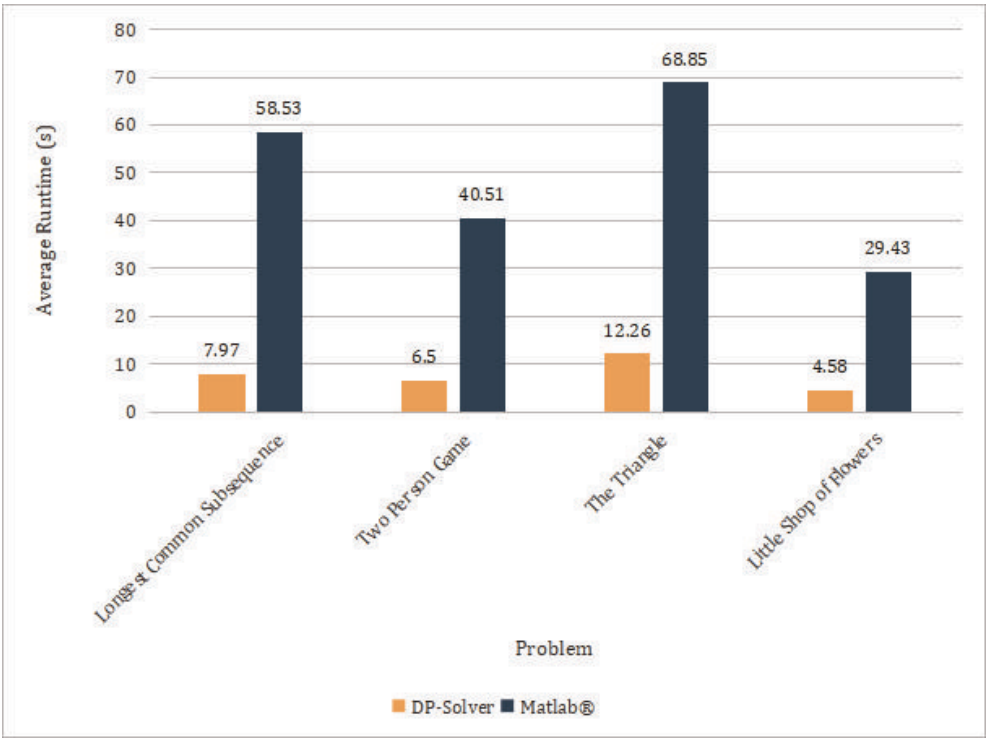


Figure 5: DP-solver vs. Matlab® results

We performed the comparison for the DP solutions of the problems referred below (for details see the corresponding references). The analysis was based on the average run-time after 10 measurements (the brackets include the size of the corresponding one- or bi-dimensional input array(s)):

- Longest Common Subsequence [23] ([1000], [500])

- Two Person Game ([1000])
- The Triangle [21] ([1000]x[1000])
- Little Shop of Flowers [22] ([750]x[1000])

As shown in Figure 5 DP-solver was 6.1 times faster than the Matlab® implementations of the DP solutions.

8 Conclusions

In this paper we presented DP-solver, an easy to use software tool for those who are familiar with the mathematical fundamentals of dynamic programming. The application uses as intermediate representation the d-graph model of the DP problem to be solved. Our experimental results shows that the tool can provide the optimal solution more than six times faster than the corresponding Matlab implementation. The application also includes a visualization module. Consequently, DP-solver is both a scientific and educational software tool.

References

[1] R. Bellman, *Dynamic Programming*, Princeton University Press, New Jersey, 1957. ⇒362

[2] I. Chadès, G. Chapron, M. J. Cros, F. Garcia, R. Sabbadin, MDPtoolbox: a multi-platform toolbox to solve stochastic dynamic programming problems, *Ecography*, **37,** 9 (2014) 916–920. ⇒363

[3] D. Ferone, P. Festa, S. Fugaro, T. Pastore, A dynamic programming algorithm for solving the k-color shortest path problem, *Optimization Letters*, (2020) 1–20. ⇒362

[4] M. Holger, DP2PN2Solver: A flexible dynamic programming solver software tool, *Control and Cybernetics*, **35** (2002) 687–702. ⇒362

[5] Z. Kátai, Dynamic programming and d-graphs, *Studia Universitatis Babes-Bolyai, Informatic*, **51,** 2 (2006) 41–52. ⇒363

[6] Z. Kátai, Dynamic programming strategies on the decision tree hidden behind the optimizing problems, *Informatics in Education - An International Journal*, **6,** 1 (2007) 115–138. ⇒363

[7] Z. Kátai, The single-source shortest paths algorithms and the dynamic programming, *Teaching Mathematics and Computer Science*, **6** (2008) 25–35. ⇒363, 365

[8] Z. Kátai, Dynamic programming as optimal path problem in weighted digraph, *Acta Mathematica Academiae Paedagogicae Nyíregyháziensis*, **24** (2008) 201–208. ⇒363

[9] Z. Kátai, Á. Csíki, Automated dynamic programming, *Acta Universitatis Sapientiae, Informatica*, **1,** 2 (2009) 149–164. ⇒363
[10] Z. Kátai, Modelling dynamic programming problems by generalized d-graphs, *Acta Universitatis Sapientiae, Informatica*, **2,** 2 (2010) 210–230. ⇒363, 365
[11] Z. Kátai, P. I. Fülöp, Modeling dynamic programming problems: Petri nets versus d-graphs, *Proceedings of 8th International Conference on Applied Informatics (ICAI)*, Eger, Hungary, 2010, pp. 217–226. ⇒363
[12] Z. Kátai, Solving Markov decision processes by d-graph algorithms, *Control and Cybernetics*, **41,** 3 (2012) 577–593. ⇒363
[13] Z. Lu, G. Tian, S. Onori, Multistage time-optimal control for synchronization process in electric-driven mechanical transmission with angle alignment considering torque response process, *Journal of Dynamic Systems, Measurement, and Control*, **143,** 4 (2021) 041006. ⇒362
[14] B. C. Lubow, SDP: generalized software for solving stochastic dynamic optimization problems, *Wildlife Society Bulletin*, (1995) 738–742. ⇒362
[15] M. Maiolo, S. Ulzega, M. Gil, M. Anisimova, Accelerating phylogeny-aware alignment with indel evolution using short time Fourier transform, *NAR Genomics and Bioinformatics*, **2,** 4 (2020) lqaa092. ⇒362
[16] O. De Moor, Dynamic programming as a software component, *Proceedings of IEEE Computer Society, Conference on Circuits, Systems, Computers and Communications (CSCC)*, Athens, Greece, 1999. ⇒363
[17] C. Nop, R. M. Fadhil, K. Unami, A multi-state Markov chain model for rainfall to be used in optimal operation of rainwater harvesting systems, *Journal of Cleaner Production*, (2020) 124912. ⇒362
[18] E. Savku, G. W. Weber, A regime-switching model with applications to finance: markovian and non-markovian cases, *Dynamic Economic Problems with Regime Switches*, Springer, Cham, 2021, pp. 287–309. ⇒362
[19] P. Shang, L. Yang, Z. Zeng, L. C. Tong, Solving school bus routing problem with mixed-load allowance for multiple schools, *Computers & Industrial Engineering*, (2020) 106916. ⇒362
[20] ∗ ∗ ∗ exp4j, https://lallafa.objecthunter.net/exp4j ⇒365
[21] ∗ ∗ ∗ International Olympiad in Informatics (IOI), The Triangle, 1994. ⇒371
[22] ∗ ∗ ∗ International Olympiad in Informatics (IOI), Little Shop of Flower, 1999. ⇒371
[23] ∗ ∗ ∗ Longest common subsequence problem ⇒370
[24] ∗ ∗ ∗ VisuAlgo, https://visualgo.net/en ⇒363

Received: December 10, 2021 • Revised: December 26, 2021

Acta Univ. Sapientiae, Informatica **13,** 2 (2021) 373–400

DOI: 10.2478/ausi-2021-0018

Nonlinearity of the non-Abelian lattice gauge field theory according to the spectrum of Kolmogorov-Sinai entropy and complexity

Ágnes FÜLÖP
Eötvös Lorand University, Faculty of Informatics
email: fulop@inf.elte.hu

Abstract. The quark-gluon plasma is written by the non-Abelian gauge theory. The dynamics of the gauge SU(2) are given by the Hamiltonian function, which contains the quadratic part of the field strength tensor $F^a_{\mu\nu}$ expressed in Minkowski space. The homogeneous Yang-Mills equations are solved on lattice N^d considering classical approximation, which exhibits chaotic dynamics. We research the time-dependent entropy-energy relation, which can be shown by the energy spectrum of Kolmogorov-Sinai entropy and the spectra of the statistical complexity.

1 Introduction

In order to know the microscopic mechanisms of high-energy physics, non-Abelian gauge theory provides an appropriate theoretical model. It plays an important role in understanding non-equilibrium processes where energy and momentum are in local equilibrium. Within the framework of perturbative quantum field theory the equilibrium and transport processes are studied.

Computing Classification System 1998: F.2.1
Mathematics Subject Classification 2010: 68U20
Key words and phrases: non-Abelian gauge field, lattice field, Kolmogorov entropy, statistical complexity, chaos

The quantum field theory introduction of quark-gluon plasma was described by the Feynman path integral, which was derived using gauge transformations for non-Abelian gauge fields in a continuous case. The solution of the Yang-Mills equation was determined on a lattice by classical approaching, the equations of motion contain a field strength tensor square.

The time evolution of this system showed to be chaotic[35, 3, 27]. This dynamical quantity is well characterized by the Kolmogorov-Sinai entropy depending on energy and time resp. the statistical complexity as a function of energy and entropy.

The relation between average energy and Kolmogorov entropy was first introduced in [21] for pure SU(2) Yang-Mills systems. The finite size extrapolated initial evolution and as a function of the scaled energy was researched in [19]. The Lyapunov exponent was determined by monodromy matrix and extrapolated ($N \to \infty$), the scaling properties were studied at a given time range. The spectra of the maximal Lyapunov exponent were calculated depending on the time and energy. The Kolmogorov-Sinai entropy was investigated by Pesin form. In this article, we derived the spectrum of this quantity as a function of time and energy.

The idea of complexity has been presented several times recently as algorithmic complexity (Kolmogorov) [25], the amount of information about the optimal predicts the future according to the expected past (Crutchfield, Young) [13, 8], finite series complexity (Lempel, Ziv) [29].

Effective entropy was published by P. Grassberger [20], taking into account the combination of order and disorder, regularity, and randomness, since most systems do not have the highest Shannon information[40] (random structure) or the lowest (ordered structures) alone.

The definition of statistical complexity is introduced by R. López-Ruiz, L. Manchini, X. Calbet (LMC) [11, 32, 2] and J. Shiner, M. Davison, P. Landsberg (SDL) [41].

The generalized statistical complexity measure (M. Martin, A. Palestino, O. Rosso) [33] is based on the LMC concept, which describes the finite time series of nonlinear systems together with the associated probability distribution of the dynamic method. It was extended to Tsallis (T_q), Wootters, Rényi (R_q) [38] entropy and Kulbach-Shannon, Kulbach-Tsallis, Kulbach-Rényi, Jensen divergence. Tsallis suggested generalizing the degree of entropy of the famous Shannon-Boltzmann-Gibbs (S_{BGS}) entropy measure[43]. The new entropy function plays an important role together with its associated generalized thermostats (1998). The Euclidean distance was criticized by Wootters [46], who studied this concept in a quantum mechanical context. Since the related

consideration is an internal statistical measure, this concept can be applied to any probability space. Remark that S_{BGS}, T_q, R_q considered as special cases of the (h, φ) entropies [39] for the study of asymptotic probability distributions. These quantities were generalized to quantum information theory [39]. This includes the Neumann entropy [36] and a quantum version of Tsallis' and Renyi entropies, which have been applied for example to the detection of quantum entanglement [10].

In addition, we use information theory tools to analyze complex signals, as entropies, distances, and statistical differences play a crucial role in forecasting, estimation, detection, and transmission processes. This concept has been widely used in the chaotic field [15, 18], symbolic sequences [1], pseudo-random bit generator [22], number system [17].

We consider creating a quantitative statistical metric complexity. It is based on a statistical description of the system imposing on the physical model. Suppose the system has N available states $\{x_1, \ldots, x_n\}$ on a given scale and determine the appropriate probabilities $p_1, \ldots, p_n$ of each state.

The LMC statistical measure of complexity[2] is described in two components, i.e. entropy or information stored in the system and distance from the equilibrium probability distribution, an imbalance giving its corresponding asymptotic properties well-behaved measure of complexity.

This quantity is often characterized by a controversial situation an elaborate dynamics created from relatively simple systems. If the system itself is already enough contained and it is consisted of many different parts, you can support a complex dynamic without it appearance of typical characteristic patterns[26]. Therefore a complex system does not necessarily produce a complex output.

Statistical approaches are easier to implement than to solve equations of motion, and in many cases offer a solution for treatment otherwise difficult problems.

The structure of the article: In the second section we introduced the field theory by Feynman path integral and considered the gluonic part of gauge fields. These quantities play an important role in particle and statistical physics. In the third section, we discretized these quantities on the lattice by parallel transporter and Wilson action. In the fourth we study the maximal Lyapunov spectrum and Kolmogorov-Sinai entropy energy relation. In the fifth section, the statistical complexity is introduced on the probability space and we consider the states sequence along the time-evolution of the gauge field, where the state means all links of the lattice at a given time moment.

2 Path integral

Numerous representations of the field theory are known, Schrödinger wave mechanics resp. Heisenberg operator algebra. One of the best-known methods of quantum production with the Feynman path integral. The advantage of this method is that the analogy between statistical physics and particle physics can be easily drawn. It is well applicable to the formulation of the gauge theory and also accurately reproduces its symmetries. This method is briefly described through quantum mechanics.

2.1 Field theory

2.1.1 Feynman path integral

Using canonical transformations in classical and quantum mechanics, the action is a general function of the canonical transformation. Dirac [14] applied this procedure in the quantum mechanics to the Hamiltonian function H at time t in the q' state, respectively. At the moment T for a system where the transient amplitude is:

$$\langle q'_t|q_T\rangle \sim \exp\left(\frac{i}{\hbar}\int_T^t L\,dt\right). \tag{1}$$

On a finite time interval $T-t$, the range $T-t$ is divided into N infinitesimal time intervals, $t_a = t + a\varepsilon$, $N\varepsilon = T - t$. Let $q_a = q_{t_a}$ for all t_a. We apply the $\int dq|q\rangle\langle q| = 1 = 1$ correlation then it is following:

$$\langle q'_t|q_T\rangle = \int dq_1 dq_2 \dots dq_{N-1} \langle q'_t|q_1\rangle \langle q_1|q_2\rangle \dots \langle q_{N-1}|q_T\rangle .$$

The transient amplitude prescribed by the path integral for infinitesimal time interval δt introduced by Feynman[16]:

$$\langle q'_t|q_{t+\delta t}\rangle = \lim_{\substack{N\to\infty \\ (N\varepsilon=konst.)}} A^N \int \left(\prod_{i=1}^{N-1} dq_i\right)\left(\prod_{i=1}^{N-1} dp_i\right) \times \exp\left(-\frac{i}{\hbar}\int_T^t dt L(q,\dot{q})\right), \tag{2}$$

where A^N is the normalization factor dividing this coefficient by a factor A for each instant of time. This expression is equivalent to the integral of the action

function as follows

$$\equiv \int \mathcal{D}q\mathcal{D}p \exp\left(-\frac{i}{\hbar}S(t,T,q)\right). \tag{3}$$

The boundary conditions are the value of orbit at the initial and at the final moment. The above expression gives the probability amplitude of the particle, assuming that it was at t moment in q' state and at time T was in q state. The transient amplitude is expressed as the sum of each of the possible orbits, which begins in q at time T and ends in q' at time t, weighted by the exponential expression $(-\frac{i}{\hbar}S)$ for each trajectory.

The expression of the transient amplitude for Hamiltonian systems can be described as

$$\left\langle q'_t \right| q_T \left\rangle = \int \cdots \int \mathcal{D}q\mathcal{D}p \exp\left(i\int_T^t d\tau \left[p\frac{dq}{d\tau} - H(p,\langle q\rangle)\right]\right), \tag{4}$$

where $\langle q\rangle$ is the average of q over a given interval.

2.1.2 Relation between statistical physics and particle physics

Statistical mechanics is closely related to the Feynman path integral of quantum mechanics. Creutz showed in 1977 [12] that the transfer matrix method simplifies the problem of quadratic functions operator diagonalization in Hilbert space.

The Lagrange function of the free nonrelativistic particle, which measure is m moving in potential $V(x)$ (imaginaries time lattice):

$$L(x,\dot{x}) = K(\dot{x}) + V(x), \qquad K(\dot{x}) = \frac{1}{2}m\dot{x}^2.$$

The action function of any trajectory is following

$$S = \int dt L\left(\dot{x}(t), x(t)\right), \tag{5}$$

with which we can specify with the path integral:

$$Z = \int [dx(t)] \exp(-S). \tag{6}$$

The integral is for all trajectories $x(t)$. The time component of the lattice is discretized. Investigate the trajectories over the entire τ time interval, which

is decomposed into discrete time slices of length $a = \tau/N$. The coordinate for the i-th time slice is x_i. The time derivative $\dot{x}$ is approximated with the difference of the neighbors:

$$S = a \sum_i \left[\frac{1}{2} m \frac{(x_{i+1} - x_i)^2}{a^2} + V(x_i) \right]. \tag{7}$$

Expression (6) is written with x_i coordinates using the Z integral approximation:

$$Z = \int \left(\prod_i dx_i \right) \exp(-S). \tag{8}$$

Equation (8) is no different then the shape of partition functions in a statistical physical system.

The procedure that leads from the path integral to the expression of the quantum mechanical Hilbert space in three steps is: First, we define the path integral on a time-like lattice. We construct the transfer matrix in Hilbert space. We finally take the logarithm of the transfer matrix, where the linear term expresses the temporal evolution of the system. If the i-th eigenvalue of the transfer matrix is λ_i, then $Z = \sum \lambda_i^N$. Since the number of time slices goes to infinity, therefore, the expression can be characterized by the maximum self-values λ_0:

$$Z = \lambda_0^N \left[1 + O\left(\exp\left[-N \ln\left(\frac{\lambda_0}{\lambda_1} \right) \right] \right) \right].$$

2.2 Gauge fields

Several introductions of the gauge fields are known. The simplest way is an extension of the Abelian gauge theory describing the electromagnetic field. The components of the antisymmetric tensor are electromagnetic fields, which are four-dimensional vectors:

$$F_{\mu\nu} = \partial_\mu A_\nu - \partial_\nu A_\mu \quad \mu, \nu = 0, 1, 2, 3.$$

The space-time indices are denoted by μ, ν, and the group generators by α, β, γ. Yang and Mills [47] proposed (1954) to assign the isospin index to A_μ and $F_{\mu\nu}$:

$$A_\mu \to A_\mu^\alpha \qquad F_{\mu\nu} \to F_{\mu\nu}^\alpha \quad \alpha = 1, \ldots, N,$$

a further antisymmetric term is added to the expression and the shape of $F_{\mu\nu}$ is:

$$F^{\alpha}_{\mu\nu} = \partial_\mu A^{\alpha}_{\nu} - \partial_\nu A^{\alpha}_{\mu} + g_0 C^{\alpha\beta\gamma} A^{\beta}_{\mu} A^{\gamma}_{\nu}, \tag{9}$$

where g_0 is the bare coupling constant, $C^{\alpha\beta\gamma}$ is the structural constant of the Lie algebra of a G Lie group. Here we use only uniter groups, the fundamental representation of the G group. We parameterize the elements of G with the set of generators $g = \exp(i\omega^{\alpha}\zeta^{\alpha})$, where ω^{ζ} is the set of parameters and λ^{α} is the set of Hermitian matrices that generalize the group. The structure constants are defined by the following context:

$$\left[\zeta^{\alpha}, \zeta^{\beta}\right] = iC^{\alpha\beta\gamma}\zeta^{\gamma}.$$

The generators are orthonormal: $\mathrm{tr}(\zeta^{\alpha}\zeta^{\beta}) = \frac{1}{2}\delta^{\alpha\beta}$. The simplest non-Abelian theory uses the $SU(2)$ group, which is generalized by Pauli matrices $\zeta^{\alpha} = \frac{1}{2}\tau^{\alpha}$, $C^{\alpha\beta\gamma} = \varepsilon^{\alpha\beta\gamma}$. The Maxwell equations can be derived from the Lagrange density:

$$\mathcal{L} = \frac{1}{4} F_{\mu\nu} F_{\mu\nu} + j_\mu A_\mu,$$

where j_μ is the external source as the electrodynamic fields. The non-Abelian Lagrange density starts in the same way, except for the amount for isospin and $F_{\mu\nu}$ contains an additional member. The classical equation of motion of electrodynamics is the equation $\partial_\mu F_{\mu\nu} = j_\nu$. In the non-Abelian theory $(D_\mu F_{\mu\nu})^{\alpha} = j^{\alpha}_{\nu}$. Here are the covariant derivatives:

$$(D_\mu F_{\mu\nu})^{\alpha} = \partial_\mu F^{\alpha}_{\mu\nu} + g_0 C^{\alpha\beta\gamma} A^{\beta}_{\mu} F^{\gamma}_{\mu\nu}. \tag{10}$$

The non-Abelian analog of current conservation following

$$D_\mu j_\mu = 0.$$

Second definition of gauge fields uses the local symmetry of the action function. Gauge symmetry of electrodynamics: $A_\mu + \partial_\mu \Lambda$, where the gauge function $\Lambda(x)$ is an arbitrary function of the space-time coordinates. In the case of non-Abelian, A_μ is transformed as follows: $A_\mu \to g^{-1} A_\mu g + \frac{i}{g_0} g^{-1} \partial_\mu g$, where g is an element of a suitably chosen group of gauges. In electrodynamics, this transformation is written by a simple phase: $g(x) = \exp(-ig_0\Lambda(x)))$. This is the so-called $U(1)$ gauge theory of electrodynamics. Then, using the transformation formula given above, the transformation of $F_{\mu\nu}$ can be simply

written: $F_{\mu\nu} \to g^{-1}F_{\mu\nu}g$. The gauge transformation of the covariance derivative can be given in a similar form.

A third possible introduction to gauge theory is phase theory (Mandelstam (1962)[31], Yang (1975)).

In this article we mention the introduction of gauge theories to canonical Hamiltonian formalism following Steven Weinberg (1965) [44].

Group In this article, we apply some basic properties of the invariant integral introduced by Haar [23] in Wilson on compact Lie groups. Haar-measure satisfies the following condition:

$$\int_G f(U)dU = \int_G f(U^{-1})dU.$$

In the case of $G = SU(2)$ the group elements can be parameterized in the following way:

$$U = x^0\mathbf{1} + i\vec{x}\vec{\tau} = \begin{pmatrix} x^0 + ix^3, & x^2 + ix^1 \\ -x^2 + ix^1, & x^0 - ix^3 \end{pmatrix}.$$

The parameters x^i must be sufficient to satisfy the condition:

$$\det U = x^2 = (x^0)^2 + \vec{x}^2 = 1.$$

that specifies the S^3 key. In the case of numerical calculation, we used the quaternion representation x_0, x_1, x_2, x_3, because the runtime is faster and the memory requirement is smaller than the matrix representation.

3 Lattice field theory

Continuous gauge quantities are introduced on a lattice [34]. We consider the Wilson action and the Yang-Mills theory by these discretized quantities.

3.1 Discrete parallel transporter

Consider hypercubic lattice of size a and the regularization of the continuous Euclidean lattice. The scalar field $\phi(x)$ is interpreted on the lattice point. Local gauge transformation is following:

$$\phi(x) \to \phi'(x) = \Lambda^{-1}(x)\phi(x).$$

In this case, the nearest non-zero lattice spacing a must be introduced on the hypercube grid.

The elementary parallel transporters are closely connected by the links b, which connect the neighboring points. Let x be an arbitrary point on the lattice. Nearest neighbour points can be written in the form $x + a\hat{\mu}$, where $\mu = 1, 2, 3, 4$ and $\hat{\mu}$ denotes the μ-th unit vector. The links from x to $x + a\hat{\mu}$ can be denoted by the following ordered pair: $b = (x + a\hat{\mu}, x) \equiv (x, \mu)$. The parallel transporter can be described by the link b:

$$U(b) \equiv U(x + a\hat{\mu}, x) \equiv U_{x\mu} \in G, \tag{11}$$

where G is the gauge group. The link thus introduced satisfies the corresponding properties of the parallel transporter. Arbitrary path $C = b_n \circ b_{n-1} \circ \cdots \circ b_1$ corresponds to the parallel transporter $U(b) = U(b_n) \dots U(b_1) \equiv \prod_{b \in C} U(b)$ on lattice, which describes the link variables. These are denoted by $\{U(b)\}$[30]. Transformation of link variables is following:

$$U'(y, x) = \Lambda^{-1}(y) U(y, x) \Lambda(x),$$

where $\Lambda \in SU(N)$ and the size of matrix is $N \times N$. We define the covariance derivative:

$$D_\mu \phi(x) = \frac{1}{a}(U^{-1}(x, \mu)\phi(x + a\hat{\mu}) - \phi(x)).$$

The term of derivatives are substituted by covariate derivatives in the kinetic expression:

$$\frac{1}{2} \sum_x a^4 D_\mu \phi D_\mu \phi = -a^2 \sum_{\langle xy \rangle} \phi(x) U(x, y) \phi(y) + 4a^2 \sum_x \phi(x)^2.$$

The smallest closed loop on the lattice is called a plaque. A plaque is enclosed by 4 links and it contains the following points: $x, x + a\hat{\mu}, x + a\hat{\mu} + a\hat{\nu}, x + a\hat{\nu}$, denoted by $p = (x; \mu\nu)$. The corresponding parallel transporter can be written in the following form:

$$\begin{aligned} U_p \equiv U_{x;\mu\nu} \equiv \; & U(x, x + a\hat{\nu}) U(x + a\hat{\nu}, x + a\hat{\mu} + a\hat{\nu}) \times \\ & U^\dagger(x + a\hat{\mu} + a\hat{\nu}, x + a\hat{\mu}) U^\dagger(x + a\hat{\mu}, x), \end{aligned} \tag{12}$$

which we call the plaque variables. Wilson's suggestion [24, 45] is to write the theoretical definition of a simple lattice gauge with the plaque variables: $S[U] = \sum_p S_p(U_p)$, that is, the action is summed for all p, i.e. $\sum_p = \sum_x \sum_{1 \le \mu, \nu \le 4}$

means. The action is written on the elementary plaque (showing only one direction):

$$S_p(U_p) = \beta \left\{ 1 - \frac{1}{N} \mathrm{Re} \mathrm{tr} U_p \right\}. \tag{13}$$

3.2 Wilson action, lattice Hamiltonian

Wilson action is gauge invariant quantity because $\mathrm{tr}U'_p = \mathrm{tr}U_p$ is appropriately chosen for group $SU(N)$, further real and positive. We consider the Yang-Mills action by the Wilson action. We introduced the vector potential: $A_\mu(x) = -igA^b_\mu(x)T_b$. Lie-algebra value vector field was defined on the lattice:

$$U(x,\mu) \equiv \exp(-aA_\mu(x)) = 1 - aA_\mu(x) + \frac{a^2}{2} A_\mu(x)^2 + \dots$$

we apply $A_\nu(x + a\hat{\mu}) = A_\nu(x) a\Delta^f_\mu A_\nu(x)$ where $\Delta^f_\mu f(x) = \frac{1}{a}(f(x + a\hat{\mu}) - f(x))$.

The Campbell-Baker-Hausdorff expression:

$\exp(x)\exp(y) = \exp(x + y + \frac{1}{2}[x,y] + \dots)$ therefore we get:

$$U_{x;\mu\nu} = \exp\left(-a^2 G_{\mu\nu}(x)\right), \quad \text{where} \quad G_{\mu\nu}(x) = F_{\mu\nu}(x) + O(a)$$

$$F_{\mu\nu}(x) = \Delta^f_\mu A_\nu(x) - \Delta^f_\nu A_\mu(x) + [A_\mu(x), A_\nu(x)].$$

Therefore

$$1 - \frac{1}{N} \mathrm{Re}\ \mathrm{tr} U_p = 2\mathrm{tr}\mathbf{1} + a^4 \mathrm{tr}(F_{\mu\nu}(x))^2 + O(a^5),$$

where $\mathrm{Re}\ \mathrm{tr}$ means the real value of the trace U_p, since $\mathrm{tr}G_{\mu\nu}(x) = 0$ and $\sum_p \mathrm{tr}(F_{\mu\nu}(x))^2 = \frac{1}{2}\sum_{x;\mu\nu} \mathrm{tr}(F_{\mu\nu}(x))^2$. We get the following expression of the Wilson action:

$$S = -\frac{\beta}{4N} \sum_x a^4 \mathrm{tr} F_{\mu\nu}(x) F^{\mu\nu}(x) + O(a^5). \tag{14}$$

Because, the leading member coincides with the Yang-Mills action for small a if $\beta = \frac{2N}{g^2}$ and g correspond to the bare coupling constant of the lattice theory. We split the action into time-space components

$$S = \frac{2}{g^2} \sum_{p_t} (N - \mathrm{tr} U_{p_t}) - \frac{2}{g^2} \sum_{p_s} (N - \mathrm{tr} U_{p_s}), \tag{15}$$

where g is the continuous limitation value of the coupling constant, the (-) sign is derived from the Minkovski space-time structure. The Taylor series of U_{p_t} is explained in time-dependent term

$$U_{p_t} = U(t)U^{\dagger}(t + a_t) = UU^{\dagger} + a_t U\dot{U}^{\dagger} + \frac{a_t^2}{2} U\ddot{U}^{\dagger} + \ldots,$$

The expressions appear in the Wilson Action:

$$N - \mathrm{tr}U_{p_t} = -\frac{a_t^2}{2}\mathrm{tr}(U\ddot{U}^{\dagger}) \qquad O(a_t^3) \quad \text{correction.}$$

Since $UU^{\dagger} = 1$, trace disappears N. It follows from the first derivative of this term that $\mathrm{tr}(U\dot{U}^{\dagger}) = 0$ and the second derivative is $\ddot{U}U^{\dagger} + 2\dot{U}\dot{U}^{\dagger} + U\ddot{U}^{\dagger} = 0$. Therefore the Hamiltonian lattice action is following:

$$\Delta S_H = \frac{2}{g^2}\left(\frac{a_t^2}{2}\sum_i \mathrm{tr}\left(\dot{U}_i\dot{U}_i^{\dagger}\right) - \sum_{ij}\left(N - \mathrm{tr}\left(U_{ij}\right)\right)\right). \tag{16}$$

The generalized discretized ansatz can be written:

$$S = a_t \sum_t a_s^3 \sum_s L.$$

The scaled Hamilton density is able to write in the following form.

$$a_t H = \frac{2}{g^2}\left(\frac{a_t^2}{2}\sum_{x,i}\mathrm{tr}\left(\dot{U}_{x,i}, \dot{U}_{x,i}^{\dagger}\right) + \sum_{x,ij}\left(N - \mathrm{tr}\left(U_{x,ij}\right)\right)\right), \tag{17}$$

namely

$$H = a_s^3 \sum_s \left(\mathrm{tr}\left(\dot{U}, \frac{\partial L}{\partial \dot{U}}\right) - L\right).$$

On the lattice, the gauge field can be specified by configuring the link variables. The expected value of the quantity denoted by the $\{U(b)\} \equiv U$ and $\Theta(\{U(b)\})$ link variables:

$$\langle\Theta\rangle = \frac{1}{Z}\int \prod_b dU(b)\Theta \exp(-S(U)), \tag{18}$$

where $Z = \int \prod_b dU(b) \exp(-S(U))$ and $S(U)$ are Wilson actions. If we introduce $\phi(x)$ the field "material" is given by the corresponding integral:

$$\langle\Theta\rangle = \frac{1}{Z}\int \prod_b dU(b) \prod_x d\phi(x)\Theta \exp(-S(U,\phi)).$$

In these expressions, the integration measures $dU(b)$, must be chosen to be gauge invariant. During the gauge transformation it is written:

$$U'(x,y) = \Lambda^{-1}(x)U(x,y)\Lambda(y)$$

because the action is invariant: $dU = dU'$, $S(U) = S(U')$.

3.3 Lattice Yang-Mills theory

In the following, we use Hamiltonian formulation of the classical lattice $SU(2)$ gauge theory [4]. The Hamilton function is considering:

$$H' = \frac{g^2 a H}{4} = \sum_{x,i} \frac{a^2}{4} \mathrm{tr}\left(\dot{U}^{\dagger}_{x,i}, \dot{U}_{x,i}\right) + \sum_{x,ij}\left[1 - \frac{1}{2}\mathrm{tr}U_{x,ij}\right],$$

where $U_{x,i}$ is the group element $SU(2)$, this term means the $x+ae_i$ link pointing in the i direction starting at $x = (x_1, x_2, x_3)$ on the lattice. $U_{x,ij}$ denotes the elementary plaque which is expressed by link $U_{x,ij} = U_{x,i}U_{x+i,j}U^{\dagger}_{x+j,i}U^{\dagger}_{x,j}$ lying in the plane stretched by the elementary vectors i and j starting at x. We apply the link variables only in the expressions H:

$$H = \sum_{x,i}\left[\frac{1}{2}\langle \dot{U}_{x,i}, \dot{U}_{x,i}\rangle + \left(1 - \frac{1}{4}\langle U_{x,i}, V_{x,i}\rangle\right)\right], \tag{19}$$

where the complement link variable $V_{x,l}(U)$ is following:

$$V_{x,l} = \frac{1}{4} \sum_{\left(\begin{smallmatrix}(l,s):\{(i,j),(k,j),\\(-i,j),(-k,j)\}\end{smallmatrix}\right)} U_{x+l,s}U^{\dagger}_{x+l+s,-l}U^{\dagger}_{x+l,-l}, \quad \text{where}$$

i, j, k are the unit vectors of the three-dimensional lattice.

In the gauge field section (2.2) we introduced the quaterinon representation, which is defined in the following way on a lattice:

$$U = u_0\mathbf{1} + i\vec{\tau}\vec{u} \qquad U = \begin{pmatrix} u_0 + iu_3, iu_1 + u_2 \\ iu_1 - u_2, u_0 - iu_3 \end{pmatrix}.$$

The equations of motion are derived from the Hamiltonian function:

$$\begin{aligned} \dot{U} &= P, \\ \dot{P} &= V - \langle U, V\rangle U - \langle P, P\rangle U, \end{aligned} \tag{20}$$

where $\langle P, P\rangle = \frac{1}{2}\sum_j P_j P^j$.

The lattice equation of motion [5] follows:

$$\begin{aligned} U_{t+1} - U_{t-1} &= 2h(P_t^\star - \varepsilon U_t^\star), \\ P_{t+1} - P_{t-1} &= 2h(V(U_t^\star) - \mu U_t^\star + \varepsilon P_t^\star), \quad \text{where} \end{aligned} \tag{21}$$

$$\varepsilon = \frac{\langle U_t^\star, P_t^\star\rangle}{\langle U_t^\star, U_t^\star\rangle}, \qquad \mu = \frac{\langle V(U_t^\star), U_t^\star\rangle + \langle P_t^\star, P_t^\star\rangle}{\langle U_t^\star, U_t^\star\rangle}, \quad \text{and}$$

$$U_t^\star = aU_{t+1} + bU_t + cU_{t-1}.$$

The quantities ε, μ denote the Lagrange multipliers. The energy of the Hamiltonian system was constant and Gaussian law is satisfied [7] during the movement. A periodic boundary condition was used to solve the system of equations. The color charge was defined following:

$$\Gamma_i = \sum_{l_+} P_l U_l^\dagger - \sum_{l_-} U_l^\dagger P_l, \quad i = 1, \ldots N$$

The measure of change is written by this term:

$$\dot{\Gamma}_i = \sum_{l_+} (VU^\dagger - \langle V, U\rangle \mathbf{1}),$$

where $P_1 = QU_1$ and $P_1 = QU_1$ and $P_n = U_{n-1}^\dagger P_{n-1} U_n$, $1 < n < N$. The condition of neutrality formulated as

$$Q - F^\dagger Q F = 0, \mathrm{tr} Q = 0,$$

from which it follows

$$Q = \frac{q}{2}(F^\dagger - F), \quad \text{where} \quad F = \prod_{i=1}^{N-1} U_i \quad \text{oriented product}$$

the initial color charge is Q and the final state is $-F^\dagger Q F$.

4 Nonlinearity

In this section, we numerically determined the Lyapunov spectrum on the three-dimensional lattice of the SU(2) Yang-Mills field. The spectra of Kolmogorov-Sinai entropy are studied by the eigenvalues of the monodromy matrix from the classical chaotic dynamics to extrapolate on a lattice with a large size limit.

Monodromy matrix We consider a periodic orbit of the energy E, with initial phase space coordinates ($p = p_0, x = x_0$) and final coordinates ($p = p_0$, $x = x_0$). We study the behavior of the neighborhood path of the periodic orbits, how these trajectories develop in the case of small transverse perturbation.

This means the same situation when considering the deviation of flow on the Poincare surface of the section transverse to the path. Then the relation between the initial $\{\delta y_{0i}, \delta p_{0i}\}$ and final state $\{\delta y_i, \delta p_i\}$ deviation is following:

$$\delta y_i = \sum_{j=1}^{d-1} \left(\frac{\partial y_i}{\partial y_{0i}}\right) \delta y_{0j} + \left(\frac{\partial y_i}{\partial p_{0i}}\right) \delta p_{0j} = \sum_{j=1}^{d-1} A_{ij}\delta_{0j} + C_{ij}\delta p_{0j}$$

and

$$\delta p_i = \sum_{j=1}^{d-1} \left(\frac{\partial p_i}{\partial y_{0i}}\right) \delta y_{0j} + \left(\frac{\partial p_i}{\partial p_{0i}}\right) \delta p_{0j} = \sum_{j=1}^{d-1} C_{ij}\delta_{0j} + D_{ij}\delta p_{0j}$$

It is written by matrix form:

$$\begin{pmatrix} \delta\overline{y} \\ \delta\overline{p} \end{pmatrix} = \begin{pmatrix} \overline{A}, & \overline{B} \\ \overline{C}, & \overline{D} \end{pmatrix} \begin{pmatrix} \delta\overline{y}_0 \\ \delta\overline{p}_0 \end{pmatrix} = \overline{M} \begin{pmatrix} \delta\overline{y}_0 \\ \delta\overline{p}_0 \end{pmatrix}, \tag{22}$$

where $\delta\overline{y}$ and $\delta\overline{p}$ are $1 \times (d-1)$ dimensional column matrices, and $\overline{A}, \overline{B}, \overline{C}, \overline{D}$ are $(d-1) \times (d-1)$ dimensional square matrices where A_{ij}, B_{ij}, C_{ij} D_{ij} matrix elements. This $(2d-2) \times (2d-2)$ dimensional square matrix $\overline{M}$ means the monodromy matrix according to the equation motion [37].

The shape of the monodromy matrix by the lattice equations of motion [19] is following

$$M = \begin{pmatrix} \frac{\partial \dot{U}}{\partial U} & \frac{\partial \dot{U}}{\partial P} \\ \frac{\partial \dot{P}}{\partial U} & \frac{\partial \dot{P}}{\partial P} \end{pmatrix}. \tag{23}$$

We write down each partial derivative by the equation of motion:

$$\frac{\partial \dot{U}^a}{\partial U^b} = 0, \qquad \frac{\partial \dot{U}^a}{\partial P^b} = \delta^{ab},$$

$$\frac{\partial \dot{P}^a}{\partial U^b} = \frac{\partial V^a}{\partial U^b} - \left(\sum_{c=1}^{N} U_c \frac{\partial V^c}{\partial U^b}\right) U^a - V^b U^a - \sum_{c=1}^{N} \left(U_c V^c + P_c P^c\right) \delta^{ab},$$

$$\frac{\partial \dot{P}^a}{\partial P^b} = -2P^b U^a, \quad \text{where}$$

$$\frac{\partial V_k^{\alpha_q}}{\partial U^{\beta_q}} = \sum_{l=1}^{\mathcal{N}} \frac{\partial V_k^{\alpha_q}(U_1, \ldots, U_{\mathcal{N}})}{\partial U_l^{\beta_q}}, \quad \text{where } \mathcal{N} = 12, \quad \alpha_q, \beta_q = 0, 1, 2, 3.$$

The shape of the characteristic equation is then:

$$\det\left[\begin{pmatrix} \mathbf{0} & \mathbf{1} \\ \frac{\partial\dot{P}}{\partial U} & \frac{\partial\dot{P}}{\partial P} \end{pmatrix} - \Lambda_i \mathbf{1}\right] = 0. \tag{24}$$

We showed the stability of the trajectories along the trajectory in the vicinity of any point on the (U, P) phase space. The time evolution of a small $(\delta U, \delta P)$ perturbation is determined by the monodromy matrix. Among the eigenvalues of the stability matrix, real and positive quantities indicate an exponential departure of adjacent trajectories, i.e., motion is unstable. At the long-term limit, the Lyapunov exponents are obtained from the eigenvalues.

4.1 Spectrum of the maximal Lyapunov exponent

We investigated the ergodization of the SU(2) lattice gauge theory due to classical chaotic dynamics [19]. We get a good approximation to the real maximum Lyapunov spectrum by monodromy matrix of time-evolving field configurations. The lattice size was chosen to be $N = 2, 3, 4, 5, 6, 7$. The initial configurations are randomized we choose according to the Haar measure and the total energy constraint.

The Lyapunov exponent L_i is introduced with eigenvalues Λ_i of monodromy matrix:

$$L_i = \lim_{T\to\infty} \frac{\int_0^T \Lambda_i(t)dt}{T} \quad i = 1, \ldots, f, \tag{25}$$

where $\Lambda_i(t)$ is the solution of the characteristic equation:

$$\det[\Lambda_i(t)\mathbf{1} - M(t)] = 0, \tag{26}$$

in which M is the linear stability matrix, f is the number of degrees of freedom. Conservative dynamical systems satisfy the Liouville theorem: $\sum_{i=0}^{f} L_i = 0$. In numerical calculations, we use the definition of the discrete Lyapunov spectrum

$$L_i' = \langle\Lambda_i\rangle^{(n)} = \frac{1}{n}\sum_{j=1}^{n} \Lambda_i(t_{j-1}), \quad i = 1, \ldots, f, \tag{27}$$

where t_j is the time series during the trajectory evolution of the gauge field configuration.

The quantities L_i' are extrapolated to a long-term ($N \to \infty$) limit with fixed time steps. We assumed it converges to the L_i Lyapunov exponent in noncompact configuration space.

The eigenvalues of the monodromy matrix were determined along the time-evolution of a single gauge trajectory which allows us to know the behavior of the Lyapunov spectrum as a function of time.

In the numerical simulation the total number of degrees of freedom $f = 4 \times 3 \times N^3 = 12 \times N^3$, where the group element $SU(2)$ is represented by 4 real quaternions (thus the phase space has a dimension of $2f = 24N^3$). Due to the conditions of survival (unity, orthogonality), the number of physically relevant degrees of independent freedom decreases [6].

The spectrum of the $2f \times 2f$ stability matrix although rare is large enough to determine the eigenvalue with sufficient accuracy. Since it requires $O(f^2)$ memory to calculate eigenvalues, $N = 7$ ($2f = 24N^3 = 8232$ dimensional phase space) was the maximum size of the system, which could be examined by the capacity of the computer, which is due to the fact that the Hamiltonian system is conservative (energy is time-independent).

In the literature, it has been shown that in the semiclassical limit the real-time Hamiltonian dynamics of SU(2) gauge theory exhibits deterministic chaos on a spatial lattice [35]. The largest Lyapunov exponent of the gauge field was calculated as a function of energy density. Numerical integration of the equations of motion has been applied considering the conservation of energy and Gaussian law. The exponential divergence of two trajectories was studied on the lattice gauge field configuration. The gauge-invariant metric is proportional to the absolute local difference in the magnetic energy of two different gauge fields. The nearest neighboring configurations were chosen randomly and along the time-evolution, the distance between the two trajectories increased exponentially until it is saturated. This process is known as the rescaling method.

In this paper, we determine the maximum value of the Lyapunov exponent along with the real-time evolution of a single long trajectory using the monodromy matrix. Our goal is to calculate the spectrum of maximal Lyapunov exponent depending on the energy resp. time and we consider the scaling behavior of this system.

Therefore the first step we extrapolated the real maximal Lyapunov exponent ($N \to \infty$) to the thermodynamical limit from the dataset, which is taken for $N = 2, 3, 4, 5, 6, 7$ at the different energies $g^2aE \in [0.0, 0.7]$ range considering the finite-size scaling.

Figure (1) shows the real maximal Lyapunov exponent's aL_0 dependence on scaling time t/a and scaling energy g^2aE, where a is a lattice size and g means the strong coupling constant (Section 3.).

The scaling of the maximal Lyapunov exponent as a function of scaling energy has been studied [19]. In the past, the research on the scaling behavior of

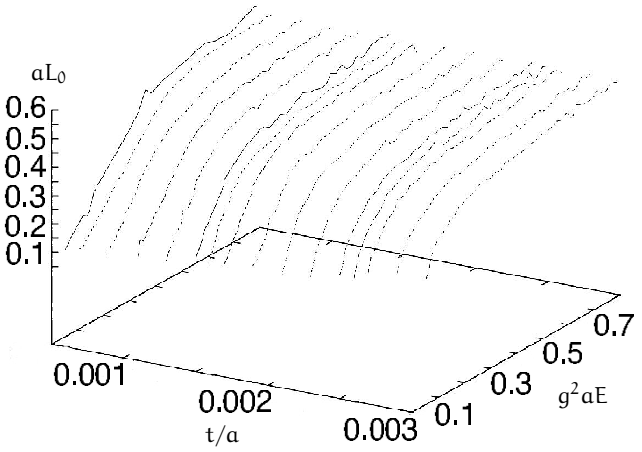


Figure 1: Maximal Lyapunov spectrum aL_0 as a function of the scaling time t/a and scaling energy g^2aE.

maximal Lyapunov exponent has been debated whether it is linear or not in the long-time limit[19]. According to some research results, this would be $L_0 \sim E^{\frac{1}{4}}$ relation. It has been shown that linear scaling at low energy is acceptable using the rescaling method in the long-term boundary case.

In the Figure (1) the scaling of the maximal Lyapunov exponent at short time range $t/a = 0.0005$ satisfies the linear $L_0 \sim E$ relation before the curve saturates. In the long-time limit at $t/a = 0.003$ the scaling becomes logarithmic rather than $L_0 \sim E^{\frac{1}{4}}$ relation [19]. It can be considered that too long a trajectory and the compactness of the configuration space create the calculated eigenvalues, which is the Hamiltonian lattice field theory artifact. In the following, we imply linear scaling.

The extrapolation of the maximum Lyapunov exponent values was plotted on the Figure (1) i.e. the thermodynamic limit $N \to \infty$ at different energies. The finite-size scaling of this quantity to be almost linear:

$$L_0 \sim \frac{1}{\sqrt{f}} \sim N^{-\frac{3}{2}}.$$

This corresponds to sampling ergodic states [9].

4.2 Spectrum of Kolmogorov-Sinai Entropy

The relation between average energy and Kolmogorov-Sinai entropy was first published in [35] for the simple $SU(2)$ Yang-Mills system.

We define the Kolmogorov-Sinai entropy by the term Pesin:

$$h^{KS} = \sum_i L_i \Theta(L_i), \tag{28}$$

where the value of the function $\Theta(x)$ equals 1 if the argument is positive and 0 otherwise. The dimension of the quantity h^{KS} is a rate (1/time). Therefore, the entropy can be given on an N^3 lattice by normalizing quantity:

$$S = \frac{h^{KS}}{\mathrm{Re}(L_0)N^3}. \tag{29}$$

The state equation can be derived from the simulations of the dynamics. The finite-size scaling is extrapolated to infinity ($\frac{1}{N} \to 0$) on the lattice. We consider the Kolmogorov-Sinai entropy as a function of time and energy. This leads to the state equation, which is the relation of entropy-energy $S(E)$ in the thermodynamic limit of infinite volume.

The normalized Kolmogorov-Sinai entropy is derived from the extrapolated L_i data, which depends only slightly on the initial values and scaling linearly according to the energy.

$$\langle S \rangle \sim b \lg(g^2 E a) + c,$$

where $b, c \in \mathbb{R}$. This is an appropriate estimation of the inverse temperature:

$$\frac{1}{T} = \frac{\partial \langle S \rangle}{\partial E} \sim \frac{0.5}{E}$$

Thus the equipartition, i.e. the energy per degree of freedom:

$$E = \frac{1}{2} kT.$$

In the Figure (2) the entropy spectrum S depending on the scaling timet/a and scaling energy $g^2 aE$ is plotted on the ranges $t/a \in [0, 0.004]$, $g^2 aE \in [0, 0.7]$. The closest relation of the entropy S as a function of scaling energy $g^2 aE$ is the ideal gas $S \sim \lg E$ within the interval of scaling time t/a [0.001,0.004]. In the short range of the scaling time t/a $[0, 0.001]$ the lattice artifact appears.

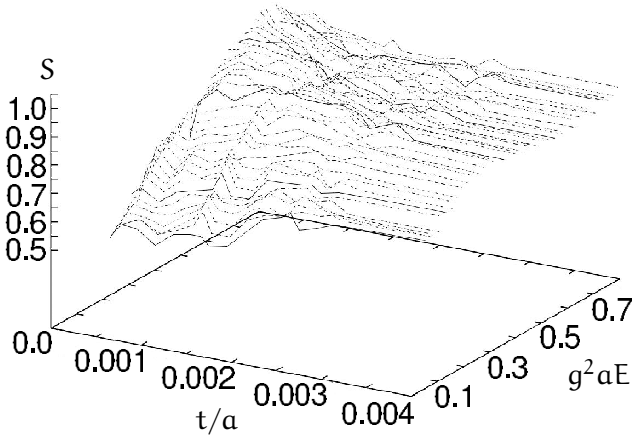


Figure 2: Entropy spectrum S depends on the scaling time t/a and scaling energy g^2aE.

Since the Kolmogorov-Sinai entropy was determined from the Lyapunov exponents with the Pesin form, the lattice artifact experienced in the numerical calculation of the Lyapunov exponents manifests in the Kolmogorov-Sinai entropy spectrum.

It has been shown that the entropy of the SU(2) lattice gauge field has a first-order phase transition [42]. The entropy as a function of energy was expressed by the action on the microcanonical ensemble (section 2.1.2).

In our case lattice SU(2) system $S(E)$ curve would show a first-order two-phase structure containing a break somewhere or crossover (two-phase structure) at the range of time [0.001,0.003] on the interval of the energy [0.1,0.6]. To decide this, we need to filter out lattice artifacts and reduce entropy fluctuations to give a clear answer. The numerical error can be derived by maximal Lyapunov exponents determination, resp. calculation of the eigenvalue of rare matrices.

5 Spectrum of statistical complexity

5.1 Statistical complexity

The family of statistical complexity measures C is introduced by the functional product form $C = H \cdot Q$ for difference disorder H and disequilibrium Q measures on the probabilistic space[33].

The information measure $\mathcal{L}$ is able to be described by a given probability distribution $P = \{p_j, j = 1, \ldots, n\}$, and this quantity corresponds to the measure of uncertainty of a physical system. The amount of disorder H is defined:

$$H[P] = \mathcal{L}[P]/\mathcal{L}_{max}, \tag{30}$$

where $\mathcal{L}_{max} = \mathcal{L}[P_e]$ and $P_e = \{1/n, \ldots, 1/n\}$ is the uniform distribution which maximizes the information measure ($0 \geq H \geq 1$).

To take into account the idea of statistical complexity, a disequilibrium Q needs to be identified.

The measure of this quantity is examined at some distance D to the equal probability distribution P_e.

$$Q[P] = Q_0 D[P, P_e], \tag{31}$$

where Q_0 is a normalization factor ($0 \leq Q \leq 1$). This concept describes the structure of systems as larger than zero if there are possibly more steady states among the possible situations.

Therefore, we take the following functional form for the statistical complexity measure:

$$C[P] = H[P] \cdot Q[P]. \tag{32}$$

This quantity $C[P]$ characterizes the amount of information stored and its disequilibrium in this system altogether [32]. The definition of this concept can be divided into three categories: (c1) this quantity increases monotonically as the function of entropy;

(c2) it is a convex function that contains the maximum value of C_{max} for the probability distribution P_e and the minimum value of C_{min} that occurs at the extreme values of entropy, i.e. $H = 0$ or $H = 1$;

(c3) the third type decreases monotonically with increasing entropy [32].

The two extreme situations can be understood as follows:

(i) Each set of sequences has the same probability distribution. All of them accept the information stored in an equal measure similar to the ideal gas[28].

The probability distribution is the same for all series. All of them accept the information stored in the equivalent measure as the ideal gas [28].

(ii) If we research a system with certain symmetry properties and distance, then this object is able to write by minimum information as a mineral or symmetrical in quantum mechanics or the system is completely disordered.

The statistical complexity is characterized by the scale because it was introduced in a finite system. At each scale of measurement, a new set of available simulated series occurs with its appropriate probability distribution P; so the complexity is changing.

In statistical mechanics, isolated systems often occur that have arbitrary initial conditions and a discrete equal probability distribution [11]. It was concluded that in the case of time-evolving isolated systems and their statistical complexity, the measurements should not take arbitrary values in the C_{LMC} as a function of H. These constrain the bounds of complexity to certain limits of minimum and maximum value.

We use the Shannon entropy measure and Euclidean distance on the probability space as the statistical complexity was investigated by Lopez-Ruiz, Manchini, and Calbet (LMC)[32].

Information measure We consider the Shannon logarithmic information measure on the $P \equiv \{p_1, \ldots, p_n\}$ discrete probability distribution in this article as follows:

$$\mathcal{L}[P] = -\sum_{j=1}^{n} p_j \log(p_j) \tag{33}$$

The maximal value $\mathcal{L}_{max}$ is calculated by the uniform probability $P_e = \left\{\frac{1}{n}, \ldots, \frac{1}{n}\right\}$ fulfilling this criterion $\sum_{j=1}^{n} p_j = 1$ so, $\mathcal{L}_{max} = \ln n$. If $\mathcal{L}[P] = 0$, it means that the possible outcomes j whose probabilities are given by p_j will currently take place. The knowledge of the advantaged process is corresponded by the probability distribution, in this case, is maximal. Anyway this quantity turns into largest for a uniform distribution, when $\mathcal{L}[P] = S_{max}$. These two extreme criteria correspond to the (i) perfect order and (ii)maximum randomness as trivial ones.

Disequilibrium Evidently, the Euclidean statistical distance is taken to give the quantity D, i.e., the quadratic distance between the probability distributions of each state to the equiprobability. If D means the Euclidean norm in

$\mathbb{R}^n$, we find

$$D_E[P, P_e] = \| P - P_e \| = \sum_{i=1}^{n} (p_i - p_e)^2, \tag{34}$$

where $p_e = 1/n$. The maximum disequilibrium is gained for overwhelming simulation sequences with $p_i \sim 1$ and $D \to 1$ for increasing n, as long as this quantity disappears $D \sim 0$ for $p_i \sim 1/n$ for all i. In other probability distribution, the value of the disequilibrium D will vary between these two extreme rates. Then, the expression of the normalization factor of the Euclidean statistical distance fulfills $Q_0 = \frac{n}{n-1}$.

5.2 Complexity of the lattice Yang-Mills equation

In the section (5.1) we introduced the statistical complexity which is based on the probability distribution providing a statistical estimation of the series of dynamical systems. There are n finite different elements on the sequence $\{x_1, x_2, \ldots, x_n\}$ corresponding to the set of discrete probability distribution $P \equiv \{p_1, p_2, \ldots, p_n\}$, where $p_i := P(x_i)$, $(\sum_{i=1}^{n} p_i = 1)$, and $p_i > 0$ for all i.

We study the real-time evolution of the gauge field by the Yang-Mills equation on the lattice. Random initial values are chosen which fulfill the constraint (unitarity, orthogonality, and energy). The length of trajectory is taken as $n = 10000$, the subsequent along the orbit is $m = 2$. The lattice size was chosen $N = 2, 3, 4, 5, 6, 7$.

The state of the gauge field at time t contains all $U_{x,i}$ links on a lattice of size a. The number of links is $dim * N^3$. The lattice gauge field configuration characterizes the state at a given time instant by the links altogether.

The value of entropy (30), disequilibrium (31), and the statistical complexity (32) can be calculated by the simulation unambiguously. Since the probability distribution of element is discontinuous in three-dimensional lattice gauge space, some complexity and disequilibrium values do not appear for certain entropy quantities.

In the Figure (3) the complexity C as a function of scaling energy g^2aE and entropy H is presented and the lattice size is $N = 7$. The spectrum of complexity C was calculated for 8 different energy values, $g^2aE = 0.075, 0.11, 0.17, 0.22, 0.33, 0.4, 0.5, 0.7$. The spectrum of complexity C is finite and limited but not necessarily a unique function of entropy H and there exists a convex boundary and larger internal structure between the minimal value C_{min} and the maximal value C_{max} for different energy range [0.075,08]. The minimal and maximal boundary is increasing as the energy is growing.

The eight different spectra of the statistical complexity C as a function of entropy H and energy g^2aE are determined with the same dynamics, i.e. their internal structure leads to a similar probability distribution along time-evolution.

The inner structure can be seen better in Figure (4), where the complexity C dependence on the entropy H is shown for eight different energy rates. The value of the complexity C becomes to zero at $H \sim 0$ and $H \sim 1$ and the curve is convex on the interval $H \in [0, 1]$. This behavior of complexity belongs to a class (c2). In the immediate neighbour of the $C_{max} \sim 0.07$ values for entropy $H \sim 0.5$, i.e. near to the equilibrium distribution P_e, the values of complexity are more strongly scattered than in the case of $H \sim 0$ or $H \sim 1$.

As we have seen in these Figures their lower boundary C_{min} shows slightly scattered curves with decreasing entropy values, where the maximum value of each curve increases in proportion to the energy in the range entropy H [0.5,1.0]. The upper bound values of the complexity C are widely scattered in the neighbor of equilibrium distribution P_e. On the interval of the entropy [0,0.5], the figure does not show any internal structure, where C_{max} and C_{min} belonging to the dynamics of the Yang-Mills system assume almost the same value.

In the Figure (5) the complexity C as a function entropy H and disequlibrium D is plotted. Because the number of points on a long trajectory is finite, C is a function H shows scaling behavior, i.e., the bigger complexity appears at less entropy with a larger discrete probability distribution. Due to the symmetry SU(2) of the non-Abelian gauge field and the constraint of the total energy and Gaussian law, the system does not reach all states of phase space. In constract to the ideal gas [28], where all state of phases space was available, the internal structure evenly filled in the range between the C_{max} and C_{min} boundary.

The statistical complexity of the non-Abelian gauge theory was studied for a long time evolution along the trajectory n. It is showed internal structure in the immediate vicinity of the equilibrium distribution P_e, the further research allows us to narrow the energy range to be examined for the immediate vicinity of the entropy, because the $S(E)$ curve would present a first-order two-phase structure i.e. having a break somewhere within a certain energy range, that the lattice artifact could be filtered out.

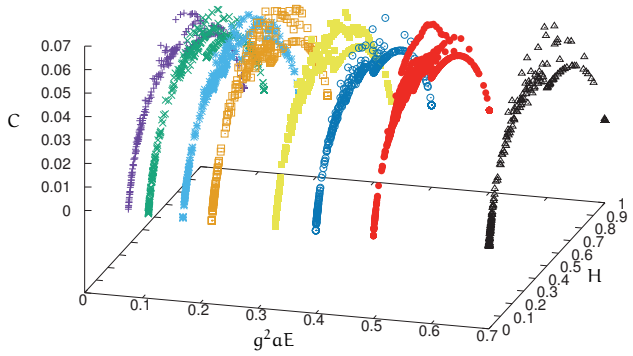


Figure 3: left:Complexity spectra C as a function of the H and $0 < g^2aE < 1$ (0.075,0.11,0.17,0.22,0.33,0.4,0.5,0.7) and the lattice size $N = 7, m = 2$.

6 Summary

In this article, we considered the Hamiltonian function on lattice gauge theory in especially the maximal real Lyapunov spectrum of the non-Abelian gauge theory. The spectra of Kolmogorov-Sinai entropy were studied as a function of energy and lattice size approaching the thermodynamical limit for SU(2) lattice gauge theory. Long time evolution of the equation of motion of gauge fields was characterized by statistical complexity in a probability space. The inner structure of this quantity as a function of entropy allows a more accurate determination of the phase transition in non-Abelian SU(2) lattice space theory using a monodromy matrix with appropriate parameter range on growing lattice size by eliminating the effect of the lattice artifact.

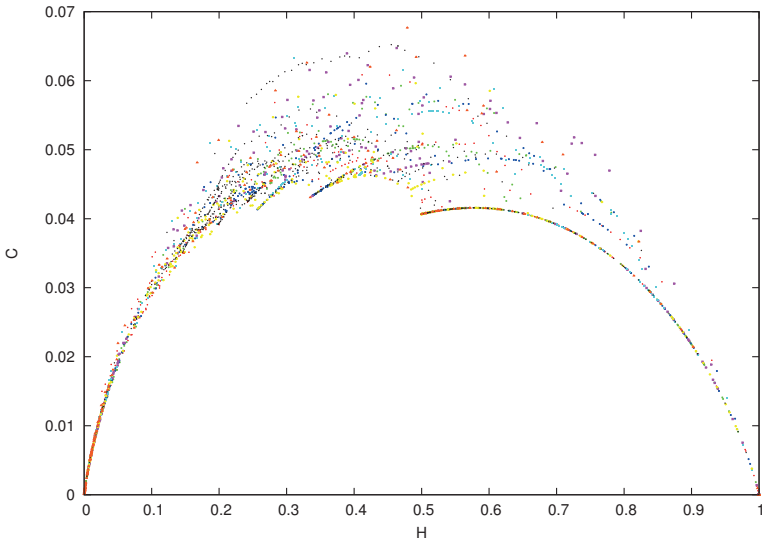


Figure 4: C depends on H on the lattice size $N = 7, m = 2$ for eight different energy rates.

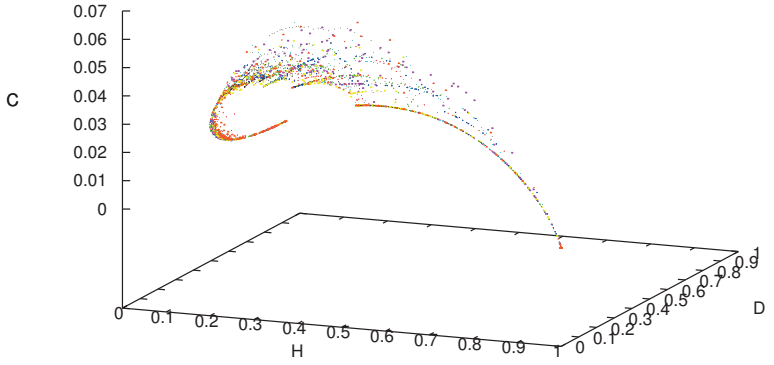


Figure 5: left:Complexity C as a function of the H and D, $0 < g^2 aE < 1$ on the lattice size $N = 7, m = 2$ for eight different energy rates.

References

[1] C. Adami, N. T. Cerf, Physical complexity of symbolic sequences, *Physica D* **137** (2000) 62–69. ⇒375

[2] C. Anteneodo, A.R. Plastino, Some features of the López-Ruiz-Mancini-Calbet (LMC) statistical measure of complexity *Physics Letters A* **223** (1996) 348–354. ⇒374, 375

[3] V. Bannur, Statistical mechanics of Yang-Mills classical mechanics, *Phys. Rev. C* **72** (2005) 024904. ⇒374

[4] T. S. Biró, C. Gong, B. Müller, A. Trayanov, Hamiltonian dynamics of Yang-Mills fields on a lattice *Int. Journ. of Modern Phys. C* **5** (1994) 113–149. ⇒384

[5] T. S. Biró, Conserving algorithms for real-time nonabelian lattice gauge theories *Int. Journ. of Modern Phys. C***6** (1995) 327–344. ⇒385

[6] T. S. Biró, S. G. Matinyan, and B. Müller, *Chaos and Gauge Field Theory* World Scientific, Singapore, 1995. ⇒388

[7] T. S. Biro, A. Fülöp, C. Gong, S. Matinyan, B. Müller, A. Trayanov, Chaotic dynamics in classical lattice field theories, *Lecture Notes in Physics* (1997) 164–176. ⇒385

[8] G. Boffetta, M. Cencini, M. Falcioni, A. Vulpiani, Predictability: a way to characterize complexity *Phys. Reports* **356** (2002) 367–474. ⇒374

[9] J. Bolte, B. Müler, and A. Schafer, Ergodic properties of classical SU(2) lattice gauge theory *Phys. Rev. D* **61** (2000) 054506. ⇒389

[10] G. M. Bosyk, S. Zozor, F. Holik, M. Portesi, P. W. A. Lamberti, A family of generalized quantum entropies: Definition and properties *Quantum Inf. Process.* **15** (2016) 3393–3420. ⇒375

[11] X. Calbet and R. López-Ruiz, Tendency towards maximum complexity in a nonequilibrium isolated system *Phys. Rev. E* **63** (2001) 066116. ⇒374, 393

[12] M. Creutz, Gauge fixing, the transfer matrix, and confinement on a lattice *Phys. Rev. D* **15** (1977) 1128. ⇒377

[13] J. P. Crutchfield, K. Young, Inferring statistical complexity*Phys. Rev. Lett.* **63** (1989) 105. ⇒374

[14] P. A. M. Dirac, On the analogy between classical and quantum mechanics *Rev. Mod. Phys.* **17** (1945) 195. ⇒376

[15] G.L. Ferri, F. Pennini, A. Plastino, LMC-complexity and various chaotic regimes *Physics Letters A* **373** (2009) 2210–2214. ⇒375

[16] R. P. Feynman, Space-Time Approach to Non-Relativistic Quantum Mechanics, *Modern Physics* **20.** 2 (1948) 367. ⇒376

[17] A. Fülöp, Statistical complexity and generalized number system, *Acta Univ. Sapientiae, Informatica* **6,** 2 (2014) 230–251. ⇒375

[18] A. Fülöp, Statistical complexity of the time dependent damped L84 model *Chaos* **29** (2019) 083105. ⇒375

[19] A. Fülöp, T. S. Biró, Towards the equation of state of a classical SU(2) lattice gauge theory, *Phys. Rev. C* **64** (2001) 064902. ⇒374, 386, 387, 388, 389

[20] P. Grassberger, Toward a quantitative theory of self-generated complexity *Int. Journ. Theor. Phys.* **25** (1986) 907–938. ⇒374
[21] C. Gong, Lyapunov spectra in SU(2) lattice gauge theory *Phys. Rev. D* **49** (1994) 2642. ⇒374
[22] C. M. Gonzalez, H. A Larrondo, O. A. Rosso,, Statistical complexity measure of pseudorandom bit generators *Physica A* **354** (2005) 281. ⇒375
[23] A. Haar, Der Massbegriff in der Theorie der kontinuierlichen Gruppen *Ann. Math.* **34** (1933) 147. ⇒380
[24] J. Kogut, L. Susskind, Hamiltonian formulation of Wilson's lattice gauge theories *Phys. Rev. D* **11** (1975) 395–408. ⇒381
[25] A. N. Kolmogorov, Entropy per unit time as a metric invariant of automorphism *Doklady of Russian Academy of Sciences* **124** (1959) 754–755. ⇒374
[26] A-M. Kowalski, M-T. Martin, A. Plastino, O-A. Rosso, M. Casas, Distances in probability space and the statistical complexity setup *Entropy* **13** (2011) 1055–1075. ⇒375
[27] V. Kuvshinov, A. Kuzmin, Deterministic chaos in quantum field theory, *Prog. Theor. Phys. Suppl.* **150** (2003) 126–135. ⇒374
[28] P. T. Landsberg, J. S. Shiner, Disorder and complexity in an ideal non-equilibrium Fermi gas *Phys. Lett. A* **245** (1998) 228. ⇒392, 393, 395
[29] A. Lempel, J. Ziv, On the complexity of finite sequences, *IEEE Trans. Inform. Theory* **22** (1976) 75–81. ⇒374
[30] G. Mack, Physical principles, geometrical aspects, and locality properties of gauge field theories, *Fortsch. Phys.* **29** (1981) 135. ⇒381
[31] S. Mandelstam, Quantum electrodynamics without potentials *Ann. Phys.* **19** (1962) 1. ⇒380
[32] M. T. Martin, A. Plastino, O. A. Rosso, Statistical complexity and disequilibrium *Phys. Lett A* **311** (2003) 126. ⇒374, 392, 393
[33] M. T. Martin, A. Plastino, O. A. Rosso, Generalized statistical complexity measures: Geometrical and analytical properties *Physica A* **369** (2006) 439–462. ⇒374, 392
[34] I. Montvay, G. Münster, *Quantum fields on a lattice*, Cambridge University Press, Cambridge CB2 1RP, 1994. ⇒380
[35] B. Müller, A. Trayanov, Deterministic Chaos on Non-Abelian Lattice Gauge Theory, *Phys. Rev. Letters* **68** 23 (1992) 3387–3390. ⇒374, 388, 390
[36] J. von Neumann, Thermodynamik quantenmechanischer Gesamtheiten, *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen* **1927** (1927) 273–291. ⇒375
[37] L. E. Reichl, *The Transition to Chaos*, *Springer-Verlag*, 1992. ⇒386
[38] A. Rényi, On measures of entropy and information. In *Proc. of the 4th Berkeley Symposium on Mathematics, Statistics and Probability* Neyman, J., Ed.; University of California Press: Berkeley, CA, USA, (1961) 547–561. ⇒374
[39] M. Salicrú, M. L. Menéndez, D. Morales, L. Pardo, Asymptotic distribution of $(\mathfrak{h}, \varphi)$-entropies *Commun. Stat. Theory Meth.* **22** (1993) 2015–2031. ⇒375

[40] C. E. Shannon, A mathematical theory of communication *Bell Syst. Tech. J.* **27** (1948) 379–423. ⇒374
[41] J-S. Shiner, M. Davison, P-T. Landsberg, *Phys. Rev. E* **59,** 2 (1999) 1459–1464. ⇒374
[42] D. R. Stump, Entropy of the SU(2) lattice gauge field, *Phys. Rev. D* **36** (1987) 520. ⇒391
[43] C. Tsallis, Possible generalization of Boltzmann–Gibbs statistics, *J. Stat. Phys.* **52** (1988) 479–487. ⇒374
[44] S. Weinberg, *The quantum theory of fields* Cambridge Univ. Press 1996. ⇒380
[45] K. G. Wilson, Confinement of quarks, *Phys. Rev. D* **10** (1974)2445. ⇒381
[46] W. K. Wootters, Statistical distance and Hilbert space, *Phys. Rev. D* **23** (1981)357. ⇒374
[47] C. N. Yang, Conservation of isotopic spin and isotopic gauge invariance, *Phys. Rev.* **96(1)**(1954)191. ⇒378

Received: November 30, 2021 • Revised: January 10, 2022

Acta Universitatis Sapientiae

The scientific journal of Sapientia Hungarian University of Transylvania publishes original papers and surveys in several areas of sciences written in English.
Information about each series can be found at
http://www.acta.sapientia.ro.

Acta Universitatis Sapientiae, Informatica

Each volume contains two issues.


Sapientia University


Sciendo by De Gruyter


Scientia Publishing House

ISSN 1844-6086

http://www.acta.sapientia.ro

Information for authors

Acta Universitatis Sapientiae, Informatica publishes original papers and surveys in various fields of Computer Science. All papers are peer-reviewed.

Papers published in current and previous volumes can be found in Portable Document Format (pdf) form at the address: http://www.acta.sapientia.ro.

The submitted papers should not be considered for publication by other journals. The corresponding author is responsible for obtaining the permission of coauthors and of the authorities of institutes, if needed, for publication, the Editorial Board is disclaiming any responsibility. Submission must be made by email (acta-inf@acta.sapientia.ro) only, using the LaTeX style and sample file at the address http://www.acta.sapientia.ro. Beside the LaTeX source a pdf format of the paper is necessary too.

Prepare your paper carefully, including keywords, ACM Computing Classification System codes (http://www.acm.org/about/class/1998) and AMS Mathematics Subject Classification codes (http://www.ams.org/msc/).

References should be listed alphabetically based on the Intructions for Authors given at the address http://www.acta.sapientia.ro.

Illustrations should be given in Encapsulated Postscript (eps) format.

Contact address and subscription:
Acta Universitatis Sapientiae, Informatica
RO 400112 Cluj-Napoca
Str. Matei Corvin nr. 4.
Email: acta-inf@acta.sapientia.ro

Printed by F&F INTERNATIONAL
Director: Enikő Ambrus

Supported by:

ISSN 1844-6086
http://www.acta.sapientia.ro