Quarterly Journal of
Optimization In Soft Computing

Vol. 2, Issue 1, Spring 2024

Journal of Optimization of Soft Computing (JOSC)

Vol. 2, Issue 1, pp: (1-7), Spring-2024

Journal homepage: https://sanad.iau.ir/journal/josc

Research paper

Bicyclic graphs with minimum and maximum forgotten and inverse degree indices

Mohammad Ali Manian, Shahram Heidarian* and Farhad Khaksar Haghani

Department of Mathematics, Shahrekord Branch, Islamic Azad University, Shahrekord, Iran

Article Info

Article History:

Received: 2024/03/02

Revised: 2024//05/22

Accepted: 2024/06/09

Keywords:

graph transformations, bicyclic graphs, forgotten index, inverse degree index

**Corresponding Author's Email Address: Heidarianshm@gmail.com*

Abstract

For a connected simple graph G, the inverse degree index and forgotten index are defined as $F(G) = \sum_{uv \in E(G)}[d_u^2 + d_v^2]$ and $ID(G) = \sum_{u \in V(G)} \frac{1}{d_u}$ respectively, where d_u represents the degree of vertex u in G. In this paper, we use some graph transformations and determine the minimum and maximum values of forgotten index and inverse degree index on the class of bicyclic graphs and characterize their corresponding extremal graphs.

1. Introduction

During the last years a large number of topological indices were introduced and found some applications in applied and numerical chemistry [8, 11, 15, 16 and 19]. A topological index is a type of molecular descriptor that is calculated based on the molecular graph of a chemical compound. Topological indices are major invariant to characterize some properties of the graph of a molecule. Several topological indices have been defined and many of them have been found applications as means to model chemical, pharmaceutical and other properties of molecules. Topological indices are generally classified into three kinds: degree-based indices [4, 5, 10], distance-based indices [2] and spectrum-based indices [3]. Topological indices can be used as simple numerical descriptors to compare chemical, physical or biological parameters of molecules in Quantitative Structure Property Relationships (QSPR) and in Quantitative Structure Activity Relationships (QSAR).

All graphs considered in this paper are assumed to be simple and connected. The vertex set and edge set of a graph G are denoted by $V(G)$ and $E(G)$, respectively. The number of neighbors of a vertex u is said to be the degree of u and will be denoted by $d_G(u) = d_u$. A vertex $v \in V(G)$ is said to be pendant, if its neighborhood contains exactly one vertex, i.e. $d_v = 1$. If e is an edge connecting the vertices u and v, then we write $e = uv$. If n is the order of G, then the $n - tuple$ $D = (d_1, d_2, \ldots, d_n)$ of vertex degrees, where $d_i \geq d_{i+1}$ for each $i = 1, 2, \ldots, n$, is called the degree sequence of G. A path P from u_1 to u_i in G is a sequence $P : u_1, e_1, u_2, \ldots, e_{i-1}, u_i$ of alternating vertices and edges such that for every $j = 1, \ldots, i$, e_j is an edge incident with vertices u_j and u_{j+1} and the length of P is $i - 1$. If $u_1 = u_i$, then P is a cycle. A path P is simple, if no vertex occurs twice in P. In addition, P_n and C_n denote the path and the cycle on n vertices, respectively. Let G be a simple graph with p vertices and q edges. If

G has n components, then $\gamma = \gamma(G) = q - p + n$ is called the cyclomatic number of G. A connected graph with $\gamma = 0$ is said to be a tree and graphs with $\gamma = 1, 2$ are called unicyclic and bicyclic, respectively.

The study of topological indices goes back to the seminal work by Wiener [17] in which he used the sum of all shortest path distances nowadays known as the wiener index of a graph. One of the oldest graph invariants is the Zagreb indices first introduced in [9] by Gutman and Trinajstic. For a graph G the inverse degree index was introduced in [6] as

$$ID(G) = \sum_{uv \in E(G)} (\frac{1}{d_u^2} + \frac{1}{d_v^2}) = \sum_{u \in V(G)} \frac{1}{d_u}.$$

In 2006, Zhang [18] introduced the first general Zagreb index of a graph G as follows:

$$M_1^\alpha = M_1^\alpha(G) = \sum_{u \in V(G)} d_u^\alpha.$$

Where α is an arbitrary real number. The forgotten index is just the special case of the first general Zagreb index for $\alpha = 3$. Furtula and Gutman, restudied this index and named it as "forgotten", topological index, or F-index [7]. Thus, F-index is defined as follows:

$$F = F(G) = \sum_{u \in V(G)} d_u^3 = \sum_{uv \in E(G)} (d_u^2 + d_v^2).$$

In 2017, Bozovic et al. [1] used graph transformations and obtained extremal values of total multiplicative sum Zagreb index and first multiplicative sum Zagreb coindex of unicyclic and bicyclic graphs. In [12], we determined extremal values of the inverse degree and the forgotten indices on the class of all unicyclic graphs. In addition, in [13], we found the extremal values of the inverse degree index and forgotten index on the class of unicyclic graphs. In this paper, we use some graph transformations and obtain extremal values of the inverse degree index and forgotten index on the class of all bicyclic graphs.

2. Some graph transformations

Let G be a graph, $V_1 \subseteq V(G)$ and $E_1 \subseteq E(G)$. The subgraph of G obtained by removing the vertices of V_1 and the edges incident with them is denoted by $G - V_1$. Similarly, the subgraph of G obtained by deleting the edges of E_1 is denoted by $G - E_1$. Let $a, b \in V(G)$, $V_1 = \{a\}$ and $E_1 = \{ab\}$. Then the subgraphs $G - V_1$ and $G - E_1$ will be written as $G - a$ and $G - ab$, respectively. In addition, $G \cdot ab$ is a graph, obtained from G by the contraction of edge ab onto vertex a. Finally, if a, b are nonadjacent vertices of G, then $G + ab$ is the graph obtained from G by adding an edge ab.

Theorem 2.1. *Let G and $\widetilde{G}$ be two graphs and $V(G) = V(\widetilde{G}) = V$. Suppose that $u, v \in V(G)$, where $d_G(u) = m$, $d_G(v) = n$, $d_{\widetilde{G}}(u) = m + k$ and $d_{\widetilde{G}}(v) = n - k$, for some $k \geq 0$. If $d_G(a) = d_{\widetilde{G}}(a)$ for each $a \in V \setminus \{u, v\}$, then the following statements hold:*
(i) $k = 0$ or $k = n - m$ if and only if $ID(\widetilde{G}) = ID(G)$,
(ii) $k<n - m$ if and only if $ID(\widetilde{G})<ID(G)$,
(iii) $k>n - m$ if and only if $ID(\widetilde{G})>ID(G)$.

Proof. We set $W = V - \{u, v\}$. Then $d_G(a) = d_{\widetilde{G}}(a)$, for each $a \in W$. Now, by the definition of the inverse degree index, we have

$$\begin{aligned}
ID(G) - ID(\widetilde{G}) &= \sum_{a \in V(G)} \frac{1}{d_a} - \sum_{a \in V(\widetilde{G})} \frac{1}{d_a} \\
&= (\frac{1}{d_G(u)} + \frac{1}{d_G(v)}) \\
&\quad -(\frac{1}{d_{\widetilde{G}}(u)} + \frac{1}{d_{\widetilde{G}}(v)}) \\
&= \frac{1}{m} + \frac{1}{n} - (\frac{1}{m+k} + \frac{1}{n-k}) \\
&= \frac{kn^2 - nk^2 - m^2k - mk^2}{mn(m+k)(n-k)} \\
&= \frac{k(n^2 - m^2) - k^2(n+m)}{mn(m+k)(n-k)} \\
&= \frac{k(n+m)(n-m-k)}{mn(m+k)(n-k)}.
\end{aligned}$$

Now, it is easy to see that (i), (ii) and (iii)are hold. □

Theorem 2.2. *Let G and $\widetilde{G}$ be two graphs and $V(G) = V(\widetilde{G}) = V$. Suppose that $u, v \in V(G)$, where $d_G(u) = m$, $d_G(v) = n$, $d_{\widetilde{G}}(u) = m + k$ and $d_{\widetilde{G}}(v) = n - k$, for some $k \geq 0$. If $d_G(a) = d_{\widetilde{G}}(a)$ for each $a \in V \setminus \{u, v\}$, then the following statements hold:*
(i) $k = 0$ or $k = n - m$ if and only if $F(\widetilde{G}) = F(G)$,
(ii) $k<n - m$ if and only if $F(\widetilde{G})<F(G)$,
(iii) $k>n - m$ if and only if $F(\widetilde{G})>F(G)$.

Proof. By the same way as in the proof of Theorem 2.1 and the definition of the forgotten index, we have

$$\begin{aligned} F(G) - F(\widetilde{G}) &= \sum_{a\in V(G)} d_a^3 - \sum_{a\in V(\widetilde{G})} d_a^3 \\ &= (d_G^3(u) + d_G^3(v)) \\ &\quad -[(d_{\widetilde{G}}^3(u) + d_{\widetilde{G}}^3(v)] \\ &= m^3 + n^3 - (m+k)^3 - (n-k)^3 \\ &= 3n^2k - 3m^2k - 3nk^2 - 3mk^2 \\ &= 3k(n^2 - m^2) - 3k^2(m+n) \\ &= 3k(m+n)(n-m-k). \end{aligned}$$

Now, we can see that (i), (ii) and (iii) are hold. □

A graph transfformation converts the information from the primary graph into a new converted structure. Now, we present several well-known graph transformations [1, 14] that will be used to attain our main results.

Pendant-Path (PP) Transformation. Let G be a nontrivial graph, $u, v \in V(G)$ and $d_G(v) \geq 3$. Suppose that $P_1 : uu_1u_2 \dots u_s$ and $P_2 : vv_1v_2 \dots v_t$ are two paths, that hang on u and v, respectively. Let $\widetilde{G}$ be the graph achieved from G by interconnecting P_1 and P_2 (see Fig. 1). Therefore, the $uu_1...u_sv_1...v_t$ is a path in the new graph $\widetilde{G}$.

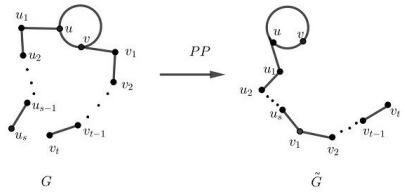

Figure 1. PP Transformation

If we use PP transformation on G, then the degree of u_s increases by $k = 1$ and the degree of v decreases by $k = 1$. Also, the degrees of other vertices in G and $\widetilde{G}$ are equal. Therefore, Theorem 2.1 and Theorem 2.2 show that $ID(\widetilde{G}) < ID(G)$ and $F(\widetilde{G}) < F(G)$.

Contraction to path (EP) Transformation. Let G be a nontrivial graph. Consider two adjacent vertices u and v in G and let $d_G(u) \geq 3$ and $P : uu_1u_2 \dots u_s$ be a path in G. We remove the edge uv and add the new edge u_sv. We denote the new obtained graph by $\widetilde{G}$ (see Fig. 2).

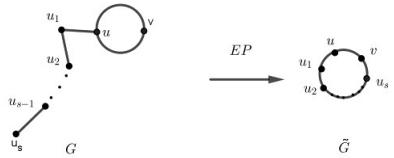

Figure 2. EP Transformation

Let $d_G(u_s) = m$, $d_G(u) = n$ and $n \geq 3$. If we use EP transformation on G, then $d_{\widetilde{G}}(u_s) = m + 1$, $d_{\widetilde{G}}(u) = n - 1$, and the degrees of other vertices in G and $\widetilde{G}$ are the same. Now by Theorem 2.1 and Theorem 2.2, $ID(\widetilde{G}) < ID(G)$ and $F(\widetilde{G}) < F(G)$.

Contraction to star (CS) Transformation. Suppose that G is a graph, $u, v \in V(G)$ and u and v have no shared neighbor. Let $d_G(u) = m$ and $d_G(v) = n$, where $m \geq n \geq 2$. If $e = uv$, we show $(G.e) + uv$ by $\widetilde{G}$ (Fi. 4).

Let u and v be two vertices of a graph G with $d_G(u) = m$ and $d_G(v) = n$. If we use CS transformation on G, then $d_{\widetilde{G}}(v) = n - (n-1) = 1$ and $d_{\widetilde{G}}(u) = m+(n-1)$ and the degrees of other vertices in G and $\widetilde{G}$ are equal. By Theorem 2.1 and Theorem 2.2, if $m \geq n \geq 2$, then $ID(\widetilde{G}) > ID(G)$ and $F(\widetilde{G}) > F(G)$.

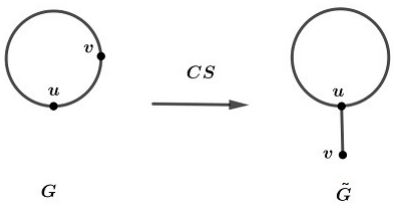

Figure 3. CS Transformation

.

Star translation (ST) Transformation. Let G be a graph, $v \in V(G)$ and $v_1, v_2, \dots, v_t$ be pendant vertices and neighbors of vertex v. If $u \in V(G) - \{v_1, v_2, \dots, v_t\}$, then we show $(G - \{vv_1, vv_2, \dots, vv_t\}) + \{uv_1, uv_2, \dots, uv_t\}$ by $\widetilde{G}$ (Fig. 4). Now, if $d_G(u) = m$ and $d_G(v) = n$, then by Theorem 2.1 and Theorem 2.2 we have

(a) If $t = n - m$, then $ID(\widetilde{G}) = ID(G)$ and $F(\widetilde{G}) = F(G)$,

(b) If $n - m > t$ and $t>0$, then $ID(\widetilde{G}) < ID(G)$ and $F(\widetilde{G}) < F(G)$,

(c) If $t > n - m$, then $ID(\widetilde{G}) > ID(G)$ and $F(\widetilde{G}) > F(G)$.

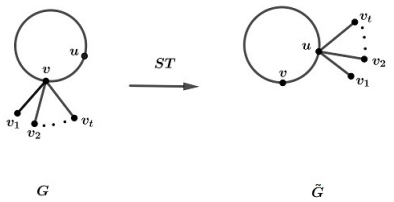

Figure 4. ST Transformation

.

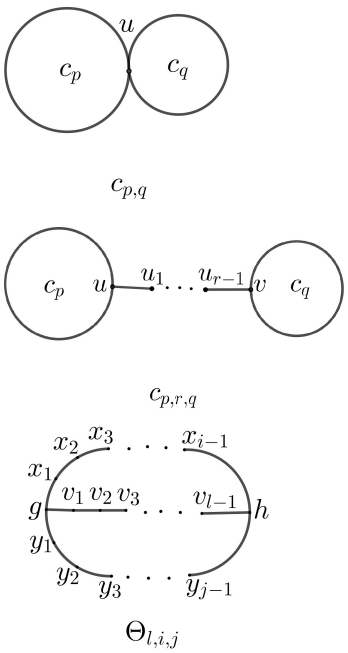

Figure 5. Main subgraphs of bicyclic graphs

.

3. Maximal and minimal values of inverse degree and forgotten indices on bicyclic graphs

In this section, we achieve extremal graphs with respect to the inverse degree and the forgotten indices on the class of all bicyclic graphs, by considering the main subgraphs of the bicyclic graphs.

If G is a bicyclic graph, then G possesses two independent cycles and these cycles are denoted by C_p and C_q as in [16]. Now, one of the following cases will be occurred:

(I) The subgraph C_p and C_q in graph G possess exactly one shared vertex u.

(II) The subgraph C_p and C_q in graph G are connected by a path of length r, where $r>0$.

(III) The subgraph C_{l+i} and C_{l+j} in G, posses a shared path of length l, where $0 < l \leq min\{i, j\}$.

The subgraphs $C_{p,q}$, $C_{p,r,q}$ and $\Theta_{l,i,j}$, depending on the previous three cases, are called main subgraphs of G, respectively (see Fig. 5).

If G is a bicyclic graph, we transform each tree of G into a path by frequentative use of PP transformation. Then, we transform all the paths with the same transformation into a unique path such as $P : w_1w_2 \dots w_t$. Thereupon, we acquire a bicyclic graph $\widetilde{G}$ such that the main subgraph of $\widetilde{G}$ is one of the graphs $C_{p,q}$, $C_{p,r,q}$ or $\Theta_{l,i,j}$ and it has a unique pendent path P attached at one of its vertices. According to the main subgraph of $\widetilde{G}$, we consider the following three cases:

$Case\ A$: **The main subgraph of $\widetilde{G}$ is $C_{p,q}$**

If the path P was connected to the vertex u, then we displace P and paste it to a vertex in C_p or C_q except u. Thus, we assume that P is attached to the vertex s, where $s \neq u$. Now, if the EP transformation is used for s and one of its neighbors of degree 2, we obtain a graph of type I, where the length of one of its cycles is increased by the length of P as Fig. 6. Also, we show that EP transformation decreases the inverse degree and the forgotten indices.

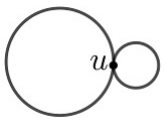

Figure 6. Bicyclic graph φ_1 of type I

.

$Case\ B$: **The main subgraph of $\widetilde{G}$ is** $C_{p,r,q}$

If the path P was connected to the vertices $\{u, u_1, u_2, \ldots, u_{r-1}, v\}$, then we displace and paste it to a vertex s in C_p or C_q except u and v. Now, if the EP transformation is used for s and one of its neighbors of degree 2, then we obtain a graph of type II, where the length of one of its cycles is increased by the length of P. In addition, the vertices $\{u_1, u_2, \ldots, u_{r-1}\}$ can be inserted into one of the cycles C_p or C_q. In this way, the degree sequence of the graph does not change, so the inverse degree and the forgotten indices do not change. Eventually, we achieve a bicyclic graph, where its cycles are linked by uv as in the Fig. 7.

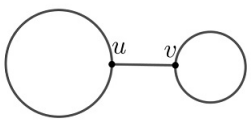

Figure 7. Bicyclic graph φ_2 of type II

.

$Case\ C$: **The main subgraph of $\widetilde{G}$ is** $\Theta_{l,i,j}$

Similar to the previous cases, if the path P was connected to one of the vertices $\{g, v_1, v_2, \ldots, v_{l-1}, h\}$, then we displace and paste it to a vertex s in C_p or C_q except g and h. Now, if the EP transformation is used for s and one of its neighbors of degree 2, then we obtain a graph of type III, where the length of one of its cycles is increased by the length of P. In addition, the vertices $\{v_1, v_2, \ldots, v_{l-1}\}$ can be inserted into one of the cycles C_p or C_q. In this way, the degree sequence of the graph does not change, so the inverse degree and the forgotten indices do not change. Eventually, we achieve a bicyclic graph of type III, where its cycles are shared in the edge gh as Fig. 8.

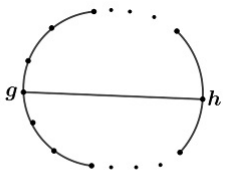

Figure 8. Bicyclic graph φ_3 of type III

.

Therefore, every bicyclic graph G can be converted into one of the graphs φ_1 (Fig. 6), φ_2 (Fig. 7) or φ_3 (Fig. 8). The degree sequences of φ_2 and φ_3 are the same. In the next lemma, we compare the inverse degree and the forgotten indices of φ_1 and φ_2.

Lemma 3.1. *Let φ_1, φ_2 and φ_3 be the simple bicyclic graphs Fig. 6, Fig. 7 and Fig. 8, respectively. Then*

(a) $ID(\varphi_1) > ID(\varphi_2) = ID(\varphi_3)$,

(b) $F(\varphi_1) > F(\varphi_2) = F(\varphi_3)$.

Proof. (a) We can see that the degree sequence of φ_1 is (4,2,2,...,2) and φ_2 is (3,3,2,2,...,2). Then by the definition we have

$$\begin{aligned} ID(\varphi_1) - ID(\varphi_2) &= \sum_{u\in V(G)} \frac{1}{\acute{d}_u} - \sum_{u\in V(G)} \frac{1}{d_u} \\ &= (\frac{1}{2} + \ldots + \frac{1}{2} + \frac{1}{4}) \\ &\quad -(\frac{1}{2} + \ldots + \frac{1}{2} + \frac{1}{3} + \frac{1}{3}) \\ &= \frac{1}{12}. \end{aligned}$$

Thus, $ID(\varphi_1) - ID(\varphi_2) > 0$ and $ID(\varphi_2) < ID(\varphi_1)$, for each $n \geq 5$.

(b)With a similar argument for the forgotten index , we have

$$\begin{aligned} F(\varphi_1) - F(\varphi_2) &= \sum_{u\in V(G)} \acute{d}_u^3 - \sum_{u\in V(G)} d_u^3 \\ &= (2^3 + \ldots + 2^3 + 4^3) \\ &\quad -(2^3 + \ldots + 2^3 + 3^3 + 3^3) \\ &= 18. \end{aligned}$$

Therefore, $F(\varphi_1) - F(\varphi_2) > 0$ and $F(\varphi_2) < F(\varphi_1)$, for each n $n \geq 5$. □

If G is a bicyclic graph, then by iterative use of the CS transformation, G can be converted to a bicyclic graph $\widetilde{G}$, where $\widetilde{G}$ consists of two triangles and some hanging stars and the inverse degree index of $\widetilde{G}$ is greater than the inverse degree index of G and its forgotten index is greater than the forgotten index of G. Now, we repeat ST transformation until G converts to ψ_1 or ψ_2 of Fig. 9. Also, ST transformation increases the inverse degree and the forgotten indices.

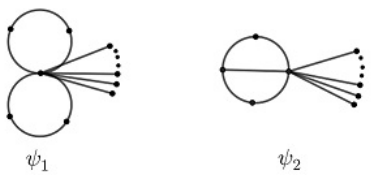

Figure 9. Bicyclic graphs ψ_1 and ψ_2

In the next lemma, we compare the inverse degree and forgotten indices of the graphs ψ_1 and ψ_2.

Lemma 3.2. *Let ψ_1 and ψ_2 be the simple bicyclic graphs Fig. 9. Then*
(a) $ID(\psi_1) < ID(\psi_2)$,
(b) $F(\psi_1) < F(\psi_2)$.

Proof. (a) We can see that the degree sequence of ψ_1 is (n-1,2,2,2,2,1,1,…,1) and ψ_2 is (n-1,3,2,2,1,1,…,1), so by the definition we have

$$\begin{aligned} & ID(\psi_1) - ID(\psi_2) = \sum_{u\in V(G)} \frac{1}{\acute{d}_u} - \sum_{u\in V(G)} \frac{1}{d_u} \\ = \; & (\frac{1}{1}+\ldots+\frac{1}{1}+\frac{1}{2}+\frac{1}{2}+\frac{1}{2}+\frac{1}{2}+\frac{1}{n-1}) \\ - \; & (\frac{1}{1}+\ldots+\frac{1}{1}+\frac{1}{2}+\frac{1}{2}+\frac{1}{3}+\frac{1}{n-1}) \\ = \; & \frac{-1}{3}. \end{aligned}$$

Thus, $ID(\psi_1) - ID(\psi_2) < 0$ and $ID(\psi_1) < ID(\psi_2)$, for each $n \geq 5$.

(b)With a Similar argument above for forgotten index, we have

$$\begin{aligned} & F(\psi_1) - F(\psi_2) = \sum_{u\in V(G)} \acute{d}_u^3 - \sum_{u\in V(G)} d_u^3 \\ = \; & 1^3+\ldots+1^3+2^3+2^3+2^3+2^3+(n-1)^3 \\ & -(1^3+\ldots+1^3+2^3+2^3+3^3+(n-1)^3) \\ = \; & -12. \end{aligned}$$

Therefore, $F(\psi_1) - F(\psi_2) < 0$ and $F(\psi_1) < F(\psi_2)$, for each $n \geq 5$. □

Now, by the above explanations we can prove the main results of this section.

Theorem 3.3. *Let G be a bicyclic graphs of order n, where $n \geq 5$. Then*
(i) $ID(\varphi_2) \leq ID(G) \leq ID(\psi_2)$,
(ii) $F(\varphi_2) \leq F(G) \leq F(\psi_2)$.

Proof. (i) If G is a bicyclic graph, then we show that G can be converted into one of the graphs φ_1, φ_2 or φ_3 of Lemma 3.1, by the PP and EP transformations and these transformations decrease the inverse degree index. Therefore, Lemma 3.1 implies that $ID(\varphi_2) \leq ID(G)$. Similarly, G can be converted into one of the graphs ψ_1 or ψ_2 of Lemma 3.2 by the CS and ST transformations and these transformations increase the inverse degree index. Therefore, Lemma 3.2 shows that $ID(G) \leq ID(\psi_2)$.

Another part can be proved by a similar argument. □

4. Conclusion

In this article, we used some well known graph transformations to obtain the extremal graphs with respect to the forgotten index and the inverse degree index on the class of all bicyclic graphs. We showed that, if G is a bicyclic graph of order n, where $n \geq 5$, then $\frac{n-2}{2} + \frac{2}{3} \leq ID(G) \leq \frac{n^2-5n+5}{n-1} + \frac{4}{3}$ and $8n + 38 \leq F(G) \leq n^3 - 3n^2 + 4n + 42$.

References

[1] V. Bozovic, Z. Kovijanic, G. Popivoda, Extremal values of total multiplicative sum Zagreb index and first multiplicative sum Zagreb coindex on unicyclic and bicyclic graphs, *MATCH Commun. Math. Comput. Chem.* **78** (2017) 417-430.

[2] K. C. Das, S. Sorgun, On Randic energy of graph, *MATCH Commun. Math. Comput. Chem.* **72** (2014) 227-238.

[3] M. Dehmer, F. Emmert-Streid, M. Grabner, A computational approach to construct a multivariate complate graph invariant, *Inf. Sci.* **260** (2014) 200-208.

[4] D. Dimitrov, On structured properties of tree with minimal atom-bond connectivity index II: bound on B_1 and B_2-branches, *Discrete Appl. Math.* **204** (2016) 90-116.

[5] D. Dimitrov, Z. Du, C. M. da Foneseca, On structured properties of tree with minimal atom-bond connectivity index III: trees with pendant path of length three, *Appl. Math. Comput.* **282** (2016) 279-290.

[6] S. Fajtlowicz, On conjectures of Graffiti-II, *Congr. Number* **60**(1978) 187-197.

[7] B. Furtula, I. Gutman, A forgotten topological index, *J. Math. Chem.* **53**(2015) 1184-1190.

[8] V. Gayathiri, A. Manimaran, Computing certain topological indices of cilicate triangle fractal network modeled by the Sierpiński triangle network, *Contemporary Math.* **5** (2024) 2150-2165.

[9] I. Gutman, N. Trinajstic, Graph theory and molecular orbital total $\varphi - electron$ energy of alternanthy-drocarbons, *Chem. Phys. Lett.* **17** (1972) 535-538.

[10] Y. Huang, H. Liu, Bounds of modified Sombor index, spectral radius energy, *AIMS Math.* **6** (2021) 11263-11274.

[11] H. Liu, L. You, Y. Huang, Ordering chemical graphs by Sombor indices and its applications, *MATCH Commun. Math. Comput. Chem.* **87** (2022) 5-22.

[12] M. A. Manian, S. Heidarian, F. Khaksar Haghani, Maximum and minimum values of inverse degree and forgotten indices on the class of all unicyclic graphs, *AKCE Int. J. graphs Comb.* **20** (2023) 57-60.

[13] M. A. Manian, S. Heidarian, F. Khaksar Haghani, On extremal values of total structure connectivity and Narumi-Katayama indices on the class of all unicyclic and bicyclic graphs, *Iranian J. Math. Chem.* **14** (2023) 171-181.

[14] J. Meng, B. Wu, Basic properties of total transformation graphs, *J. Math. study* **34** (2001) 109-116.

[15] S. Noureen, A. Ali, A. Bhatti, A. Alanazi, Y. Shang, Predicting enthalpy of formation of benzenoid hydrocarbons and ordering molecular trees using general multiplicative Zagreb indices, *Heliyon* **10** (2024) 1-10.

[16] V. S. Umadi, A. M. Sangogi, Computing topological indices of certain networks, *Comm. App. Nonlinear Anal.* **31** (2024) 416-429.

[17] H. Wiener, Structural determination of paraffin boiling points, *J. Am. Chem. Soc.* **69** (1947) 17-20. BHK. Xu, I. Gutman, The largest Hosoya index of bicyclic graphs with given maximum degree, *MATCH Commun. Math. Comput. Chem.* **66** (2011) 795-824.

[18] S. Zhang, H. Zhang, Unicyclic graphs with the first three smallest and largest first general Zagreb index, *MATCH Commun. Math. Comput. Chem.* **55**(2006) 427-438.

[19] T. Zhou, Z. Lin, L. Miao, The Sombor index of tree and unicyclic graphs with given maximum degree, *discrete Math. Lett.* **7** (2021) 24-29.

Journal of Optimization of Soft Computing (JOSC)

Vol. 2, Issue 1, pp: (8-18), Spring-2024

Journal homepage: http://josc.rimag.ir/en

Research paper

Introducing two classes of optimal codes derived from one-weight $\mathbb{F}_q\mathbb{F}_q[u]$-additive codes

Narjes Mohsenifar[1] and Sadegh Sadeghi[2*]

[1] *Department of Electrical Engineering, Shahrekord Branch, Islamic Azad University, Shahrekord, Iran.*

[2] *Department of Mathematics, Shahrekord Branch, Islamic Azad University, Shahrekord, Iran.*

Article Info

Article History:

Received: 2024/4/27

Revised: 2024/5/29

Accepted: 2024/6/9

Keywords:

Additive code, Constacyclic code, One-weight code.

**Corresponding Author's Email Address: sadegh.sadeghi@gmail.com*

Abstract

Let $\mathbb{F}_q$ be a finite field with q elements, where $q = p^m$, and $R = \mathbb{F}_q + u\mathbb{F}_q$ denotes the ring $\frac{\mathbb{F}_q[u]}{\langle u^2\rangle}$. For positive integers α and β, a nonempty subset C of $\mathbb{F}_q^\alpha \times R^\beta$ is called an $\mathbb{F}_q\mathbb{F}_q[u]$-additive code if C is an R-submodule of $\mathbb{F}_q^\alpha \times R^\beta$. In this paper, we obtain the generator matrix of these codes and the structure of their dual codes are given. we introduce Lee weight and homogenous weight over these codes. Also, we give some bounds on the minimum distance of these codes with respect to homogenous and Lee weights. At the end, we study one-weight codes and obtain $[q^2+q, 2, q^2]$ and $[2(q+1), 2, 2q]$ one-weight optimal codes over $\mathbb{F}_q$.

1. Introduction

A subgroup of $\mathbb{Z}_2^\alpha \times \mathbb{Z}_4^\beta$, where α and β are positive integers, is called a $\mathbb{Z}_2\mathbb{Z}_4$-additive code [5]. The studies on $\mathbb{Z}_2\mathbb{Z}_4$-additive codes and their algebraic structures have attracted many researchers; see [1, 3, 4, 5, 12]. In [12], Dougherty et al. described one weight $\mathbb{Z}_2\mathbb{Z}_4$-additive codes. They described the structure and possible weights for all one weight $\mathbb{Z}_2\mathbb{Z}_4$-additive codes.

Later, these codes were generalized to $\mathbb{Z}_2\mathbb{Z}_2[u]$-additive codes [2]. These codes have been studied with respect to Lee weight [6]. In particular, one-Lee weight $\mathbb{Z}_2\mathbb{Z}_2[u]$-additive codes have been studied in [19]. Also, cyclic $\mathbb{Z}_2\mathbb{Z}_2[u]$-additive codes have been studied [25].

Recently, $\mathbb{Z}_2\mathbb{Z}_2[u]$-additive codes were generalized to $\mathbb{Z}_p\mathbb{Z}_p[u]$-additive codes, where p is a prime number [21]. Also, these codes have been studied with respect to Lee weight. In [21], the linear and cyclic structures of these codes were given. Among other results, some optimal codes were obtained from a subclass of these codes [21].

In this paper, we give a comprehensive study on $\mathbb{F}_q\mathbb{F}_q[u]$-additive codes, where $q = p^m$ for some prime number p and a positive integer m. We study these codes with respect to Lee and homogenous weights. The structure of this paper is as follows.

In Section 2, we introduce some notations and basic facts which will be utilized later in our discussion. In Section 3, we introduce $\mathbb{F}_q\mathbb{F}_q[u]$-additive codes and obtain the generator matrix of these codes in a case that q is a power of a prime number. Moreover, the structure of dual codes are given. In Section 4, we generalize the Lee weight over $\mathbb{F}_q\mathbb{F}_q[u]$-additive codes. Also, we introduce another weight function, homogenous weight, over these codes. We obtain some bounds on minimum distance of these codes with respect to these weight functions. In Section 5, we study one-weight codes

with respect to Lee and homogenous weights. In this section, by the Gray image of these codes, we obtain $[q^2+q, 2, q^2]$ and $[2(q+1), 2, 2q]$ one-weight optimal codes over $\mathbb{F}_q$. In Section 6, we introduce constacyclic codes over $\mathbb{F}_q\mathbb{F}_q[u]$-additive codes and we obtain the structure of these codes.

2. Preliminaries

We begin with some definitions for codes over rings.

From now on, suppose that R is a finite commutative ring with identity. The Jacobson radical of R is denoted by $rad(R)$. For an R-module M, left or right, the socle of M, denoted by $soc(M)$, is the sum of all minimal nonzero submodules of M.

Definition 2.1. [17] Let R be a artinian ring. If $soc(R) \cong R/rad(R)$ as right R-modules and as left R-modules, then R is called a Frobenius ring.

Definition 2.2. [20] A ring R is called a finite chain ring, if its ideals are linearly ordered by inclusion.

Lemma 2.3. *If R be a chain ring, then R is a local ring.*

Proof. The proof is clear by the definition of chain ring. □

A code is a subset of R^n and a linear code over R is an R-submodule of R^n. In this case we say that the code has length n. For a code C, we define the rank of C, denoted by $rank(C)$, to be the minimum number of generators of C.

The Hamming distance between two vectors $u, v \in R^n$ is the number of coordinates in which u and v differ from one another and it is denoted by $d_H(u, v)$. The Hamming weight of a vector $u \in R^n$, denoted by $w_H(u)$, is the number of non-zero coordinates of u. The minimum Hamming distance of a code C is denoted by $d_H(C)$ and is given as: $d_H(C) = min\{d_H(u, v) : u, v \in C, u \neq v\}$. The minimum Hamming weight of a code C, denoted by $w_H(C)$, is the minimum value of $w_H(u)$ for $u \in C \setminus \{0\}$.

We define the inner product of vectors u and $v \in R^n$ as follows:

$$u.v = \sum_{i=1}^{n} u_i v_i.$$

Let C be a linear code over R. The dual code of C, denoted by $C^\perp$, is defined as follows:

$$C^\perp = \{v \in R^n | \ u.v = 0, \text{ for all } u \in C\}.$$

Lemma 2.4. [26, Theorem 3] *Let C be a code of length n over a finite Frobenius ring. Thus, $|C||C^\perp| = |R^n|$.*

The Gray maps, which are defined in each case, have been used as tools to linked codes over rings and codes over finite fields. Gray maps from $\mathbb{Z}_4^n$ to $\mathbb{Z}_2^{2n}$ were effectively used by Sloane, Calderbank, et al. in their work [14], as a tool to obtain the binary nonlinear Kerdock, Preparata, and Goethals codes as the Gray images of linear codes over $\mathbb{Z}_4$. Carlet, in [6], extended this map to $\mathbb{Z}_{2^k}$ with the homogeneous weight and used this to obtain the generalized Kerdock codes that were non-linear binary codes with large minimum distances. Several other authors, like Ling generalized the notion of Gray maps to more general rings with certain homogeneous weights defined on them in [18]. In this paper, we use the Gray map used in paper [16], which is defined for chain rings.

3. $\mathbb{F}_q\mathbb{F}_q[u]$-additive codes

Throughout this paper, $\mathbb{F}_q$ denotes a finite field with q elements, where $q = p^m$ is a power of a prime number p. The ring $\mathbb{F}_q + u\mathbb{F}_q$ consists of all polynomials of degrees 0 and 1 in an indeterminate u over $\mathbb{F}_q$, and it is closed under polynomial addition and multiplication modulo u^2. Thus $\mathbb{F}_q + u\mathbb{F}_q = \frac{\mathbb{F}_q[u]}{\langle u^2 \rangle} = \{\delta + \theta u : \ \delta, \theta \in \mathbb{F}_q\}$. It is easy to see that $\mathbb{F}_q + u\mathbb{F}_q$ is a chain ring with the maximal ideal $\mathfrak{m} = \langle u \rangle$.

Also, we have the following ring homomorphism:

$$\tau : \mathbb{F}_q + u\mathbb{F}_q \longrightarrow \mathbb{F}_q, \ \delta + \theta u \longmapsto \delta.$$

Hence $\mathbb{F}_q$ is an $(\mathbb{F}_q + u\mathbb{F}_q)$-module, where for $\delta + \theta u \in \mathbb{F}_q + u\mathbb{F}_q$ and $c \in \mathbb{F}_q$, the scalar multiplication is defined as $(\delta + \theta u).c = \delta c \in \mathbb{F}_q$. From now on, we denote $\mathbb{F}_q + u\mathbb{F}_q$ by R.

In this section, we introduce and study basic facts of $\mathbb{F}_q\mathbb{F}_q[u]$-additive codes. In particular, the structure of these codes and their dual codes are given.

Definition 3.1. Let α and β be two positive integers. A nonempty subset C of $\mathbb{F}_q^\alpha \times R^\beta$ is called an $\mathbb{F}_q\mathbb{F}_q[u]$-additive code if C is an R-submodule of $\mathbb{F}_q^\alpha \times R^\beta$ with the following scalar multiplication, where for $r = \delta + \theta u \in R$ and $(a_\alpha, b_\beta) = (a_0, a_1, ..., a_{\alpha-1}, b_0, b_1, ..., b_{\beta-1}) \in C$,

$$r.(a_\alpha, b_\beta) = (\delta a_\alpha, r b_\beta) = (\delta a_0, \delta a_1, ..., \delta a_{\alpha-1}, r b_0, r b_1, ..., r b_{\beta-1}).$$

By the same argument as [21, Theorem 1], the following theorem gives the generator matrix of $\mathbb{F}_q\mathbb{F}_q[u]$-additive codes.

Theorem 3.2. *Let $C \subseteq \mathbb{F}_q^\alpha \times R^\beta$ be an $\mathbb{F}_q\mathbb{F}_q[u]$-additive code. Then C is permutation equivalent to an $\mathbb{F}_q\mathbb{F}_q[u]$-additive code with the standard form matrix*

$$G = \left(\begin{array}{ccc|ccc} I_{k_0} & A_1 & & 0 & 0 & uB \\ 0 & A_2 & & I_{k_1} & D_1 & D_2 + uD_3 \\ 0 & 0 & & 0 & uI_{k_2} & uD_4 \end{array}\right), \quad (3.1)$$

where I_{k_0}, I_{k_1} and I_{k_2} denote the $k_0 \times k_0$, $k_1 \times k_1$ and $k_2 \times k_2$ identity matrices. respectively, A_1, A_2, B, D_1, D_2, D_3 and D_4 are matrices over $\mathbb{F}_q$. Moreover, $|C| = q^{k_0+2k_1+k_2}$.

An $\mathbb{F}_q\mathbb{F}_q[u]$-additive code $C \subseteq \mathbb{F}_q^\alpha \times R^\beta$ with the generator matrix given in Equation 3.1 is said to be of type $(\alpha, \beta, k_0, k_1, k_2)$, where $k = k_0 + k_1 + k_2$ is called the rank of C and denoted by $\mathrm{rank}(C)$.

The inner product for two elements $x, y \in \mathbb{F}_q^\alpha \times R^\beta$ is defined as follows:

$$x.y := u(\sum_{i=0}^{\alpha-1} x_i y_i) + \sum_{i=\alpha}^{\alpha+\beta-1} x_i y_i.$$

For an $\mathbb{F}_q\mathbb{F}_q[u]$-additive code C, $C^\perp$ is the dual of C with respect to the above inner product.

Proposition 3.3. *Let $C \subseteq \mathbb{F}_q^\alpha \times R^\beta$ be an $\mathbb{F}_q\mathbb{F}_q[u]$-additive code. Then*

(1) $|C||C^\perp| = |\mathbb{F}_q^\alpha \times R^\beta|$.

(2) $(C^\perp)^\perp = C$.

Proof. (1) Define $\varphi : \mathbb{F}_q^\alpha \times R^\beta \longrightarrow R^\alpha \times R^\beta$ by $(x, y) \longmapsto (xu, y)$. It is easy to see that φ is an injective R-module homomorphism. Hence, $\varphi(C)$ is a linear code over R of length $\alpha + \beta$, where $|\varphi(C)| = |C|$. Now, let $\varphi(C)^\perp$ be the dual of $\varphi(C)$ with respect to the standard inner product over R. Then, $\varphi(C)^\perp = \{(a+bu, c+du) : (a, c + du) \in C^\perp\}$. Since $b \in \mathbb{F}_q^\alpha$, so $|\varphi(C)^\perp| = |C^\perp||\mathbb{F}_q|^\alpha = |C^\perp|q^\alpha$. Since R is a Frobenius ring, by Lemma 2.4, we have $|\varphi(C)^\perp||\varphi(C)| = |R|^{\alpha+\beta} = q^{2\alpha+2\beta}$. Thus, $|C^\perp|q^\alpha = \frac{q^{2\alpha+2\beta}}{|\varphi(C)|} = \frac{q^{2\alpha+2\beta}}{|C|}$. This shows that $|C^\perp||C| = \frac{q^{2\alpha+2\beta}}{q^\alpha} = q^{\alpha+2\beta} = |\mathbb{F}_q^\alpha \times R^\beta|$.

(2) Clearly $C \subseteq (C^\perp)^\perp$. Now, by part (1), $|C| = \frac{|\mathbb{F}_q^\alpha \times R^\beta|}{|C^\perp|}$ and $|C^\perp| = \frac{|\mathbb{F}_q^\alpha \times R^\beta|}{|(C^\perp)^\perp|}$. Thus $|C| = |(C^\perp)^\perp|$, which completes the proof. □

The following theorem gives the generator matrix of $C^\perp$.

Theorem 3.4. *Let C be an $\mathbb{F}_q\mathbb{F}_q[u]$-additive code of type $(\alpha, \beta, k_0, k_1, k_2)$ with the standard form matrix defined in Equation 3.1. Then the generator matrix for $C^\perp$ is given by*

$$H = \left(\begin{array}{ccc|c} -A_1^t & I_{\alpha-k_0} & & -uA_2^t \\ -B^t & 0 & & -(D_2 + uD_3)^t + D_4^t D_1^t \\ 0 & 0 & & -uD_1^t \\ \hline & & 0 & 0 \\ & & -D_4^t & I_{\beta-k_1-k_2} \\ & & uI_{k_2} & 0 \end{array}\right).$$

Proof. Let $\widetilde{C}$ be the $\mathbb{F}_q\mathbb{F}_q[u]$-additive code, which is generated by H. If G be the generator matrix of C, it is easy to see that $HG^t = 0$. Hence, $\widetilde{C} \subseteq C^\perp$. From the generator matrices of C and $\widetilde{C}$, we have that $|C| = q^{k_0+2k_1+k_2}$ and $|\widetilde{C}| = q^{\alpha-k_0+2(\beta-k_1-k_2)+k_2} = q^{\alpha+2\beta-k_0-2k_1-k_2}$. By the proposition 3.3, $|C^\perp| = \frac{|\mathbb{F}_q^\alpha \times R^\beta|}{|C|} = \frac{q^{\alpha+2\beta}}{q^{k_0+2k_1+k_2}} = |\widetilde{C}|$. Hence, $\widetilde{C} = C^\perp$ and the proof is completed. □

4. Weight functions over $\mathbb{F}_q\mathbb{F}_q[u]$-additive codes

In this section, we introduce two weight functions, homogenous weight and Lee weight, over $\mathbb{F}_q\mathbb{F}_q[u]$-additive codes. First, note that R is a chain ring with the maximal ideal $\mathfrak{m} = \langle u \rangle$, nilpotency index 2, and residue field $R/\mathfrak{m} = \mathbb{F}_q$. A homogenous weight over R is defined as follows:

$$\omega_{hom}(t) = \begin{cases} q-1, & t \in R \setminus \mathfrak{m}; \\ q, & t \in \mathfrak{m} \setminus \{0\}; \\ 0, & t = 0. \end{cases}$$

Now, the weight function ω over $\mathbb{F}_q^\alpha \times R^\beta$ is defined as $\omega(x, y) = \omega_H(x) + \omega_{hom}(y)$, where $(x, y) \in \mathbb{F}_q^\alpha \times R^\beta$ and ω_H denotes the Hamming weight. Also, the distance between any two codewords is the weight of their difference; for $(x, y), (x', y') \in \mathbb{F}_q^\alpha \times R^\beta$, $d_\omega((x, y), (x', y')) = \omega(x - x', y - y')$. In partiqular, for an $\mathbb{F}_q\mathbb{F}_q[u]$-additive code $C \subseteq \mathbb{F}_q^\alpha \times R^\beta$, the nonzero minimum distance between the codewords in C is denoted by $d_{hom}(C)$. By [16, Proposition 3.1], there exists a Gray map from (R^β, d_{hom}) to $(\mathbb{F}_q^{q\beta}, d_H)$, where d_H denotes the Hamming distance on $\mathbb{F}_q^{q\beta}$. Let φ_{hom} :

$R^\beta \longrightarrow \mathbb{F}_q^{q\beta}$ be the defined Gray map in [16]. Then the map $(id, \varphi_{hom}) : \mathbb{F}_q^\alpha \times R^\alpha \longrightarrow \mathbb{F}_q^{\alpha+q\beta}$ is an isometry which transforms the homogenous distance in $\mathbb{F}_q^\alpha \times R^\beta$ to the Hamming distance in $\mathbb{F}_q^{\alpha+q\beta}$. In this paper, we denote (id, φ_{hom}) by Φ_{hom}.

Example 4.1. Let C be an $\mathbb{F}_5\mathbb{F}_5[u]$-additive code of type $(6, 31, 1, 1, 0)$ with the following standard form matrix

$$G = \left(\begin{array}{cccccc|cccccc} 1 & 0 & 2 & 3 & 4 & 1 & 0 & \mathbf{0} & u & \mathbf{B} & \mathbf{B} & \mathbf{B} \\ 0 & 1 & 4 & 3 & 2 & 4 & 1 & \mathbf{D} & 0 & \mathbf{1} & \mathbf{2} & \mathbf{3} \end{array} \begin{array}{ccc} \mathbf{B} & \mathbf{B} & \mathbf{B} \\ \mathbf{4} & \mathbf{u} & \mathbf{1+u} \end{array}\right),$$

where $\mathbf{B} = (u\ 2u\ 3u\ 4u)$ and $\mathbf{D} = (2\ 3\ 4\ u\ 1{+}u)$. By Theorem 3.2, $|C| = 5^{k_0+2k_1+k_2} = 5^3$. Also, we can see that $\Phi_{hom}(C)$ is a $[161, 3, 125]$ linear code over $\mathbb{F}_5$ with respect to Hamming weight.

Now, we introduce another weight function over these codes. In [21], the Lee weight over $\mathbb{F}_q\mathbb{F}_q[u]$-additive codes, in the case that q is a prime number, is defined. We give this weight function on $\mathbb{F}_q\mathbb{F}_q[u]$-additive codes in general.

The Gray map defined on R can be expressed as follows:

$$\phi' : R \longrightarrow \mathbb{F}_q^2,\ \phi'(a + bu) = (b, a + b).$$

The Lee weight is $\omega_L(a + bu) = \omega_H(b, a + b)$. The map ϕ' and the Lee weight ω_L generalize over R^β naturally; for all $c := (a_1 + b_1u, \cdots, a_\beta + b_\beta u) \in R^\beta$, we have $\phi'(c) = (b_1, \cdots, b_\beta, a_1 + b_1, \cdots, a_\beta + b_\beta)$ and $\omega_L(c) = \omega_H(\phi(c))$.

A weight function ω' over $\mathbb{F}_q^\alpha \times R^\beta$ is defined as $\omega'(x, y) = \omega_H(x) + \omega_L(y)$, where $(x, y) \in \mathbb{F}_q^\alpha \times R^\beta$. Also, the distance between any two codewords is defined by the argument as homogenous weight. Now, the following map is an isometry which transforms the Lee distance in $\mathbb{F}_q^\alpha \times R^\beta$ to the Hamming distance in $\mathbb{F}_q^{\alpha+2\beta}$.

$$\Phi_{Lee} : \mathbb{F}_q^\alpha \times R^\beta \longrightarrow \mathbb{F}_q^{\alpha+2\beta},\ (x, y) \longmapsto (x, \phi'(y)).$$

From now on, we call this weight function, Lee weight [21]. In addition, If $C \subseteq \mathbb{F}_q^\alpha \times R^\beta$ is an $\mathbb{F}_q\mathbb{F}_q[u]$-additive code, we denote the non-zero minimum distance between the codewords in C by $d_L(C)$.

Example 4.2. Let C be an $\mathbb{F}_3\mathbb{F}_3[u]$-additive code of type $(4, 2, 1, 1, 0)$ with the following standard form matrix

$$G = \left(\begin{array}{cccc|cc} 1 & 0 & 1 & 1 & 0 & u \\ 0 & 1 & 1 & 2 & 1 & 0 \end{array}\right).$$

By Theorem 3.2, $|C| = 3^{k_0+2k_1+k_2} = 3^3$. Also, one can see that $\Phi_{Lee}(C)$ is a $[8, 3, 2]$ linear code over $\mathbb{F}_3$ with respect to Hamming weight.

Lemma 4.3. [24, Theorem 1] *Let C be a q-ary code of parameters (n, M, d_H), where M is the size of C. Then, the Singleton bound is as follows:*

$$d_H(C) \leq n - \log_q |M| + 1.$$

The following theorem gives some bounds on $\mathbb{F}_q\mathbb{F}_q[u]$-additive codes with respect to homogenous and Lee weights.

Theorem 4.4. *Let C be an $\mathbb{F}_q\mathbb{F}_q[u]$-additive code of type $(\alpha, \beta, k_0, k_1, k_2)$. Then*

(1) $\frac{d_{hom}(C)-1}{q} \leq \frac{\alpha}{q} + \beta - \frac{k_0+2k_1+k_2}{q}$;

(2) $\lfloor \frac{d_{hom}(C)-1}{q} \rfloor \leq \alpha + \beta - (k_0 + k_1 + k_2)$;

(3) $\frac{d_L(C)-1}{2} \leq \frac{\alpha}{2} + \beta - \frac{k_0+2k_1+k_2}{2}$;

(4) $\lfloor \frac{d_L(C)-1}{2} \rfloor \leq \alpha + \beta - (k_0 + k_1 + k_2)$,

where $\lfloor x \rfloor$ refers to the greatest integer less than or equal to x.

Proof. (1) Consider the Gray map $\Phi_{hom} : \mathbb{F}_q^\alpha \times R^\beta \longrightarrow \mathbb{F}_q^{\alpha+q\beta}$. Since $\Phi_{hom}(C) \subseteq \mathbb{F}_q^{\alpha+q\beta}$, thus $\Phi_{hom}(C)$ is a code over $\mathbb{F}_q$ of length $\alpha + q\beta$. Now by the Singleton bound, we have $d_H(\Phi_{hom}(C)) \leq \alpha + q\beta - \log_q |\Phi_{hom}(C)| + 1$. But $d_H(\Phi_{hom}(C)) = d_{hom}(C)$ and $|\Phi_{hom}(C)| = |C| = q^{k_0+2k_1+k_2}$. This completes the proof.

(2) Define the map $\rho : \mathbb{F}_q \longrightarrow R$ by $a \longmapsto au$. It is easy to see that ρ is well defined and injective. Now for $b + cu \in R$, $\rho((b + cu).a) = \rho(ba) = bau = (b + cu).au = (b + cu).\rho(a)$. Hence ρ is an R-module homomorphism. Also, denote the natural generalization of ρ on $\mathbb{F}_q^\alpha$ by ρ. Thus the map $(\rho, id) : \mathbb{F}_q^\alpha \times R^\beta \longrightarrow R^{\alpha+\beta}$ is an injective R-module homomorphism. Now $\omega_H(a) = 1 \leq q = \omega_{hom}(au)$ for all $a \in \mathbb{F}_q$. Therefore $(\rho, id)(C)$ is a linear code with $d_{hom}(C) \leq d_{hom}((\rho, id)(C))$. If A is the maximum weight of elements in R, then by [22, Theorem 3.7], $\lfloor \frac{d_{hom}(C)-1}{A} \rfloor \leq \lfloor \frac{d_{hom}((\rho,id)(C))-1}{A} \rfloor \leq \alpha + \beta - \text{rank}((\rho, id)(C)) = \alpha + \beta - \text{rank}(C) = \alpha + \beta - (k_0 + k_1 + k_2)$. But $A = q$ which completes the proof.

(3) Consider the Gray map $\Phi_{Lee} : \mathbb{F}_q^\alpha \times R^\alpha \longrightarrow \mathbb{F}_q^{\alpha+2\beta}$ instead of the Gray map Φ_{hom} in part (1). By the same argument as part (1), the result is followed.

(4) Consider the map $(\rho, id) : \mathbb{F}_q^\alpha \times R^\beta \longrightarrow R^{\alpha+\beta}$ in part (2). Then $\omega_H(a) = 1 \leq 2 = \omega_H(a, a) = \omega_L(au)$

for all $a \in \mathbb{F}_q$. Also, the maximum Lee weight of elements in R is equal to $A = 2$. Now, by the same argument as part (2), the result is obtained. □

5. One-weight $\mathbb{F}_q\mathbb{F}_q[u]$-additive codes

An $\mathbb{F}_q\mathbb{F}_q[u]$-additive code is said to have one-Lee (or homogenous) weight if every non-zero codeword has the same Lee (or homogenous) weight. In this section, we study the properties of one-Lee weight and one-homogenous weight $\mathbb{F}_q\mathbb{F}_q[u]$-additive codes. We give the exact structure of some large classes of one-weight codes. In particular, we introduce two classes of one-weight codes that their Gray images are $[q^2+q, 2, q^2]$ and $[2(q+1), 2, 2q]$ one-weight optimal codes over $\mathbb{F}_q$.

One-homogenous weight codes

Now, we study one-homogenous weight $\mathbb{F}_q\mathbb{F}_q[u]$-additive codes.

Lemma 5.1. *Let $C \subseteq \mathbb{F}_q^{\alpha} \times R^{\beta}$ be an $\mathbb{F}_q\mathbb{F}_q[u]$-additive code. Then*

$$\sum_{c\in C} \omega(c) = \frac{|C|(q-1)}{q}(\alpha + q\beta).$$

Proof. We write the codewords of C as rows of a matrix G. Consider the column j of G, where $\alpha + 1 \leq j \leq \alpha + \beta$. Let J be the ideal of R generated by all elements of the column j. Since R is a chain ring and $\mathfrak{m}$ is the maximal ideal of R, so $J = \mathfrak{m}$ or $J = R$. Now, we consider the following two cases:

Case 1 ($J = \mathfrak{m} = \langle u \rangle$). Since C is an R-submodule, any element of J is an element of the column j. Now we show that any two elements of J have the same repetition number in the column j. Let au and bu be two elements of J with the repetition numbers n_a and n_b, respectively. Then $au = ab^{-1}(bu)$ and hence $n_a \geq n_b$. By the same argument, $n_b \geq n_a$. So $n_a = n_b$. Therefore, the sum of the weights of all elements of the column j is equal to

$$\frac{|C|}{|J|}(\sum_{s\in J}\omega_{hom}(s)) = \frac{|C|}{|J|}(q|J \setminus \{0\}|) =$$

$$\frac{|C|}{q}(q(q-1)) = |C|(q-1).$$

Case 2 ($J = R$). In this case, there exists an invertible element c_j in the column j. If c'_j is an element of J, then $c'_j = (c'_j c_j^{-1})c_j$. Thus any element of $J = R$ is an element of the column j. Now, let $c_1 = a_1 + b_1 u$ and $c_2 = a_2 + b_2 u$ be two elements of J. We show that c_1 and c_2 have the same repetition number in the column j. If a_1 and a_2 are non-zero elements, then c_1 and c_2 are invertible. Hence $c_1 = (c_1 c_2^{-1})c_2$ and $c_2 = (c_2 c_1^{-1})c_1$. This shows that c_1 and c_2 have the same repetition number. If $a_1 = a_2 = 0$, then by the same argument as case 1, $n_{c_1} = n_{c_2}$. Now, let a_1 be a non-zero element and $a_2 = 0$. Then c_1 is invertible. We have that $c_2 = (c_2 c_1^{-1})c_1$ which proves $n_{c_2} \geq n_{c_1}$. Also, $c_1 = (a_1 c_j^{-1})c_j + (b_1 b_2^{-1})c_2$ which shows that $n_{c_1} \geq n_{c_2}$. Thus, the elements of J have the same repetition number $\frac{|C|}{|J|} = \frac{|C|}{q^2}$.

Therefore, the sum of the weights of all elements of the column j is equal to

$$\frac{|C|}{q^2}(\sum_{s\in R\setminus\mathfrak{m}}\omega_{hom}(s) + \sum_{s\in\mathfrak{m}\setminus\{0\}}\omega_{hom}(s)) =$$

$$\frac{|C|}{q^2}((q^2-q)(q-1) + (q-1)q) = |C|(q-1).$$

If $1 \leq j \leq \alpha$, then the ideal J of R generated by all elements of the column j is equal to $\mathbb{F}_q$. Since all elements of $\mathbb{F}_q$ are invertible, they have the same repetition number. Hence, the sum of the weights of all elements of the column j is equal to

$$\frac{|C|}{|\mathbb{F}_q|}(\sum_{s\in\mathbb{F}_q}\omega_H(s)) = \frac{|C|}{q}(q-1).$$

Therefore,

$$\sum_{c\in C} \omega(c) = \frac{|C|(q-1)}{q}(\alpha + q\beta).$$

□

Theorem 5.2. *Let $C \subseteq \mathbb{F}_q^{\alpha} \times R^{\beta}$ be a one-homogenous weight $\mathbb{F}_q\mathbb{F}_q[u]$-additive code with weight m. Then there exists a unique positive integer λ such that $m = \lambda\frac{|C|}{q}$ and $\alpha + q\beta = \lambda(\frac{|C|-1}{q-1})$.*

Proof. By the lemma 5.1, we have

$$\sum_{c\in C}\omega(c) = \frac{|C|(q-1)}{q}(\alpha + q\beta).$$

On the other hand, the sum of the weights of all codewords is $(|C|-1)m$. Hence, $\frac{|C|(q-1)}{q}(\alpha+q\beta) = (|C|-1)m$. Since $|C| = q^{k_0+2k_1+k_2}$, $\gcd(\frac{|C|}{q}, (|C|-1)) = 1$. Therefore, there exists a positive integer λ such that $m = \lambda\frac{|C|}{q}$, and hence $(q-1)(\alpha+q\beta) = \lambda(|C|-1)$. □

The following theorem, determines a class of one-homogenous weight codes.

Theorem 5.3. *Let $C \subseteq \mathbb{F}_q^{\alpha} \times R^{\beta}$ be a one-homogenous weight $\mathbb{F}_q\mathbb{F}_q[u]$-additive code with weight m. If m is odd, and $q = 2^s$ for some positive integer s, then $C = \{(\underbrace{\theta, ..., \theta}_{\alpha}, \underbrace{\theta u, ..., \theta u}_{\beta}) : \theta \in \mathbb{F}_q\}$.*

Proof. Clearly, $C = \{(\theta, ..., \theta, \theta u, ..., \theta u) : \theta \in \mathbb{F}_q\}$ is a one-homogenous weight code of weight $\alpha + q\beta$. Now, we show that any one-homogenous weight code has this form. By the Theorem 5.2, $m = \lambda\frac{|C|}{q}$. Since m is odd and $q = 2^s$, λ should be an odd integer and $\frac{|C|}{q} = 1$. Hence, $m = \lambda = \alpha + q\beta$. Let $(a, b) = (a_1, ..., a_\alpha, b_1, ..., b_\beta)$ be a codeword in C. If $a_i = 0$ for some i, or $b_j \notin \mathfrak{m}$, then $\omega(a, b) < \alpha + q\beta$; a contradiction. Hence, a_i is a non-zero element of $\mathbb{F}_q$ for all $1 \leq i \leq \alpha$, and $b_j = \theta_j u$ where θ_j is a non-zero element of $\mathbb{F}_q$ for $1 \leq j \leq \beta$. Clearly, $(\delta a, \delta b) \in C$ for any $\delta \in \mathbb{F}_q$. Thus, if $a_i \neq \theta_j$ for some $i \neq j$, $|C| > q$; is a contradiction. Hence, $C = \{(\underbrace{\theta, ..., \theta}_{\alpha}, \underbrace{\theta u, ..., \theta u}_{\beta}) : \theta \in \mathbb{F}_q\}$. □

Now, we introduce a class of optimal one-homogenous weight codes. The following lemma gives the well-known Griesmer bound for linear codes.

Lemma 5.4. [15, Theorem 2.7.4] *Let C be a q-ary code of parameters $[n, k, d_H]$, where $k \geq 1$. Then the Griesmer bound is as follows:*

$$n \geq \sum_{i=0}^{k-1} \lceil \frac{d_H}{q^i} \rceil.$$

Definition 5.5. If a linear code C over a finite field $\mathbb{F}_q$ meets the Griesmer bound, then C is called optimal.

Suppose that $\mathbb{F}_q = \{0, f_1 = 1, f_2, ..., f_{q-1}\}$. The following theorem gives a class of optimal codes.

Theorem 5.6. *Let C be an $\mathbb{F}_q\mathbb{F}_q[u]$-additive code of type $(q, q, 1, 0, 1)$ with the following standard form matrix:*

$$G = \left(\begin{array}{c|cc} \mathbf{a} & 0 & \mathbf{b} \\ \mathbf{0} & u & \mathbf{d} \end{array}\right),$$

where $\mathbf{a} = (\underbrace{1, 1, \cdots, 1}_{q \text{ times}})$, $\mathbf{b} = (\underbrace{u, u, \cdots, u}_{(q-1) \text{ times}})$ and $\mathbf{d} = (f_1 u, f_2 u, \cdots, f_{(q-1)} u)$. Then C is a one-homogenous weight code with weight $m = q^2$. Also, $\Phi_{hom}(C)$ is an optimal one-Hamming weight code with parameters $[q^2 + q, 2, q^2]$.

Proof. Let c be a codeword in C. Then by the definition of $\mathbb{F}_q\mathbb{F}_q[u]$-additive codes and the structure of G, $c = (\delta\mathbf{a} \;\; 0 \;\; \delta\mathbf{b}) + (\mathbf{0} \;\; \delta' u \;\; \delta'\mathbf{d}) = (\delta\mathbf{a} \;\; \delta' u \;\; \delta\mathbf{b} + \delta'\mathbf{d})$ for some $\delta, \delta' \in \mathbb{F}_q$. But the weight of c is equal to

$$\omega(c) = \omega_h(\delta\mathbf{a}) + \omega_{hom}(\delta' u) + \omega_{hom}(\delta\mathbf{b} + \delta'\mathbf{d}) =$$
$$q + q + \omega_{hom}(\delta\mathbf{b} + \delta'\mathbf{d}).$$

Since $\mathbb{F}_q$ is a field, there exists only one element $\theta \in \mathbb{F}_q$ such that $\delta + \theta = 0$. But $\mathbb{F}_q(\delta') = \mathbb{F}_q$. Hence, there exists only one integer $1 \leq i \leq q-1$ such that $\delta + f_i\delta' = 0$. This shows that $\omega_{hom}(\delta\mathbf{b} + \delta'\mathbf{d}) = (q-2)q$. Therefore, $\omega(c) = q^2$. It is easy to see that $\Phi_{hom}(C)$ is a one-Hamming weight code with parameters $[q^2 + q, 2, q^2]$. Now, $q^2 + q = \lceil \frac{q^2}{1} \rceil + \lceil \frac{q^2}{q} \rceil$. □

Example 5.7. Let C be an $\mathbb{F}_5\mathbb{F}_5[u]$-additive code of type $(5, 5, 1, 0, 1)$ with the following standard form matrix:

$$G = \left(\begin{array}{ccccc|ccccc} 1 & 1 & 1 & 1 & 1 & 0 & u & u & u & u \\ 0 & 0 & 0 & 0 & 0 & u & u & 2u & 3u & 4u \end{array}\right).$$

Then C is a one-homogenous weight code with weight $m = 25$. Also, $|C| = 5^{k_0+2k_1+k_2} = 5^2$, and hence $\lambda = 5$. Moreover, $\Phi_{hom}(C)$ is a one-Hamming weight $[30, 2, 25]$ code over $\mathbb{F}_5$ which is an optimal code.

One-Lee weight codes

Now, we study one-Lee weight $\mathbb{F}_q\mathbb{F}_q[u]$-additive codes.

Lemma 5.8. *Let $C \subseteq \mathbb{F}_q^{\alpha} \times R^{\beta}$ be an $\mathbb{F}_q\mathbb{F}_q[u]$-additive code. Then*

$$\sum_{c \in C} \omega'(c) = \frac{|C|(q-1)}{q}(\alpha + 2\beta).$$

Proof. By the same argument as Lemma 5.1, write the codewords of C as rows of a matrix G. Consider the column j of G, where $\alpha + 1 \leq j \leq \alpha + \beta$. Let J be the ideal of R generated by all elements of the column j. Then $J = \mathfrak{m}$ or $J = R$. Hence, we have the following two cases: Case 1 ($J = \mathfrak{m} = \langle u \rangle$). Clearly, any element of the column j is of the form bu for some $b \in \mathbb{F}_q \setminus \{0\}$. By case 1 of Lemma 5.1, any element of J is an element of the column j. Also, any two elements of J have the same repetition number in the column j. Hence, the sum of the weights of all elements of the column j is equal to

$$\frac{|C|}{|J|}(\sum_{s \in J} \omega_L(s)) = \frac{|C|}{|J|}(2|J \setminus \{0\}|) =$$
$$\frac{|C|}{q}(2(q-1)) = \frac{2|C|}{q}(q-1).$$

Case 2 ($J = R$). By case 2 of Lemma 5.1, any element of $J = R$ is an element of the column j and all elements have the same repetition number. Now, let $a + bu$ be an element of the column j. Then, we have the following cases:

(1) $a = 0, b \neq 0$;
(2) $a \neq 0, b = 0$;
(3) $a, b \neq 0$ and $a = -b$;
(4) $a, b \neq 0$ and $a \neq -b$.

Hence, the sum of the weights of all elements of the column j is equal to

$$\frac{|C|}{q^2}(2(q-1)+q-1+q-1+2(q-1)(q-2)) =$$

$$\frac{|C|(q-1)}{q^2}(4+2(q-2)) = \frac{2|C|}{q}(q-1).$$

If $1 \leq j \leq \alpha$, then by the same argument as Lemma 5.1, the sum of the weights of all elements of the column j is equal to

$$\frac{|C|}{|\mathbb{F}_q|}(\sum_{s\in\mathbb{F}_q}\omega_H(s)) = \frac{|C|}{q}(q-1).$$

Therefore,

$$\sum_{c\in C}\omega'(c) = \frac{|C|(q-1)}{q}(\alpha+2\beta).$$

□

Theorem 5.9. *Let $C \subseteq \mathbb{F}_q^\alpha \times R^\beta$ be a one-Lee weight $\mathbb{F}_q\mathbb{F}_q[u]$-additive code with weight m. Then, there exists a unique positive integer λ such that $m = \lambda\frac{|C|}{q}$ and $\alpha + 2\beta = \lambda(\frac{|C|-1}{q-1})$.*

Proof. It is proved by the same argument as Theorem 5.2. □

Now, we determine the structure of one-Lee weight codes in the case that m is odd.

Theorem 5.10. *Let C be a one-Lee weight $\mathbb{F}_q\mathbb{F}_q[u]$-additive code of type $(\alpha, \beta, k_0, k_1, k_2)$ with weight m, where m is odd. Then, $k_1 = 0$ and according to Theorem 3.2, the generator matrix of C is given by the following*

$$G = \left(\begin{array}{cc|ccc} I_{k_0} & A_1 & 0 & 0 & uB \\ 0 & 0 & 0 & uI_{k_2} & uD_4 \end{array}\right).$$

Proof. Let $k_1 \neq 0$ and c be a vector in $(0\ A_2\ I_{k_1}\ D_1\ D_2 + uD_3)$. since C is an R-submodule of $\mathbb{F}_q^\alpha \times R^\beta$, by the Definition 3.1 we have:

$$u.c \in u(0\ A_2\ I_{k_1}\ D_1\ D_2 + uD_3) =$$

$$(0\ 0\ uI_{k_1}\ uD_1\ uD_2).$$

Thus, C contains a non-zero vector of even weight and it is a contradiction. □

The following theorem gives the exact structure of one weight codes with odd distances if q is an even integer.

Theorem 5.11. *Let $C \subseteq \mathbb{F}_q^\alpha \times R^\beta$ be a one-Lee weight $\mathbb{F}_q\mathbb{F}_q[u]$-additive code with weight m. If m is odd, and $q = 2^s$ for some positive integer s, then $C = \{(\underbrace{\theta, ..., \theta}_{\alpha}, \underbrace{\theta u, ..., \theta u}_{\beta}) : \theta \in \mathbb{F}_q\}$.*

Proof. Clearly, $C = \{(\theta, ..., \theta, \theta u, ..., \theta u) : \theta \in \mathbb{F}_q\}$ is a one-Lee weight code of weight $\alpha + 2\beta$. Now, we show that any one-Lee weight code has this form. By the above theorem, $m = \lambda\frac{|C|}{q}$. Since m is odd and $q = 2^s$, λ should be an odd integer and $\frac{|C|}{q} = 1$. Hence, $m = \lambda = \alpha + 2\beta$. Now, let $(\theta, \delta + \theta u) = (\varepsilon_1, ..., \varepsilon_\alpha, \delta_1 + \theta_1 u, ..., \delta_\beta + \theta_\beta u)$ be a codeword in C. Since $m = \alpha + 2\beta$, $\varepsilon_i \neq 0$ for all i and we have the following two cases:

(1) $\delta_i = 0, \theta_i \neq 0$;
(2) $\delta_i, \theta_i \neq 0$ and $\delta_i \neq -\theta_i$.

But $|C| = q$. Hence, $\delta_i = 0$ for all i, and $\varepsilon_i = \theta_j$ for all $1 \leq i \leq \alpha$ and $1 \leq j \leq \beta$. This completes the proof. □

Now, the following theorem gives a class of one-Lee weight optimal codes.

Theorem 5.12. *Let C be an $\mathbb{F}_q\mathbb{F}_q[u]$-additive code of type $(2, q, 1, 0, 1)$ with the following standard form matrix*

$$G = \left(\begin{array}{c|cc} \mathbf{a} & 0 & \mathbf{b} \\ \mathbf{0} & u & \mathbf{d} \end{array}\right),$$

where $\mathbf{a} = (1, 1)$, $\mathbf{b} = (\underbrace{u, u, \cdots, u}_{(q-1)\ times})$ and $\mathbf{d} = (f_1 u, f_2 u, \cdots, f_{(q-1)} u)$. Then C is a one-Lee weight code with weight $m = 2q$. Also, $\Phi_{Lee}(C)$ is an optimal one-Hamming weight code with parameters $[2(q+1), 2, 2q]$.

Proof. It is proved by the same argument as Theorem 5.6. □

Example 5.13. Let C be an $\mathbb{F}_3\mathbb{F}_3[u]$-additive code of type $(2,3,1,0,1)$ with the following standard form matrix

$$G = \left(\begin{array}{cc|ccc} 1 & 1 & 0 & u & u \\ 0 & 0 & u & 2u & u \end{array}\right).$$

Then C is a one-Lee weight code with weight $m = 6$. Also, $|C| = 3^{k_0+2k_1+k_2} = 3^2$, and hence $\lambda = 2$. Moreover, $\Phi_{Lee}(C)$ is a one-Hamming weight $[8,2,6]$ linear code over $\mathbb{F}_3$ which is an optimal code.

6. Constacyclic $\mathbb{F}_q\mathbb{F}_q[u]$-additive codes

Recently, Qian et al.[21] have studied the cyclic codes over $\mathbb{Z}_p\mathbb{Z}_p[u]$-additive codes. In this section, we introduce constacyclic $\mathbb{F}_q\mathbb{F}_q[u]$-additive codes and obtain the structure of these codes.

Constacyclic codes over R

To obtain the structure of constacyclic additive codes over $\mathbb{F}_q \times R$, we need the structure of linear constacyclic codes over $\mathbb{F}_q$ and R. There are many researches about linear constacyclic codes over $\mathbb{F}_q$ and the structure of these codes for arbitrary length has been given; see [8, 23]. In this section, we remind some results about linear constacyclic codes over R.

Since R is a chain ring, all elements of $R \setminus \mathfrak{m}$ are unit. Hence, R has precisely $q(q-1)$ units, which are of the form $\delta + \theta u$, where $\delta \in \mathbb{F}_q \setminus \{0\}$ and $\theta \in \mathbb{F}_q$. Let $\lambda = \delta + \theta u$ be a unit of R. Consider the following correspondence:

$$\begin{aligned} p_\lambda : R^n &\longrightarrow R[x]/\langle x^n - \lambda\rangle, \\ (a_0, a_1, ..., a_{n-1}) &\longmapsto a_0 + a_1 x + ... + \\ a_{n-1}x^{n-1} + \langle x^n - \lambda\rangle. & \end{aligned}$$

Clearly p_λ is an R-module isomorphism. Also, it is easy to see that C is a λ-constacyclic code if and only if $p_\lambda(C)$ is an ideal of $R[x]/\langle x^n - \lambda\rangle$. We will identify $(R)^n$ with $R[x]/\langle x^n - \lambda\rangle$ under p_λ and for simplicity, we write the polynomial $a_0 + a_1x + ... + a_{n-1}x^{n-1}$ instead of the residue class $a_0 + a_1x + ... + a_{n-1}x^{n-1} + \langle x^n - \lambda\rangle$. By this correspondence, to obtain the structure of λ-constacyclic codes over R, we determine the ideals of $R[x]/\langle x^n - \lambda\rangle$.

In this paper, we denote the residue ring $R[x]/\langle x^n - \lambda\rangle$ by $R_{n,\lambda}$. Also, for the non-zero elements $\delta, \theta \in \mathbb{F}_q$, the residue rings $\mathbb{F}_q[x]/\langle x^n - \delta\rangle$ and $(\mathbb{F}_q)^2[x]/\langle x^n - (\delta,\theta)\rangle$ are denoted by $(\mathbb{F}_q)_{n,\delta}$ and $(\mathbb{F}_q)^2_{n,(\delta,\theta)}$ respectively.

The following theorem determines a class of constacyclic codes over R.

Theorem 6.1. [13, Theorem 4.13] *Let C be a λ-constacyclic code over R of length n such that $\gcd(n,p) = 1$. Then $C = \langle g_0(x), ug_1(x)\rangle \subseteq R_{n,\lambda}$, where $g_1(x)|g_0(x)|(x^n - \lambda)$.*

There are many studies about linear constacyclic codes over R of length n which is not coprime to p; see [7, 9, 10, 11]. We recalled the structure of these codes of length p^s from [9].

Theorem 6.2. [9, Theorem 4.4] *Let C be a $(\delta + \theta u)$-constacyclic code over R of length p^s such that $\delta, \theta \in \mathbb{F}_q \setminus \{0\}$ and $s = am + r$ for non-negative integers a, r with $0 \le r \le m-1$. Then $C = \langle(\delta_0 x - 1)^i\rangle \subseteq R_{p^s,\delta+\theta u}$ for some $i \in \{0, 1, \cdots, 2p^s\}$, where $\delta_0 = \delta^{-p^{m-r}}$.*

The above theorem gives the structure of $(\delta + \theta u)$-constacyclic codes of length p^s in the case that δ and θ are non-zero elements in $\mathbb{F}_q$. In the following theorems, we have the structure of these codes in the case that $\theta = 0$. First let $(\delta + \theta u) = 1$. Hence, we have cyclic codes.

Theorem 6.3. [9, Theorem 5.4] *Cyclic codes of length p^s over R, i.e., ideals of the ring $R_{p^s,1}$, are*

1) *Type 1 (trivial ideals):* $\langle 0\rangle$, $\langle 1\rangle$.

2) *Type 2 (principal ideals with non-monic polynomial generators):* $\langle u(x-1)^i\rangle$, *where* $0 \le i \le p^s - 1$.

3) *Type 3 (principal ideals with monic polynomial generators):* $\langle (x-1)^i + u(x-1)^t h(x)\rangle$, *where* $1 \le i \le p^s - 1$, $0 \le t \le i$, *and either $h(x)$ is 0 or $h(x)$ is a unit, where it can be represented as $h(x) = \sum_j h_j(x-1)^j$, with $h_j \in \mathbb{F}_q$, and $h_0 \neq 0$.*

4) *Type 4 (nonprincipal ideals):* $\langle (x-1)^i + u\sum_{j=0}^{\omega-1} c_j(x-1)^j, u(x-1)^\omega\rangle$, *where* $1 \le i \le p^s - 1$, $c_j \in \mathbb{F}_q$, *and* $\omega < T$, *where T is the smallest integer such that $u(x-1)^T \in \langle (x-1)^i + u\sum_{j=0}^{i-1} c_j(x-1)^j$; or equivalently, $\langle (x-1)^i + u(x-1)^t h(x), u(x-1)^\omega\rangle$, with $h(x)$ as in Type 3, and* $\deg(h) \le \omega - t - 1$.

Now, by the structure of cyclic codes in the Theorem 6.3 and the ring isomorphism in the following theorem, we have the structure of δ-constacyclic codes.

Theorem 6.4. [9, Proposition 6.1] *Let δ_0 be defined such as Theorem 6.2. Then the map $\phi : R_{p^s,1} \longrightarrow R_{p^s,\delta}$*

given by $f(x) \longmapsto f(\delta_0 x)$ is a ring isomorphism. In particular, for $A \subseteq R_{p^s,1}$, $B \subseteq R_{p^s,\delta}$ if $\phi(A) = B$, then A is an ideal of $R_{p^s,1}$ if and only if B is an ideal of $R_{p^s,\delta}$. Equivalently, A is a cyclic code of length p^s over R if and only if B is a δ-constacyclic code of length p^s over R.

Constacyclic $\mathbb{F}_q\mathbb{F}_q[u]$-additive codes

Now, by the definition of constacyclic codes in the above subsection, we define constacyclic $\mathbb{F}_q\mathbb{F}_q[u]$-additive codes.

Definition 6.5. Let α and β be two positive integers and $(\lambda_1, \lambda_2) \in \mathbb{F}_q \times R$. An $\mathbb{F}_q\mathbb{F}_q[u]$-additive code $C \subseteq \mathbb{F}_q^\alpha \times R^\beta$ is called (λ_1, λ_2)-constacyclic if

$$(\lambda_1 s_{\alpha-1}, s_0, ..., s_{\alpha-2}, \lambda_2 r_{\beta-1}, r_0, ..., r_{\beta-2}) \in C,$$

whenever $(s_0, s_1, ..., s_{\alpha-1}, r_0, r_1, ..., r_{\beta-1}) \in C$.

Consider the map $\pi_{(\lambda_1,\lambda_2)} : \mathbb{F}_q^\alpha \times R^\beta \longrightarrow (\mathbb{F}_q)_{\alpha,\lambda_1} \times R_{\beta,\lambda_2}$ with the following definition $(s_0, s_1, ..., s_{\alpha-1}, r_0, r_1, ..., r_{\beta-1}) \longmapsto (s_0 + s_1 x + ... + s_{\alpha-1}x^{\alpha-1} + \langle x^\alpha - \lambda_1\rangle, r_0 + r_1 x + ... + r_{\beta-1}x^{\beta-1} + \langle x^\beta - \lambda_2\rangle)$. It is easy to see that $\pi_{(\lambda_1,\lambda_2)}$ is an R-module isomorphism. We will identify $\mathbb{F}_q^\alpha \times R^\beta$ with $(\mathbb{F}_q)_{\alpha,\lambda_1} \times R_{\beta,\lambda_2}$ under $\pi_{(\lambda_1,\lambda_2)}$ and for simplicity, we write $(s_0 + s_1 x + ... + s_{\alpha-1}x^{\alpha-1}, r_0 + r_1 x + ... + r_{\beta-1}x^{\beta-1})$ for above residue class.

Lemma 6.6. *A subset C of $\mathbb{F}_q^\alpha \times R^\beta$ is a (λ_1, λ_2)-constacyclic code if and only if $\pi_{(\lambda_1,\lambda_2)}(C)$ is an $R[x]$-submodule of $(\mathbb{F}_q)_{\alpha,\lambda_1} \times R_{\beta,\lambda_1}$.*

Proof. Clearly, $(\mathbb{F}_q)_{\alpha,\lambda_1} \times R_{\beta,\lambda_2}$ is an $R[x]$-module. Since $\pi_{(\lambda_1,\lambda_2)}$ is an R-module isomorphism, C is an R-submodule if and only if $\pi_{(\lambda_1,\lambda_2)}(C)$ is an R-submodule. Now, for an element $(s_\alpha, r_\beta) = (s_0, s_1, ..., s_{\alpha-1}, r_0, r_1, ..., r_{\beta-1}) \in C$, let $\sigma(s_\alpha, r_\beta) = (\lambda_1 s_{\alpha-1}, s_0, ..., s_{\alpha-2}, \lambda_2 r_{\beta-1}, r_0, ..., r_{\beta-2})$. Thus

$$\sigma(s_\alpha, r_\beta) \in C \Leftrightarrow x\pi_{(\lambda_1,\lambda_2)}(s_\alpha, r_\beta) =$$

$$\pi_{(\lambda_1,\lambda_2)}(\sigma(s_\alpha, r_\beta)) \in \pi_{(\lambda_1,\lambda_2)}(C).$$

This completes the proof. □

We identify C with $\pi_{(\lambda_1,\lambda_2)}(C)$. Now, we find the generator polynomials of C.

Theorem 6.7. *A subset C of $(\mathbb{F}_q)_{\alpha,\lambda_1} \times R_{\beta,\lambda_2}$ is a (λ_1, λ_2)-constacyclic code if and only if $C = \langle (g, 0), (h_1, f_1), ..., (h_r, f_r)\rangle_{R[x]}$ such that*

(1) $C_1 = \langle g\rangle$ is a λ_1-constacyclic code over $\mathbb{F}_q$ of length α,

(2) $C_2 = \langle f_1, ..., f_r\rangle_{R[x]}$ is a λ_2-constacyclic code over R of length β,

(3) $g | x^\alpha - \lambda_1$ over $\mathbb{F}_q$,

(4) $h_1, ..., h_r$ are elements of $(\mathbb{F}_q)_{\lambda_1,\alpha}$,

(5) $|C| = |C_1||C_2|$.

Proof. Let $C \subseteq (\mathbb{F}_q)_{\alpha,\lambda_1} \times R_{\beta,\lambda_2}$ be a (λ_1, λ_2)-constacyclic code. Clearly, the projection map $\phi : C \longrightarrow R_{\beta,\lambda_2}$ is an $R[x]$-homomorphism. Hence, $Im(\phi)$ is an $R[x]$-submodule of R_{β,λ_2}. As $\langle x^\beta - \lambda_2\rangle.Im(\phi) \subseteq \langle x^\beta - \lambda_2\rangle.R_{\beta,\lambda_2} = 0$, $Im(\phi)$ is an ideal of R_{β,λ_2}. In other words, $Im(\phi)$ is a linear λ_2-constacyclic code over R of length β, say C_2. Let $C_2 = \langle f_1, ..., f_r\rangle_{R[x]} = \langle \phi(h_1, f_1), ..., \phi(h_r, f_r)\rangle_{R[x]}$. Now, $\ker\phi$ is an $R[x]$-submodule of C. Let $C_1 = \{g \in (\mathbb{F}_q)_{\alpha,\lambda_1} : (g, 0) \in \ker\phi\}$, then clearly C_1 is an $R[x]$-submodule of $(\mathbb{F}_q)_{\alpha,\lambda_1}$. But the map $\tau : R \longrightarrow \mathbb{F}_q$, in the R-module structure of $\mathbb{F}_q$, is surjective. Hence, C_1 is an $\mathbb{F}_q[x]$-submodule of $(\mathbb{F}_q)_{\alpha,\lambda_1}$. Since $\langle x^\alpha - \lambda_1\rangle.C_1 \subseteq \langle x^\alpha - \lambda_1\rangle.(\mathbb{F}_q)_{\alpha,\lambda_1} = 0$, C_1 is an ideal of $(\mathbb{F}_q)_{\alpha,\lambda_1}$. In other words, C_1 is a λ_1-constacyclic code over $\mathbb{F}_q$ of length α. If $C_1 = \langle g\rangle$, then $\ker\phi = \langle (g, 0)\rangle_{R[x]}$. Therefore, $C = \langle (g, 0), (h_1, f_1), ..., (h_r, f_r)\rangle_{R[x]}$. Since ϕ is an $R[x]$-homomorphism, $\frac{C}{\ker\phi} \cong C_2$, hence $|C| = |\ker\phi||C_2| = |C_1||C_2|$. □

Proposition 6.8. *With the assumptions of Theorem 6.7, let*

$$C = \langle (g, 0), (h_1, f_1), ..., (h_r, f_r)\rangle_{R[x]}$$

be a (λ_1, λ_2)-constacyclic code. Then, we can assume that $\deg h_i < \deg g$ for all i; $1 \leq i \leq r$.

Proof. Since the coefficients of g are invertible, we assume that g is monic. Let $\deg h_i \geq \deg g$ for some i; $\deg h_i - \deg g_j = \ell \geq 0$. Also, let $a \in \mathbb{F}_q$ be the leading coefficient of h_i. Then $(h_i, f_i) = (h_i - ax^\ell g, f_i) + ax^\ell(g, 0)$. Thus, $\langle (h_i, f_i), (g, 0)\rangle = \langle (h_i - ax^\ell g, f_i), (g, 0)\rangle$. Hence, we can use $h_i - ax^\ell g$ instead of h_i. By this method we can reduce $\deg h_i$. □

Proposition 6.9. *Let*

$$C = \langle (g, 0), (h_1, f_1), ..., (h_r, f_r)\rangle_{R[x]}$$

be a (λ_1, λ_2)-constacyclic code. Then $g \mid (x^\beta - \lambda_2)h_i$, for all $i = 1, 2, ..., r$.

Proof. Consider the projection map $\phi : C \longrightarrow R_{\beta,\lambda_2}$ in the proof of Theorem 6.7. Then $C_1 = \langle g\rangle = \{f \in$

$(\mathbb{F}_q)_{\alpha,\lambda_1}: (f,0) \in \ker\phi\}$. Now, $(x^\beta-\lambda_2)(h_i,f_i) = ((x^\beta-\lambda_2)h_i,0) \in \ker\phi$. Hence, $(x^\beta-\lambda_2)h_i \in \langle g\rangle$. This completes the proof. □

Remark 6.10. If $(\lambda_1,\lambda_2)=(1,1)$ as in Definition 6.5, then we have cyclic $\mathbb{F}_q\mathbb{F}_q[u]$-additive codes. Hence, by above results, we can obtain the structure of cyclic $\mathbb{F}_q\mathbb{F}_q[u]$-additive codes.

Now, we give the exact structure of constacyclic $\mathbb{F}_q\mathbb{F}_q[u]$-additive codes for some special lengths.

Corollary 6.11. *Let $C \subseteq (\mathbb{F}_q)_{\alpha,\lambda_1} \times R_{\beta,\lambda_2}$ be a (λ_1,λ_2)-constacyclic code. If α and β are coprime to p, then $C = \langle(g,0),(h_1,f_1),(h_2,uf_2)\rangle_{R[x]}$ such that $g|(x^\alpha-\lambda_1)$ and $f_2 \mid f_1|(x^\beta-\lambda_2)$*

Proof. It follows from Theorems 6.1 and 6.7. □

Let $b \in \mathbb{F}_q$ and s be a positive integer. If $s = am+r$ for non-negative integers a,r with $0 \le r \le m-1$, then define $b_0 = a^{-p^{m-r}}$. The following theorem determines all constacyclic $\mathbb{F}_q\mathbb{F}_q[u]$-additive codes in the case that α and β are powers of p.

Theorem 6.12. *Let $C \subseteq (\mathbb{F}_q)_{\alpha,\lambda_1} \times R_{\beta,\lambda_2}$ be a (λ_1,λ_2)-constacyclic code. Assume that $\alpha = p^{s_1}, \beta = p^{s_2}, \lambda_1 = \gamma$ and $\lambda_2 = \delta + \theta u$. Then $C = \langle((\gamma_0 x+1)^i,0),(h,(\delta_0 x-1)^j)\rangle$ for some $i \in \{0,1,\cdots,p^{s_1}\}$ and $j \in \{0,1,\cdots,2p^{s_2}\}$.*

Proof. By [9, Theorem 3.4], $C_1 = \langle(\gamma_0 x+1)^i\rangle$, for some $i \in \{0,1,\cdots,p^{s_1}\}$. Also, by Theorem 6.2, $C_2 = \langle(\delta_0 x-1)^j\rangle$, for some $j \in \{0,1,\cdots,2p^{s_2}\}$. Now, by Theorem 6.7, we have the result. □

Theorem 6.13. *With the assumptions of the Theorem 6.12, let $\theta = 0$ and $C \subseteq (\mathbb{F}_q)_{\alpha,\lambda_1} \times R_{\beta,\lambda_2}$ be a (γ,δ)-constacyclic code. Then,*

$$C = \langle(g,0),(h_1,f_1),...,(h_r,f_r)\rangle_{R[x]},$$

such that

(1) $C_1 = \langle g\rangle = \langle(\gamma_0 x+1)^i\rangle$ *for some* $i \in \{0,1,\cdots,p^{s_1}\}$,

(2) $C_2 = \langle f_1,...,f_r\rangle_{R[x]} = \phi(I)$ *where* $\phi : R_{p^s,1} \longrightarrow R_{p^s,\delta}$ *is given by* $f(x) \longmapsto f(\delta_0 x)$ *and I is an ideal of $R_{p^{s_2},1}$ defined in Theorem 6.3.*

Proof. It follows from Theorems 6.3, 6.4 and 6.7. □

7. Conclusion

In this paper, we studied the structure of $\mathbb{F}_q\mathbb{F}_q[u]$-additive codes. We obtained the generator matrix of these codes and described their dual codes. We defined Lee weight and homogenous weight over $\mathbb{F}_q\mathbb{F}_q[u]$-additive codes and studied one-weight codes with respect to these two weight functions. Finally, by the Gray image of these codes, we obtained $[q^2+q,2,q^2]$ and $[2(q+1),2,2q]$ one-weight optimal codes over $\mathbb{F}_q$.

References

[1] Abualrub T, Siap I and Aydin N. $\mathbb{Z}_2\mathbb{Z}_4$-additive cyclic codes. IEEE Trans. Inform. Theory 2011;60(3):1508-1514.

[2] Aydogdu I, Abualrub T and Siap I. On $\mathbb{Z}_2\mathbb{Z}_2[u]$-additive codes. Int. J. Inf. comput. math 2015;92(9):1806-1814.

[3] Bilal M, Borges J, Dougherty S. T and Fernández-Córdoba C. Maximum distance separable codes over $\mathbb{Z}_4$ and $\mathbb{Z}_2 \times \mathbb{Z}_4$. Des. Codes Cryptogr 2011;31-40.

[4] Borges J and Fernández-Córdoba C. There is exactly one $\mathbb{Z}_2\mathbb{Z}_4$-cyclic 1-perfect code. Des. Codes Cryptogr 2016.

[5] Borges J, Fernández-Córdoba C, Pujol J, Rifa J and Villanueva M. $\mathbb{Z}_2\mathbb{Z}_4$-linear codes: generator matrices and duality. Des. Codes Cryptogr 2010;54:167-179.

[6] Carlet C. $\mathbb{Z}_{2^k}$-linear Codes. IEEE Trans. Inform. Theory 1998;44:1543-1547.

[7] Chen B, Dinh H. Q, Liu H and Wang L. Constacyclic codes of length $2p^s$ over $\mathbb{F}_{p^m}+u\mathbb{F}_{p^m}$. Finite fields Appl 2016;37:108-130.

[8] Chen B, Fan Y, Lin L and Liu H. Constacyclic codes over finite fields. Finite fields Appl 2012;18:1217-1231.

[9] Dinh H. Q. Constacyclic codes of length p^s over $\mathbb{F}_{p^m}+u\mathbb{F}_{p^m}$. J. Algebra 2010;324:940-950.

[10] Dinh H. Q, Dhompongsa S and Sriboonchitta S. On constacyclic codes of length $4p^s$ over $\mathbb{F}_{p^m}+u\mathbb{F}_{p^m}$. Discrete Math 2016;http://dx.doi.org/10.1016/j.disc.2016.11.014.

[11] Dinh H. Q, Nguyen B. T, Sriboonchitta S and Vo T. On $(\alpha+u\beta)$-constacyclic codes of length $4p^s$ over $\mathbb{F}_{p^m}+u\mathbb{F}_{p^m}$. J. Algebra Appl 2019; http://dx.doi.org/10.1142/S0219498819500233.

[12] Dougherty S. T, Liu H and Yu l. One weight $\mathbb{Z}_2\mathbb{Z}_4$-additive codes. Appl. Algebra Eng. Commun. Comput 2016;27:123-138.

[13] Guenda k and Gulliver T. A. MDS and self-dual codes over rings. Finite fields Appl 2012;18:1061-1075.

[14] Hammons A.R, Kumar K, Calderbank A.R, Sloane N.J.A and Sole P. The $\mathbb{Z}_4$- linearity of Kerdock Preparata, Goethals and related codes. IEEE Trans. Inform. Theory 1994;40:301-319.

[15] Huffman W. C and Pless V. Fundamentals of Error-Correcting Codes. Cambridge University Press, Cambridge 2003.

[16] Jitman J and Udomkavanich P. The Gray image of codes over finite chain rings. International Journal of Contemporary Mathematical Sciences 2010;5(10):449-458.

[17] Lam T.Y. Lectures on Modules and Rings. Springer, New York 1998.

[18] Ling S and Blackford J.T. $\mathbb{Z}_{p^{k+1}}$-linear codes. IEEE Trans. Inform. Theory 2002;48:2592-2605.

[19] Lu Z, Zhu S, Wang L and Kai X. One-Lee weight and two-Lee weight $\mathbb{Z}_2\mathbb{Z}_2[u]$-additive codes. 2018;http://arxiv.org/abs/1609.09588v1.

[20] McDonald B. R. Finite Rings with Identity. Marcel Dekker, Inc., New York 1974.

[21] Qian L and Cao X. Bounds and optimal q-ary codes derived from the $\mathbb{Z}_q R$-cyclic codes. IEEE Trans. Inform. Theory 2019;http://dx.doi.org/10.1109/TIT.2019.2927034.

[22] Samei K and Mahmoudi S. Singlton bounds for R-additive codes. Adv. Math. Commun 2018;12:107-114.

[23] Sharma A and Rani S. On constacyclic codes over finite fields. Cryptogr. Commun 2015;http://dx.doi.org/10.1007/s12095-015-0163-4.

[24] Singleton R.C. Maximum distance q-nary codes. IEEE Trans. Inf. Theory 1964;10(2):116–118.

[25] Srinivasulu B and Bhaintwal M. $\mathbb{Z}_2(\mathbb{Z}_2 + u\mathbb{Z}_2)$-additive cyclic codes and their duals. Discrete Math. Alg. and Appl 2016;8(2):1-19.

[26] Wood J. Duality for modules over finite rings and applications to coding theory. Amer. J. Math 1999;121(3):555–575.

Journal of Optimization of Soft Computing (JOSC)

Vol. 2, Issue 1, pp: (19-25), June-2024

Journal homepage: https://sanad.iau.ir/journal/josc

Research paper

Using the fuzzy methods to examine changes in brain lesions and atrophy from MRI images for rapid diagnosis of MS

Alireza Banitalebi Dehkordi[1*]

1. College of skills and entrepreneurship, Islamic Azad University, Shahrekord branch, Shahrekord, Iran

Article Info

Article History:
Received: 2024/05/03
Revised: 2024/05/24
Accepted: 2024/06/10

DOI:

Keywords:
MRI, Multiple sclerosis, Fuzzy method, Controlled watershed.

**Corresponding Author's Email Address: banitalebe@gmail.com*

Abstract

Multiple sclerosis (MS) is a disease that affects the central nervous system, during which the myelin present on the nerve fibers that have a protective role is destroyed, and therefore the conduction of electric current is disturbed and the symptoms of MS disease appear. In this disease, the white blood cells that play a defensive role in the body attack the myelin, which is a protection for nerve fibers, as a foreign agent, and each time these blood cells attack the nerve fibers of one of the organs of the patient's body. Which is unclear, that organ will have problems. The best way to diagnose MS is to examine brain MRI images. Therefore, the existence of a fast and accurate method to evaluate changes in brain atrophy or the creation and increase of lesions (plaques) caused by this disease is a key component in diagnosing and evaluating the progress of the disease and the effectiveness of its treatment courses. Manual detection of changes in lesions (plaques) and brain atrophy caused by this disease usually requires a trained specialist and is very slow and difficult, and the results are somewhat subjective. Therefore, the existence of an automatic system for extracting and checking these changes is essential. Although many automatic methods have been proposed, the segmentation results are not accurate enough. As a result, there is a great need to develop a strong, fast, and accurate method to diagnose MS and brain lesions caused by it. In this article, by combining two fuzzy methods and the controlled watershed algorithm, we propose a fast method with high accuracy to diagnose MS from brain MR images.

1. Introduction

Paying attention to MS or multiple sclerosis is very important and necessary because we still do not know its treatment. However, scientists are working on it and hopefully in the near future we will be able to successfully assess its behavior and eventually treat it. We now know that the protective covering of axons is made of myelin, a white fatty substance, which is lost in patients suffering from MS. As a result, they experience severe sensory or motor impairment. There are different types of MS and different stages of the disease. Some patients report that their condition keeps going back and forth. This is because relapsing-remitting MS is a condition where the myelin sheath is destroyed but can be repaired, but unfortunately, it is destroyed again after a

while. On the other hand, progressive MS leads to neuronal death caused by axonal damage, which is an irreversible state. A review of past works shows that the most important factor in the process of treating the disease is the rapid diagnosis of the disease. Rondinella, A. (2023) and colleagues in their article have used a U-Net structure reinforced with a convolutional short-term memory layer to more accurately segment and measure multiple sclerosis lesions detected in magnetic resonance images. To cope with inter- and intra-observer variability and reduce the burden and complexity of lesions identification for clinicians, a large number of techniques have been proposed in the literature for the automatic segmentation of MS lesions (see Garcia-Lorenzo et al., 2013; Valverde et al., 2017; Danelakis et al., 2018 for reviews). Several challenges have been proposed to evaluate the performances of these methods (e.g., Carass et al., 2017; Commowick et al., 2021 to cite the most recent ones). Moreover, recently Bonacchi et al. (2022) proposed an overview of Artificial Intelligence applications for MS clinical practice. Initial studies mainly enrolled patients with longstanding severe progressive multiple sclerosis, when inflammatory features are less prominent and neurodegeneration is the main underlying mechanism (Trapp and Nave, 2008). Benefit generally was modest, although some patients exhibited sustained slowing or stabilization of disability, but improvement in neurologic function was rarely seen (Burt et al., 2015; Mancardi et al., 2015). Also, patients with more severe neurologic disability had increased risk of adverse events (Mancardi and Saccardi, 2008). More recent studies (Table 1) focused on relapsing-remitting multiple sclerosis and demonstrated that patients with active inflammatory features appear to derive the most benefit from this approach (Burt et al., 2012; Saccardi et al., 2012; Muraro et al., 2017).

In this article, we present a new method of rapid diagnosis of MS disease by examining changes in brain lesions and atrophy from MRI images using fuzzy methods.

2. Materials and Methods

Please be sure your sentences are complete and that there is continuity within your paragraphs. Check the numbering of your graphics and make sure that all proper references are included.

3. Page Layout

Since image noise and heterogeneity have major effects on non-brain tissue such as skull and fat, non-brain tissue is removed from the patient's MR image in order to reduce calculation time and increase efficiency. In MR brain image slices, by using two important factors to recognize the skull, i.e. thresholding on the intensity of light and also considering the approximate width of the skull, we reach very good results for removing the skull. In order to remove the skull, we first extract the image from each slice by considering a suitable threshold limit for only the skull, and then by comparing it with the original image, we separate the skull from the image in all slices of the MR image. Because MR images encounter different types of noise such as Gaussian, Poisson, Rayleigh and impact noise (pepper-salt). Therefore, the use of noise reduction and removal methods to improve the quality of these images will be vital for better diagnosis of diseases. Therefore, in the next step, using morphological operators, the noise is reduced as much as possible and the image quality is improved. First, by using the opening operation, a part of the broken narrow paths or small protrusions caused by fluctuations or non-linear effects are removed from the image. Then, in order to further reduce and eliminate noise, we use the combined opening-closing operation. By doing this, the effects of the remaining noise on the edges and also on the edges of the narrow edges of the image will be removed from the image, and an image will be obtained in which the edge noise has almost disappeared and the edges will be easily distinguishable. Figure 1 shows the result of removing the skull with this method from the middle slice of the T1-w weighted MR image.

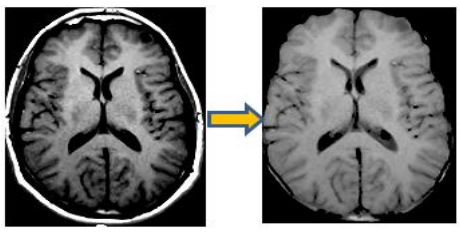

Figure 1: Removing the skull from a slice of an MRI brain image

After removing the skull using two factors of approximate width and thresholding, the area of each slice is measured and recorded so that they can be compared with each other over time and in periodic MRIs of the patient. Also, we consider the effect coefficient of each slice on the brain

volume based on the area of each slice. We calculate the coefficient of 1 for the slice that has the largest area, and the effect coefficient of the rest of the slices is calculated by dividing the area of that slice by the area of the largest slice. Since the lesions (plaques) in MS occur only in the white area of the brain, in order to increase the accuracy of the system, first, the white, gray and cerebrospinal fluid areas of the brain are extracted using the fuzzy c-mean algorithm. Now, if the detected lesion is in the white area of the brain, then the segmentation is correct and the lesion is accepted and goes to the next stage, otherwise the lesion is removed. Another factor is measuring and comparing the area of each of the white, cerebrospinal, and gray areas of the brain, which should be evaluated over time on periodic MRIs of patients.

C means clustering algorithm:

In this algorithm, the number of C clusters is specified in advance. The objective function defined for this algorithm is as follows:

$$J = \sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^{m} d_{ik}^{2} = \sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^{m} \left\| x_k - v_i \right\|^2$$

Equation 1

In equation 1, m is a real number greater than 1, which is chosen as 2 for m in most cases. is the kth sample and is the representative or center of the ith cluster. It shows the membership of the i-th sample in the k-th cluster. The symbol ||*|| The degree of similarity (distance) of the sample with (from) the center of the cluster, which can be used any function that expresses the similarity of the sample and the center of the cluster. A U matrix can be defined from and its components can take any value between 0 and 1. If all the components of the matrix U are 0 or 1, the algorithm will be similar to k average, although the components of the matrix U can take any value between 0 and 1 but the sum of the components of each of the columns must be equal to 1 and we have:

$$\sum_{i=1}^{c} u_{ik} = 1, \forall k = 1,..., n$$

Equation 2

This condition states that the membership of each sample to c clusters must be equal to 1. Using this condition and minimizing the objective function, we will have:

$$v_i = \frac{\sum_{k=1}^{n} u_{ik}^{m} x_k}{\sum_{k=1}^{n} u_{ik}^{m}}$$

Equation 3

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{d_{ik}}{d_{jk}}\right)^{2/m-1)}}$$

Equation 4

In this algorithm, the initial values for c, m and U^0 are done, the initial clusters are guessed and the centers of the clusters are calculated. Then the membership matrix is obtained from the calculated clusters. Now, if equation 5 is true, the algorithm ends, and otherwise, the algorithm returns to the previous step.

$$\left\| U1+1-U1 \right\| \le \varepsilon$$

Equation 5

The use of the fuzzy algorithm makes the curve of the membership function smoother than the classical k-means algorithm, and the border between the clusters is not defined accurately and definitively. In order to segment the brain in the proposed algorithm, the number of clusters is considered equal to 3, which includes white area, cerebrospinal fluid and gray area of the brain. After clustering, the gray area and CSF are removed so that only the white area of the brain remains and the search space is limited to this area. Figure 2 shows the results of this clustering. In the testing phase of the system, after the lesions are detected, only the lesions located in the white area of the brain are accepted, and otherwise they are removed.

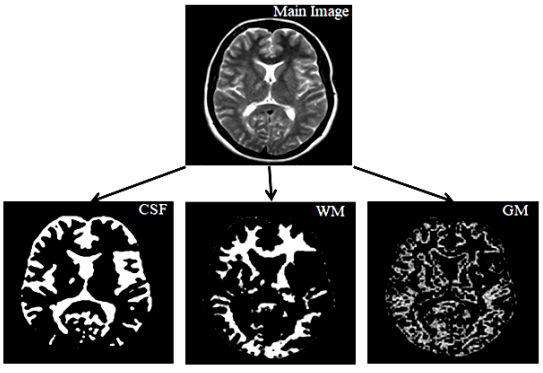

Figure 2 : The results of clustering by the c-means fuzzy algorithm by selecting three clusters

In the next step, in order to optimally separate the brain regions, the average brightness intensity of each image slice is taken and the corresponding coefficients are obtained according to the average so that an optimal threshold is applied to the corresponding slice for each MR image. By comparing the results of the implementation of the proposed system of images used in two cases, once with noise removal by Gaussian filter and once without noise removal, it was determined that due to the removal of noise, some of the information related to the texture was also removed or They change, so noise removal methods should not be used as much as possible.

Diagnosis of lesioned hemispheres

In the diagnosis of hemispheres with MS lesions, the main idea is the asymmetry of the histogram between the healthy and lesioned hemispheres. Therefore, the hemispheres of the brain must be extracted first. Dividing the brain into two hemispheres is done by finding the diameter of the oval containing the brain. The advantage of this method is its resistance to head rotation and patient movement during imaging. For this purpose, after the complete extraction of the brain, the obtained image becomes binary. The resulting mask has an area that can be separated from the right and left hemispheres by determining the ellipse containing the area and considering its large diameter as the midline of the brain. (Nabizadeh and Kubat, 2015) Figure 3 shows the separation of the right and left hemispheres of all slices of an MR image.

4. Diagnosis of lesioned hemispheres

In the diagnosis of hemispheres with MS lesions, the main idea is the asymmetry of the histogram between the healthy and lesioned hemispheres. Therefore, the hemispheres of the brain must be extracted first. Dividing the brain into two hemispheres is done by finding the diameter of the oval containing the brain. The advantage of this method is its resistance to head rotation and patient movement during imaging. For this purpose, after the complete extraction of the brain, the obtained image becomes binary. The resulting mask has an area that can be separated from the right and left hemispheres by determining the ellipse containing the area and considering its large diameter as the midline of the brain. (Nabizadeh and Kubat, 2015) Figure 3 shows the separation of the right and left hemispheres of all slices of an MR image.

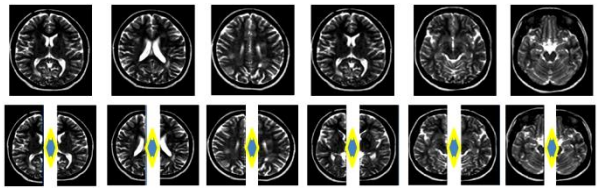

Figure 3: Separation of the right and left hemispheres of all slices of an MR image

After separating the right and left hemispheres of the brain, in the next step, we want to determine the hemisphere containing MS based on histogram information. In the algorithm used in this article, both training and test data, the number of slices for all samples in the neural network are selected equally. In this case, the corresponding slices from different samples represent the same part of the brain and therefore have the same structure. Due to this, it will be possible to create a standard histogram for healthy hemispheres for all slices using the training data. The standard histogram of all slices is extracted using the average histogram of the healthy hemispheres and fitting the Gaussian function in the training phase. In the system test phase, the obtained standard histogram is compared with the histogram of the test data and the hemisphere with the lesion is determined. To check the histograms, the sum of squared errors was used. Figure 5 shows the standard histogram extracted from the middle slice of the MR image and the histograms for the left and right hemispheres of this slice of the brain, which is slice number 255 of the database images of a 40-year-old healthy woman.

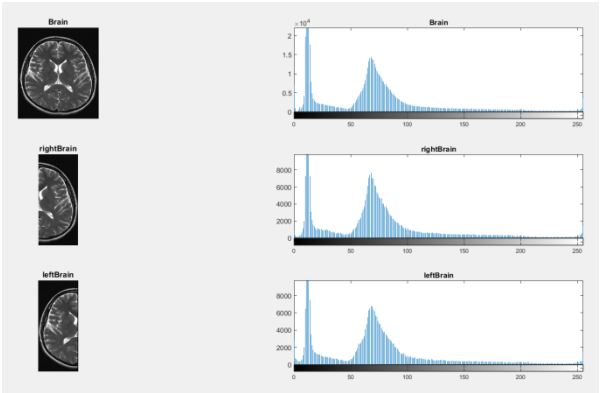

Figure 4 : The standard histogram extracted from the middle slice of the MR image and the histograms related to the left and right hemispheres of this slice from the brain of a 40-year-old healthy woman

In Figure 5, as can be seen, the standard histogram and the histogram of the left and right hemispheres of the middle slice for a 45-year-old female patient with MS have one peak, but the histogram of the hemisphere with a lesion has

another peak, which is actually related to the brightness of the lesion.

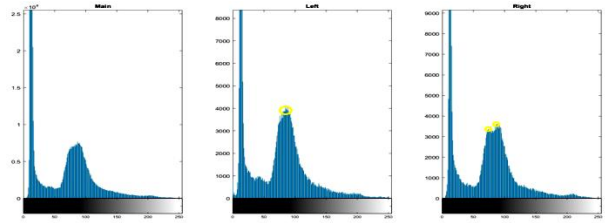

Figure 5: Standard histogram,histogram related to the left and right hemispheres of a 47-year-old female patient with MS.

In this method, by using the neural network by comparing between the hemispheres and comparing it with the histogram samples of healthy people, in addition to distinguishing sick from healthy people, it is possible to identify the hemisphere that is more involved in MS disease from the peaks created on the histogram. can be identified. Therefore, the search space of our segmentation step is reduced in this method. Since the number of slices in the samples may not be equal according to the imaging diameter, in this case it is possible to identify the disease separately for that person by comparing the histogram of the two hemispheres of the same person. In this case, if the two histograms are different from each other, it means that the person has MS. But it is not possible to accurately determine the hemisphere containing the lesion, so both hemispheres should be examined after segmentation.

At the end, in the evaluation stage of the neural network, we first obtain the distance between the standard histogram and the histogram of the investigated hemisphere by the sum of squared error method, and by considering a suitable threshold limit, we determine the presence or absence of a lesion in the hemisphere. The size of the lesion determined by this method is dependent on the determined threshold value, if this value is high, small lesions will be easily recognized, but some false positive error will increase, because some healthy slices have a diagnosis lesion. will be given On the other hand, a smaller threshold value reduces the amount of false positive error, but it will no longer be possible to detect small plaques related to MS. Therefore, in order to fully extract the images with lesions, the threshold value is considered to be relatively small so that the hemispheres in which there is a high probability of having MS can go to the next stage and be examined more closely, on the other hand, the images that are rejected in this stage are very unlikely to have MS. In other words, the false negative rate of the system will decrease. In order to extract the feature, we first sweep each hemisphere by a sliding window. We have considered the distance between the central pixel location of the window and the next location to be 5 pixels to increase the processing speed. As a result, the window will overlap at the current and next location. In the training phase, the hemispheres containing the lesion and healthy are cut and windows are obtained from them. In each window, the features are extracted and the target class is trained based on whether each training window has a lesion or not. In the test phase, a window of the same size as the window of the training phase is considered and moved throughout the brain and background area. The features are extracted and classified in this window and in any place where it is placed. Based on the classification results, the center of the window based on belonging to the lesion site or healthy brain tissue is marked. In order to determine the appropriate size of the window, two windows with sizes of 8 x 8 and 16 x 16 are considered.

After the automatic diagnosis of patients' images with the help of neural network, to determine the progress of the disease, it is necessary to extract MS plaques from the image slices and provide them to the doctor. Considering that MS plaques are part of the brain tissue, their accurate segmentation and separation will be of great help in the treatment of the disease by the doctor. In this article, in order to increase the accuracy in the separation of MS plaques, the water conversion method of marker-controlled diffusers and the integration of similar areas have been used. At first, the change is applied by applying a 6x6 Gaussian Laplacian edge filter to calculate the approximation accuracy of the gradient amplitudes. In the segmentation stage, the marker-controlled water spreader (MCWS) method is applied to the gradient domain of the obtained image. Finally, the over-segmented areas are selected and merged using histogram similarity. To obtain a better filtered image, we have used the functions of illumination and dilation, and to improve the results, opening and closing are used. In the proposed algorithm, since the image is of gray scale type, the domain gradient is used as a segmentation function. The results of the segmentation function result in an image that has the foreground and background markers that we target for segmentation. In the segmented output of the final diffuser, the original image of cerebral MS plaques is extracted from the original MRI

image. In the figure below, images of plaque extraction from slices for a 47-year-old female patient can be seen. In this figure, inside the yellow dots, you can see the plaques that were correctly extracted from the image by this method, and the black dots inside the red circles are the points that were wrongly detected as plaques.

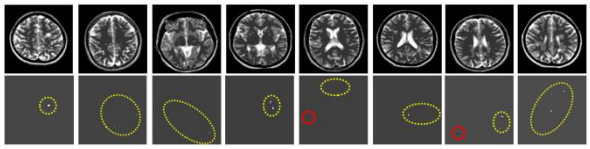

Figure 6: How to extract plaques from the middle eight slices of the MR image of a 47-year-old female patient with MS

This processing was done on 55 people including 28 female patients and 27 male patients. The images were used from the archives of Pars Imaging Center and Hajar Shahrekord Hospital. In total, 98.7% accuracy was recorded for the separation of MS plaques with this method. In similar methods used in other articles, the accuracy of separation of MS plaques was 94.8%, which shows that the method used in this article has a much higher accuracy.[3,5,10,14]

5. Conclusion

MS disease occurs at a young age, usually between 20 and 40 years old and because its complications affect patients from a young age to the end of life, therefore, its rapid diagnosis is necessary to start treatment in the early stages of affected patients. It prevents further progress and prevents severe complications in the patient's body. The proposed method in this article, considering that it is able to automatically diagnose the disease with high speed and accuracy, can be of great help to patients. The combination of fuzzy methods with the controlled watershed algorithm has enabled us to quickly diagnose the disease by comparing the histogram of the two hemispheres of the brain and to extract the plaques caused by the disease using the controlled watershed method. In the next step, after starting the treatment and repeating the MR imaging, we can use this method to evaluate the changing status of the plaques over time. The use of this method can be important in the process of examining the patient's condition in a long period of time and the effectiveness of the treatment process by the doctor.

References

1. Abhale, P., Lashkare, A., & Deshpande, A. (2022a). Early Stage Detection of Multiple Sclerosis using FCNN. 2022 10th International Conference on Emerging Trends in Engineering and Technology - Signal and Information Processing (ICETET-SIP-22). https://doi.org/10.1109/icetet-sip-2254415.2022.9791566
2. Ahmadi, A., Davoudi, S., & Daliri, M. R. (2019). Computer Aided Diagnosis System for multiple sclerosis disease based on phase to amplitude coupling in covert visual attention. Computer Methods and Programs in Biomedicine, 169, 9–18. https://doi.org/10.1016/j.cmpb.2018.11.006
3. Carass, A., Roy, S., Jog, A., Cuzzocreo, J. L., Magrath, E., Gherman, A., Button, J., Nguyen, J., Prados, F., Sudre, C. H., Cardoso, M. J., Cawley, N., Ciccarelli, O., Wheeler-Kingshott, C., Ourselin, S., Catanese, L., Deshpande, H., Maurel, P., Commowick, O., . . . Pham, D. L. (2017). Longitudinal multiple sclerosis lesion segmentation: Resource and challenge. NeuroImage, 148, 77–102. https://doi.org/10.1016/j.neuroimage.2016.12.064
4. Coll, L., Pareto, D., Carbonell-Mirabent, P., Cobo-Calvo, Á., Arrambide, G., Vidal-Jordana, Á., Comabella, M., Castilló, J., Rodríguez-Acevedo, B., Zabalza, A., Galán, I., Midaglia, L., Nos, C., Salerno, A., Auger, C., Alberich, M., Río, J., Sastre-Garriga, J., Oliver, A., . . . Tur, C. (2023). Deciphering multiple sclerosis disability with deep learning attention maps on clinical MRI. NeuroImage. Clinical, 38, 103376. https://doi.org/10.1016/j.nicl.2023.103376
5. De Santiago, L., Morla, E. M. S., Ortiz, M., López, E., Usanos, C. A., Alonso-Rodríguez, M. C., Barea, R., Cavaliere-Ballesta, C., Fernández, A., & Boquete, L. (2019). A computer-aided diagnosis of multiple sclerosis based on mfVEP recordings. PloS One, 14(4), e0214662.

https://doi.org/10.1371/journal.pone.0214662

6. Frischer, J. M., Bramow, S., Dal-Bianco, A., Lucchinetti, C. F., Rauschka, H., Schmidbauer, M., Laursen, H., Sörensen, P. S., & Lassmann, H. (2009). The relation between inflammation and neurodegeneration in multiple sclerosis brains. Brain, 132(5), 1175–1189. https://doi.org/10.1093/brain/awp070
7. García-Lorenzo, D., Prima, S., Arnold, D. L., Collins, D. L., & Barillot, C. (2011). Trimmed-Likelihood estimation for focal lesions and tissue segmentation in multisequence MRI for multiple sclerosis. IEEE Transactions on Medical Imaging, 30(8), 1455–1467. https://doi.org/10.1109/tmi.2011.2114671
8. Hayama, R., Sarid-Krebs, L., Richter, R., Fernández, V., Jang, S., & Coupland, G. (2017). PSEUDO RESPONSE REGULATORs stabilize CONSTANS protein to promote flowering in response to day length. EMBO Journal, 36(7), 904–918. https://doi.org/10.15252/embj.201693907
9. Kester, L., & Van Oudenaarden, A. (2018). Single-Cell transcriptomics meets lineage tracing. Cell Stem Cell, 23(2), 166–179. https://doi.org/10.1016/j.stem.2018.04.014
10. Lenchik, L., Heacock, L., Weaver, A. A., Boutin, R. D., Cook, T. S., Itri, J. N., Filippi, C. G., Gullapalli, R. P., Lee, J., Zagurovskaya, M., Retson, T., Godwin, K., Nicholson, J., & Narayana, P. A. (2019). Automated Segmentation of tissues using CT and MRI: A Systematic review. Academic Radiology, 26(12), 1695–1706. https://doi.org/10.1016/j.acra.2019.07.006
11. Mancardi, G. (2009). Further data on autologous haemopoietic stem cell transplantation in multiple sclerosis. Lancet Neurology, 8(3), 219–221. https://doi.org/10.1016/s1474-4422(09)70018-3
12. Pieterse, C. M. J., Pierik, R., & Van Wees, S. (2014). Different shades of JAZ during plant growth and defense. New Phytologist, 204(2), 261–264. https://doi.org/10.1111/nph.13029
13. Popescu, V. A., Ran, N., Barkhof, F., Chard, D., Wheeler-Kingshott, C. a. M., & Vrenken, H. (2014). Accurate GM atrophy quantification in MS using lesion-filling with co-registered 2D lesion masks. NeuroImage. Clinical, 4, 366–373. https://doi.org/10.1016/j.nicl.2014.01.004
14. Rondinella, A., Crispino, E., Guarnera, F., Giudice, O., Ortis, A., Russo, G., Di Lorenzo, C., Maimone, D., Pappalardo, F., & Battiato, S. (2023). Boosting multiple sclerosis lesion segmentation through attention mechanism. Computers in Biology and Medicine, 161, 107021. https://doi.org/10.1016/j.compbiomed.2023.107021
15. Scolding, N., Pasquini, M. C., Reingold, S. C., & Cohen, J. A. (2017). Cell-based therapeutic strategies for multiple sclerosis. Brain, 140(11), 2776–2796. https://doi.org/10.1093/brain/awx154
16. Snowden, J. A., Akil, M., & Kiely, D. G. (2013). Improving safety in autologous HSCT for systemic sclerosis. Lancet, 381(9872), 1081–1083. https://doi.org/10.1016/s0140-6736(12)62176-x

Journal of Optimization of Soft Computing (JOSC)

Vol. 2, Issue 1, pp: (26-30), June-2024

Journal homepage: https://sanad.iau.ir/journal/josc

Research paper

Calculation of conductivity limit for fixed temperature limited plates using panel numerical method

Ahmad Reza Abedian[1] and Farhad Raiszadeh[2*]

1. Manager of Piping Group, Monenco Company, Tehran, Iran.
2. Energy Research Center, Shahrekord Branch, Islamic Azad University, Shahrekord, Iran.

Article Info

Article History:
Received: 2024/04/03
Revised: 2024/04/25
Accepted: 2024/05/20

DOI:

Keywords:
Conductivity limit, paneling method, constant temperature limited plates, one-side and two-sided active plates

**Corresponding Author's Email Address:reiszadehfarhad@gmail.com*

Abstract

The dimensionless Nusselt number is one of the important parameters in convection heat transfer, which can be considered as a measure of the heat transferred from the body through convection. In heat transfer, the Nusselt number indicates the rate of displacement heat transfer to conductive heat transfer. When heat transfer occurs only through conduction, it is called conduction limit, and its calculation is particularly important in the industry. This research uses a panel numerical method to calculate the conductivity limit for two-sided and one-side active plates for square, circular, and triangular cases. Finally, by comparing all the results, the performance of the panel numerical method for calculating the conductivity limit was confirmed.

1. Introduction

In the case where convection heat dissipation is considered, the Nusselt number is expressed as a function of the dimensionless numbers of Reynolds, Groshof and Prandtl, as well as the aspect ratio and the characteristic length of the object [1]:

$$Nu_L = f(\mathrm{Re}_L, Gr_L, \mathrm{Pr}, AR, L) \quad (1)$$

If the Reynolds number and Grashof tend to zero, the heat transfer from the body to the surrounding environment takes place only through conduction:

$$Nu_L = f(AR, L) \quad (2)$$

In this case, the value of the Nusselt number is called the conduction limit. Two methods can be used to calculate the conductivity limit of an isothermal object. In the first method, the temperature distribution around the object is calculated according to the solution of the Laplace equation; Then, the conductivity limit value is obtained by obtaining the temperature gradient on the boundary of the object. In the second method, the amount of heat transferred from the boundary of the object can be calculated without referring to the temperature distribution around it, and according to this value, the temperature of the boundary of the object and the temperature of the surrounding environment, the conductivity limit value can be obtained [2].

According to the definition, the amount of heat transferred through conduction with the surrounding environment is as follows:

$$Q = KS(T_o - T_\infty) \quad (3)$$

In the above relation, S is called the shape coefficient of conduction. According to the definition, the value of the conduction limit will be as follows:

$$S^*_{\sqrt{A}} = \frac{S}{\sqrt{A}} = \frac{Q}{K(T_o - T_\infty)\sqrt{A}} \quad (4)$$

In fact, the panel method is used to obtain Q in the above relationship.

2. The general steps of doing the work

For a better understanding of the method, in this section, the work process is divided as follows [3]:

- Object boundary paneling:

The advantage of using the panel numerical method compared to other numerical methods,

such as finite difference, is that, instead of gridding the whole object, only the boundary of the object is gridded. Also, in the panel method, because it works on the boundary of the object, the panels are linear or surface (depending on whether the problem is two-dimensional or three-dimensional). Therefore, to calculate the conduction limit using the panel method, the boundary of the object is modeled with several panels, and according to the geometric coordinates of each panel, a control point is assigned to it to perform calculations [4].

- Picking thermal springs:

In this part, according to the panelization of the boundary of the body, a heat source is placed on each panel, the distribution of each of which is fixed on its panel. Therefore, with the content that will be explained in the following parts, the method of calculating the power of the springs is investigated, from which the amount of transferred heat can be obtained.

- Calculation of temperature distribution around a point source:

It was mentioned in the previous part that the power of thermal springs on a panel is a constant value. Therefore, a panel's differential element can be considered a point source whose temperature distribution around it satisfies the Laplace equation. In this way, solving this differential equation around a point source is be easy.

- Calculation of the thermal effect of the control points due to a thermal spring:

The temperature increase of the desired control point can be calculated by calculating the thermal effect of an element on a control point and then by integrating this value on the panel.

- Calculate the temperature of the control points:

To calculate the temperature of control points, by using the law of sum of effects, the temperature of the desired control point can be obtained by summing the effects of all panels on a control point.

- Calculation of transferred heat:

In order to calculate the heat transferred from the surface of the object, the power of the thermal springs on each panel should be calculated. According to the boundary conditions of constant temperature, a system of linear equations appears; this system of linear equations can be solved with the help of the Gauss elimination method, and the power of the panels can be calculated. By calculating the heat transferred from the boundary of the object using the law of conservation of energy for the steady state, it can be said that the heat transferred from the surface of the object is equal to the algebraic sum of the heat exchanged from the boundary of the panels.

- Calculation of conductivity limit:

Having the heat transferred from the boundary of the body and knowing the temperature of the surface and the environment, the value of the conductivity limit can be calculated according to equation 4.

3. Mathematical steps of the work method

The surface of the object must first be modeled with surface panels to calculate the conduction limit of three-dimensional objects (for example, a cube or a flat plate). Then, a control point i is added to each of these panels, which is considered in the middle of them, and a boundary point j is assigned, which represents the number of the panel. Now, if the number of panels on the surface of the object is considered to be equal to m and the area of each of them is equal to s_j, and the power per unit of the panel surface is equal to q_j/k, the temperature distribution for a differential element ds_j from panel j, according to Laplace's equation, becomes as follows [1,4]:

$$\nabla^2 \Psi_j = 0 \tag{5}$$

In this regard, Ψj is defined as follows:

$$\Psi_j = T - T_\infty \tag{6}$$

By solving the above equation, the thermal effect of the differential element ds_j, at the control point i, is equal to:

$$dT_{ij} = \frac{q_j}{4\pi k}\frac{ds_j}{R_{ij}} \tag{7}$$

By integrating the above equation on the surface s_j, it is possible to calculate the increase in the temperature of the control point i due to the effect of panel j:

$$\Delta T_{ij} = C_{ij} q_j \tag{8}$$

which in the above relation C_{ij} is equal to:

$$C_{ij} = \frac{1}{4\pi k}\iint_{s_j}\frac{ds_j}{R_{ij}} \tag{9}$$

According to the law of sum of effects, in order to calculate the temperature of the control point i, the sum of the effects of all panels should be obtained as follows:

$$\sum_{j=1}^{m} C_{ij} q_j = \sum_{j=1}^{m} \Delta T_{ij} \tag{10}$$

By applying the boundary condition of constant temperature $T_i=T_0$, the sum of the effects of the panels at a control point is equal to $T_0 - T_\infty$, so the above relationship becomes the following relationship:

$$\sum_{j=1}^{m} C_{ij} q_j = T_o - T_\infty \tag{11}$$

With the values of T_0 and C_{ij} known to calculate the values of q_j, if the above relation is used for the

values i=1,2,...,m, a system of linear equations will appear in the form of the following relation:

$$\begin{aligned} &C_{11}q_1 + C_{12}q_2 + ... + C_{1m}q_m = T_o - T_\infty \\ &\vdots \\ &\vdots \\ &C_{m1}q_1 + C_{m2}q_2 + ... + C_{m1m}q_m = T_o - T_\infty \end{aligned} \quad (12)$$

The above equation is shown in matrix form as follows [5]:

$$[C_{ij}][q_j] = [B_i] \quad (13)$$

In the above equation, i=1,...,m and j=1,...,m and Bi = T_0 - T_∞.

As was said before, to calculate the values of q_j, the system of linear equations above should be solved using the Gauss elimination method.

With the power of the panels, the heat transferred from the boundary of the body according to the law of conservation of energy in a steady state becomes as follows:

$$Q = \sum_{j=1}^{m} q_j s_j \quad (14)$$

It is worth mentioning that with Q and (T0 - T∞), the value of the shape coefficient of conduction can be calculated from the following equation:

$$S = \frac{Q}{K(T_o - T_\infty)} \quad (15)$$

Obviously, using the above relation, the conduction limit becomes dimensionless as follows:

$$S^{\bullet}_{\sqrt{A}} = Nu_{\sqrt{A}} = \frac{S}{\sqrt{A}} \quad (16)$$

Considering that obtaining C_{ij} is of special importance for calculating the conductivity limit, the method of its calculation will be explained in the next part.

3.1. Calculation of the coefficient C_{ij}

For three-dimensional objects, the C_{ij} coefficient (according to Figure 1) is shown as follows [6]:

$$C_{ij} = \frac{1}{4\pi k} \iint_{s_j} \frac{ds_j}{R_{ij}} \quad (17)$$

where the integral can be calculated as follows:

$$\iint_{s_j} \frac{1}{R_{ij}} ds_j = \iint \frac{dxdy}{\sqrt{(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2}} \quad (18)$$

The calculation of this integral has been done analytically for panels that are in one plane, as shown in the figure 2; the coordinate axes are transferred so that the z-axis is always perpendicular to the plane of the panel. With this, the above integral will be in the following form:

$$\iint_{s_j} \frac{dxdy}{\sqrt{(x-x_i)^2 + (y-y_i)^2}} =$$

$$\left[\begin{aligned} &(x-x_i)\ln\left[(y-y_i) + \sqrt{(x-x_i)^2 + (y-y_i)^2}\right] \\ &+ (y-y_i)\ln\left[(x-x_i) + \sqrt{(x-x_i)^2 + (y-y_i)^2}\right] \end{aligned} \right]_{x_j - \frac{\omega}{2}}^{x_j + \frac{\omega}{2}} \Bigg]_{y_j - \frac{\gamma}{2}}^{y_j + \frac{\gamma}{2}} \quad (19)$$

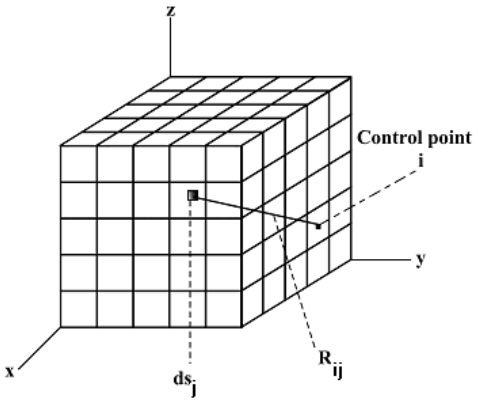

Figure 1- Paneling of a three-dimensional object

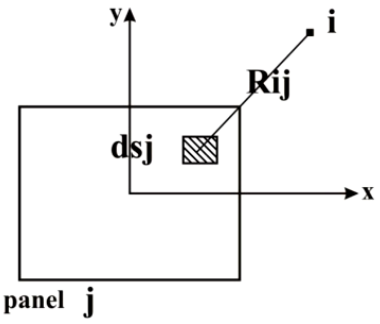

Figure 2- How to place the coordinate axes in a plane

In the relation above, ω is the length, and γ is the width of panel j. By calculating this value, the C_{ij} coefficient is obtained [7,8].

For panels not on the same plane, panel j is divided into m parts and the distance of each part to control point i is calculated using geometric relations. Finally, the coefficient C_{ij} is approximated with a sum in the following form:

$$C_{ij} = \frac{1}{4\pi k} \sum_{k=1}^{m} \frac{\Delta s_k}{R_{ik}} \quad (20)$$

To calculate the coefficient Cij, the following points should be taken into account [9]:

1- To calculate the effect of a panel on itself, it must be calculated analytically to eliminate the effect of individual points.

2- The length to width ratio of the panels should be as close as possible to a square panel.

3- The average distance value can be used to calculate Cij.

The next part will discuss the results obtained using the above method for the pages examined in this research.

4. Result and Discussion

- Two-sided active square plates

To calculate the conductivity limit of square plates like the one shown in Figure 3, we panelize the surface of the plate [10].

As shown in figure 3, the panels are selected as square because the plate is square and according to the point mentioned in the previous section. Because the screen is active on both sides, the back is also paneled similarly.

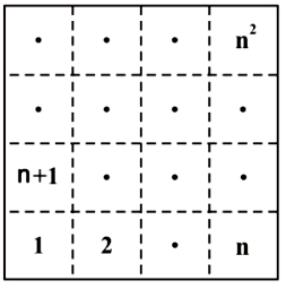

Figure 3- How to panel the square plates of the figure

By using the method explained in the previous section, the computer program related to the page of the two active sides is written, the results of which can be seen in figure 4.

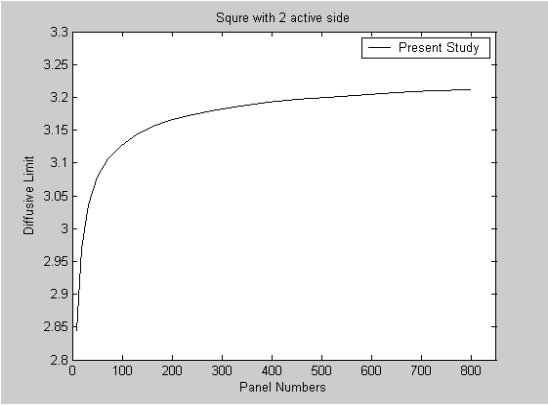

Figure 4 - The results obtained for the square plate with two active sides

As shown in the diagram above, the conduction limit tends to constant value with the increase in the number of panels. In the diagram above, it can be seen that the value of this number is equal to 3.2118.

Using a numerical method, Yovanovich [11,12] obtained the conductivity limit value for a square plate with two active sides with a small thickness. The value obtained using his method is 3.205.

The error between the values obtained from these two methods is around 0.2%, which is small.

- One-side active square plates

The working method in this part is the same as the previous part, with the difference that only one side of the square surface is paneled here.

With the increase in the number of panels, the conduction limit tends to be constant. This number equals 3.5422 for the square plates of one active side.

- Two-sided active circular plates

To calculate the conduction limit of circular plates like the one shown in figure 5, we panelize the surface of the plate.

In this part, the working method is the same as explained in the previous sections. The difference is that here, because the panels are not rectangular, it is not possible to use equation 19 for the panels on one page. Instead, equation 20 is used to examine the effect of panels on one page.

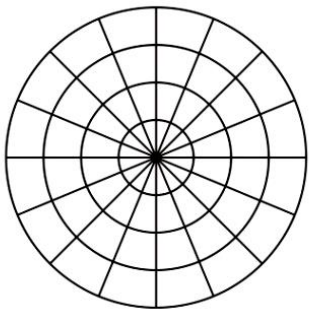

Figure 5- How to panelize the circular plates of the figure

The limit value obtained using this method tends to be 3.2053 by increasing the number of panels.

- One-side active circular plates

Here, the conduction limit tends to the number 3.5330 with increased panels.

- Two-sided active triangular plates

To calculate the conduction limit of triangular plates like the one shown in Figure 6, we panelize the surface of the plate.

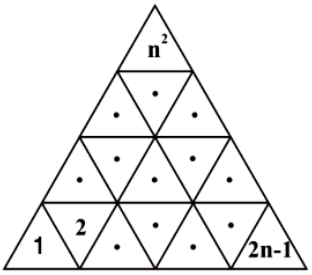

Figure 6- How to panel the triangular plates of the figure

As explained for the circular plates, relation 19 cannot be used and relation 20 must be used.

the conductivity limit tends to 3.3305 with the increase of panels.

- One-side active triangular plates

Here, the conductivity limit tends to 3.71 with the increase of panels.

5. Conclusion

As it can be seen, the values obtained for triangular plates are higher than those for the other two cases, but their difference is not great. Therefore, one value can be used as a conduction limit for all plates (by accepting a small error value). In addition, this difference is so small that it does not have much effect on the calculation of the Nusselt number.

References

[1] Liburdy J. Intermediate Fluid Mechanics. Oregon State University; 2020.

[2] Incropera FP, DeWitt DP, Bergman TL, Lavine AS. Fundamentals of heat and mass transfer. New York: Wiley; 1996 Feb 16.

[3] Bigdely, M., "Conduction Limit Calculation Using Panel Method." M.S. Thesis, Shiraz, Shiraz, Iran, 1998.

[4] Abedian, A.R., "Experimental Study of Convective Heat Transfer from Isothermal Finite Plates: Mass Transfer Analogy Using the Electro-Chemical Measurement." M.S. Thesis, Shiraz, Shiraz, Iran, 2006.

[5] Zhao W, Wasala S, Persoons T. Development and Assessment of an Inviscid Source Vortex Panel Method for Low-Reynolds Number Airfoil Cascades. InAIAA SCITECH 2024 Forum 2024 (p. 2507).

[6] Anderson D, Tannehill JC, Pletcher RH, Munipalli R, Shankar V. Computational fluid mechanics and heat transfer. CRC press; 2020 Dec 17.

[7] Abu-Hamdeh NH, Khorasani S, Oztop HF, Alnefaie KA. Numerical analysis on heat transfer of a pyramid-shaped photovoltaic panel. Journal of Thermal Analysis and Calorimetry. 2021:1-2.

[8] Ramachandran P, Rajan SC, Ramakrishna M. A fast, two-dimensional panel method. SIAM Journal on Scientific Computing. 2003;24(6):1864-78.

[9] Poncet P. Analysis of direct three-dimensional parabolic panel methods. SIAM journal on numerical analysis. 2007;45(6):2259-97.

[10] Agne MT, Hanus R, Snyder GJ. Minimum thermal conductivity in the context of diffuson-mediated thermal transport. Energy & Environmental Science. 2018;11(3):609-16.

[11] Yovanovich, M.M., "New Nusselt and Sherwood Numbers for Arbitrary Isopotential Bodies at Near Zero Peclet and Rayleigh Numbers," AIAA-87-1643, presented at AIAA 22nd Thermophysics Conference, Honolulu, Hawaii, June 8-10,1987.

[12] Yovanovich, M.M., and Jafarpur, K., "Bounds on Laminar Natural Convection from Isothermal disks and Finite Plates of Arbitrary Shape for all Orientations and Prandtl Numbers," ASME Fundamentals of Natural Convection, Vol. 264, 1993.

Journal of Optimization of Soft Computing (JOSC)

Vol. 2, Issue 1, pp: (31-36), June-2024

Journal homepage: https://sanad.iau.ir/journal/josc

Research paper

A Comprehensive Review on Service Function Chaining in Network Environments

Pouya Khosravian Dehkordi[1*]

1. Department of Computer Engineering, Faculty of Engineering, Shahrekord Branch, Islamic Azad University, Shahrekord, Iran.

Article Info

Article History:
Received: 2024/03/17
Revised: 2024/06/05
Accepted: 2024/06/08

DOI:

Keywords:
Service Function Chaining, Networking, Virtualization, Orchestration, Network Function Virtualization

**Corresponding Author's Email Address: Khosravyan@gmail.com*

Abstract

Service Function Chaining (SFC) has emerged as a critical technology in networking environments to provide efficient and flexible service delivery. This paper provides a comprehensive review of the current state of research and development in the field of SFC, covering key concepts, architectures, challenges, and trends. We first introduce the concept of SFC and its importance in modern networking environments. We then discuss different SFC architectures and their advantages and drawbacks. Furthermore, we analyze the challenges and opportunities in the deployment of SFC in real-world scenarios. Finally, we discuss the emerging trends and future directions in the field of SFC. This review aims to provide researchers and practitioners with a deeper understanding of the current landscape of SFC and guide future research efforts in this area.

1. Introduction

Networking environments are becoming increasingly complex and dynamic, with the proliferation of diverse services and applications. Service Function Chaining (SFC)[3,7,10] has emerged as a key technology to enable the efficient delivery of services in such environments by defining the sequence of service functions that data packets must traverse. By chaining service functions together, SFC enables the creation of customized service paths tailored to specific requirements, leading to improved performance, flexibility, and scalability.

In recent years, there has been growing interest in the research and development of SFC, driven by the increasing demand for network virtualization, automation, and orchestration. This paper provides a comprehensive review of the current state of research and development in the field of SFC, covering key concepts, architectures, challenges, and trends. The rest of the paper is organized as follows: Section 2 introduces the concept of SFC and its importance in networking environments. Section 3 discusses different SFC architectures and their advantages and drawbacks. Section 4 analyzes the challenges and opportunities in the deployment of SFC in real-world scenarios. Finally, Section 5 discusses the emerging trends and future directions in the field of SFC.

2. Service Function Chaining: Concepts and Importance

Service Function Chaining (SFC) is a networking technology that enables the sequential traversal of service functions by data packets to provide end-to-end services. A service function is a network function that performs specific tasks on data packets, such as firewalling [16, 19], load balancing, and encryption. By chaining service functions together, SFC allows for the creation of

customized service paths tailored to specific requirements.

The importance of SFC in networking environments stems from its ability to improve service delivery, performance, and flexibility. Traditional networking architectures rely on static service paths that are predefined and inflexible, leading to suboptimal service delivery and resource utilization. In contrast, SFC enables dynamic service chaining based on real-time requirements, allowing for on-demand allocation of service functions and resources. This flexibility is critical in modern networking environments characterized by diverse services, applications, and traffic patterns [11].

Moreover, SFC enables the decoupling of service functions from the underlying network infrastructure, leading to improved scalability, manageability, and cost-effectiveness. By virtualizing service functions and orchestrating their deployment, SFC minimizes the reliance on dedicated hardware appliances and facilitates the introduction of new services and functionalities. This decoupling also enables service providers to offer customized service chains to meet the specific needs of their customers, leading to enhanced service differentiation and customer satisfaction.

Overall, SFC plays a crucial role in the evolution of networking architectures towards virtualization, automation, and orchestration. By defining service chains dynamically based on real-time requirements, SFC enables more efficient and flexible service delivery, improving the overall performance and user experience in networking environments.

3. SFC Architectures: Advantages and Drawbacks

Several SFC architectures have been proposed in the literature to implement service function [17] chaining in networking environments. These architectures vary in terms of their design principles, deployment models, and scalability. In this section, we discuss some of the most common SFC architectures, highlighting their advantages and drawbacks.

3.1. Overlay SFC Architecture

Overlay SFC architectures involve the deployment of virtualized service functions on top of the existing network infrastructure. In this architecture, service function instances are abstracted from the underlying physical network, allowing for greater flexibility and scalability. Overlay SFC architectures leverage virtualization technologies, such as network function virtualization (NFV)[1,6,9] and software-defined networking (SDN)[2,4,8,12,13,14,15], to instantiate and orchestrate service functions dynamically.

One of the key advantages of overlay SFC architectures is their capability to decouple service functions from the underlying network infrastructure, leading to improved manageability and agility. By virtualizing service functions, overlay SFC architectures enable on-demand allocation and deployment of service functions, eliminating the need for dedicated hardware appliances. This virtualization also facilitates the chaining of heterogeneous service functions across different administrative domains, enhancing service flexibility and interoperability.

However, overlay SFC architectures also have some drawbacks, such as increased overhead and complexity. The additional layer of abstraction introduced by virtualized service functions can lead to performance degradation and resource inefficiency. Moreover, the heterogeneous nature of service functions and network environments can pose challenges in terms of interoperability, security, and quality of service. Despite these drawbacks, overlay SFC architectures remain a popular choice for implementing service function chaining in networking environments [18] due to their flexibility and scalability.

3.2. Underlay SFC Architecture

Underlay SFC architectures involve the integration of service functions into the underlying network infrastructure, leveraging dedicated hardware appliances or purpose-built devices. In this architecture, service functions are deployed as network devices or middleboxes, enabling the creation of service chains at the network level [20]. Underlay SFC architectures are commonly used in traditional networking environments where service functions are tightly coupled with network devices.

One of the key advantages of underlay SFC architectures is their efficiency and performance. By integrating service functions into the network infrastructure, underlay SFC architectures reduce the overhead associated with virtualization and orchestration, leading to improved throughput and latency. Moreover, the tight coupling of service functions with network devices enables more granular control and visibility, facilitating troubleshooting and monitoring.

However, underlay SFC architectures also have some drawbacks, such as limited flexibility and scalability. The static nature of service functions

deployed as network devices can restrict the dynamic allocation and chaining of service functions, leading to suboptimal resource utilization and service delivery. Moreover, the dependence on dedicated hardware appliances can increase the cost and complexity of managing service functions.

Despite these drawbacks, underlay SFC architectures remain a viable option for implementing service function chaining in networking environments where performance and efficiency are critical. By leveraging purpose-built devices and network integration, underlay SFC architectures provide a more streamlined and deterministic approach to service delivery, ensuring high availability and reliability.

4. Challenges and Opportunities in SFC Deployment

The deployment of Service Function Chaining (SFC)[24,25] in real-world networking environments poses several challenges and opportunities. In this section, we analyze some of the key challenges and opportunities associated with the deployment of SFC and discuss potential solutions to address them.

4.1. Scalability

One of the primary challenges in SFC deployment is scalability, especially in large-scale networking environments with diverse services and traffic patterns. As the number of service functions and service chains increases, the complexity of managing and orchestrating them also grows, leading to scalability issues. Moreover, the dynamic nature of service chaining and the need for real-time adaptation further exacerbate scalability challenges.

To address scalability challenges in SFC deployment, researchers and practitioners have proposed several solutions, such as hierarchical service function chaining, load balancing, and parallel processing. Hierarchical service function chaining involves organizing service functions into hierarchical levels based on their functionalities and dependencies, enabling more efficient orchestration and management. Load balancing techniques distribute traffic across multiple service function instances to ensure optimal resource utilization and performance. Parallel processing techniques leverage the parallelism of modern hardware architectures to accelerate the processing of service functions and improve scalability.

4.2. Interoperability

Another challenge in SFC deployment is interoperability, especially in heterogeneous networking environments with diverse service functions and protocols. The lack of standardization and compatibility among service functions and network devices can hinder the seamless integration of service chains and the interoperability of different administrative domains. Moreover, the dynamic nature of service chaining and the need for real-time adaptation further complicate interoperability issues.

To address interoperability challenges in SFC deployment, researchers and practitioners have proposed several solutions, such as standardization, protocol mediation, and abstraction layers. Standardization efforts, such as the work of the Internet Engineering Task Force (IETF) and the European Telecommunications Standards Institute (ETSI), aim to define common protocols and interfaces for service function chaining, enabling interoperability among different service functions and network devices. Protocol mediation techniques translate and adapt protocols between different service functions and network devices to ensure seamless communication and interoperability. Abstraction layers provide a common interface for managing and orchestrating service functions, shielding the underlying complexity and heterogeneity.

4.3. Security

Security is a critical concern in SFC deployment, as service function chaining involves the processing of sensitive data and the enforcement of security policies. The dynamic nature of service chaining and the integration of virtualized service functions can introduce new security vulnerabilities and risks, such as data breaches, unauthorized access, and service disruptions. Moreover, the complexity of managing and orchestrating service functions across different administrative domains can further complicate security issues.

To address security challenges in SFC deployment, researchers and practitioners have proposed several solutions, such as encryption, authentication, and access control. Encryption techniques protect sensitive data by encoding it before transmission and decoding it upon reception, ensuring confidentiality and integrity. Authentication mechanisms verify the identities of users and service functions to prevent unauthorized access and tampering. Access control policies define the permissions and restrictions for accessing service functions and

resources, ensuring compliance with security policies and regulations.

5. Emerging Trends and Future Directions in SFC

Service Function Chaining (SFC)[21,22,23] is a rapidly evolving technology with several emerging trends and future directions that are shaping the future of networking environments. In this section, we discuss some of the key emerging trends and future directions in SFC and their potential impact on the networking landscape.

5.1. Multi-Domain Service Function Chaining

Multi-domain service function chaining involves the chaining of service functions across multiple administrative domains and network boundaries. This trend is driven by the increasing demand for end-to-end services that span different service providers and network operators. Multi-domain service function chaining enables seamless service delivery and interoperability across diverse network environments, enhancing service flexibility and user experience.

To enable multi-domain service function chaining, researchers and practitioners are exploring new techniques and protocols for inter-domain coordination, policy enforcement, and service negotiation. Inter-domain coordination mechanisms facilitate the exchange of service chaining information and policies between different administrative domains, ensuring consistent service delivery and enforcement of security, quality of service, and service level agreements. Policy enforcement mechanisms enforce access control policies and service agreements across multiple domains, ensuring compliance with regulatory requirements and contractual obligations. Service negotiation mechanisms enable service providers and network operators to negotiate and establish service chains dynamically based on real-time requirements, enabling more personalized and flexible service delivery.

5.2. Service Function Virtualization

Service function virtualization involves the virtualization of service functions to decouple them from the underlying network infrastructure and enable dynamic instantiation and orchestration. This trend is driven by the increasing demand for network agility, scalability, and cost-effectiveness. Service function virtualization enables service providers and network operators to deploy and manage service functions more efficiently, leading to improved resource utilization and service delivery.

To enable service function virtualization, researchers and practitioners are exploring new virtualization techniques and architectures for service function instantiation, management, and orchestration. Virtualization techniques, such as containerization and microservices, enable the efficient deployment and scaling of service functions in virtualized environments. Virtualization architectures, such as cloud-native and edge computing, provide the infrastructure and platforms for hosting and managing virtualized service functions, enabling on-demand allocation and chaining. Orchestration frameworks, such as Kubernetes and OpenStack, automate the deployment and lifecycle management of virtualized service functions, ensuring efficient resource utilization and service availability.

5.3. Machine Learning and AI

Machine learning and artificial intelligence (AI) are playing an increasingly important role in shaping the future of Service Function Chaining (SFC) by enabling intelligent service orchestration, optimization, and automation. Machine learning algorithms and AI models can analyze and learn from network data, traffic patterns, and service requirements to make informed decisions on service chaining and resource allocation. By leveraging machine learning and AI, service providers and network operators can optimize service delivery, improve performance, and reduce operational costs.

To enable machine learning and AI in SFC, researchers and practitioners are developing new algorithms and models for service orchestration, optimization, and automation. Machine learning algorithms, such as reinforcement learning and deep learning, can analyze and predict service requirements, traffic patterns, and system performance to optimize service chaining and resource allocation. AI models, such as neural networks and decision trees, can automate the decision-making process for service orchestration and management, enabling more efficient and adaptive service delivery.

6. Conclusion

Service Function Chaining (SFC) has emerged as a critical technology in networking environments to provide efficient and flexible service delivery. This paper has provided a comprehensive review of the current state of research and development in the field of SFC, covering key concepts,

architectures, challenges, and trends. We have discussed different SFC architectures, their advantages and drawbacks, as well as the challenges and opportunities in the deployment of SFC in real-world scenarios. Furthermore, we have analyzed the emerging trends and future directions in the field of SFC, such as multi-domain service function chaining, service function virtualization, and machine learning and AI.

Overall, SFC plays a crucial role in the evolution of networking architectures towards virtualization, automation, and orchestration. By defining service chains dynamically based on real-time requirements, SFC enables more efficient and flexible service delivery, improving the overall performance and user experience in networking environments. The emerging trends and future directions in SFC are shaping the future of networking environments, leading to more intelligent, scalable, and cost-effective service delivery.

References

1. ETSI. "Network Functions Virtualization (NFV); Architectural Framework (ETSI GS NFV 002 V1.2.1)." 2013.
2. Kreutz, D., Ramos, F. M. V., Veríssimo, P. E., Rothenberg, C. E., Azodolmolky, S., and Uhlig, S. "Software-Defined Networking: A Comprehensive Survey." Proceedings of the IEEE, vol. 103, no. 1, 2015, pp. 14-76.
3. IETF. "Service Function Chaining (SFC) Architecture (RFC 7665)." 2015.
4. Jain, S., Kumar, A., Mandal, S., Ong, J., Poutievski, L., Singh, A., Vahdat, A., Bahl, P., and S. S. "B4: Experience with a Globally-deployed Software Defined WAN." ACM SIGCOMM, 2013.
5. Chowdhury, N. M. M. K., Boutaba, R., Aib, I., and Ayoubi, S. "VHCP: Virtual Home Carrier Gateway placement in the cloud." IEEE/IFIP Network Operations and Management Symposium (NOMS), 2014.
6. Boyd, P., Lancaster, C., and Guo, Y. "Software-Defined Networking (SDN) and Network Functions Virtualization (NFV) Integration in a Real-time Decision-making Platform." IEEE Transactions on Network and Service Management, vol. 17, no. 4, 2020, pp. 2029-2042.
7. Cisco. "Service Function Chaining: Service Chaining in Provider Networks." 2019.
8. Lantz, B., Heller, B., McKeown, N., and Rexford, J. "A network in a laptop: rapid prototyping for software-defined networks." ACM SIGCOMM Computer Communication Review, vol. 43, no. 3, 2013, pp. 63-74.
9. Barr, J., Baudoin, G., Bognar, A., Ciciliano, F., Heijenk, G., Pascucci, F., and Skoldstrom, P. "Description and Definition of Network Functions Virtualisation (NFV) Management and Orchestration." ETSI GS NFV-MAN 001 V1.1.1, 2019.
10. Nakagawa, E., Tomida, T., and Banerjee, A. N. "A Service Function Chaining Framework for Software-Defined Networks." IEEE Communications Magazine, vol. 56, no. 8, 2018, pp. 104-109.
11. Dhamdhere, A., Dovrolis, C., Feamster, N., Huffaker, B., and Gao, L. "A Longitudinal Study of Cloud Network Traffic Characteristics." ACM Internet Measurement Conference, 2014.
12. Jain, S., Kumar, A., Mandal, S., Ong, J., Poutievski, L., Singh, A., Subramanya, V. S., Vahdat, A., Jon Feldman, M., and Zhaogang Wang. "B4: Experience with a Globally-deployed Software Defined WAN." ACM SIGCOMM, 2013.
13. Kreutz, D., Ramos, F. M., Esteves Verissimo, P., Rothenberg, C. E., Azodolmolky, S., and Uhlig, S. "Software-Defined Networking: A Comprehensive Survey." Proceedings of the IEEE, vol. 103, no. 1, 2015, pp. 14-76.
14. OASIS. "Topology and Orchestration Specification for Cloud Applications Version 1.0." 2016.
15. Surendran, A., Sibi, R. T., and Yavuz, A. A. "Software Defined Networking (SDN): An Architectural Framework for Virtualized Resource Management." IEEE Communications Magazine, vol. 52, no. 4, 2014, pp. 168-175.
16. Check Point Software Technologies. "How Check Point's FireWall-1 implements network address translation." White Paper, 1994.
17. Cisco. "Service Function Chaining Design Guide." 2020.
18. Microsoft. "Azure Networking: What's new." Blog post, 2021.
19. Radhakrishnan, M., Cervino, J., and Chowdhury, K. R. "Firewall policy verification with Snort IDS." ACM SIGCOMM Computer Communication Review, vol. 37, no. 3, 2007, pp. 3-14.
20. Kang, D., Kim, D., and Jeon, S. "Network-based Attack Detection and Mitigation." IEEE Transactions on Network and Service Management, vol. 10, no. 4, 2019, pp. 611-623.
21. Asaeda H, Matsuzono K, Hayamizu Y, HLAING HH, Ooka A. A Survey of Information-Centric Networking: The Quest for Innovation. IEICE Transactions on Communications. 2024 Jan 1;107(1):139-53.
22. Yukun S, Bo L, Juniin L, Haonan H, Xing Z, Jing P, Wenbo W. Computing power network: A survey. China Communications. 2024 Apr 9.
23. Karapantelakis A, Alizadeh P, Alabassi A, Dey K, Nikou A. Generative ai in mobile networks: a survey. Annals of Telecommunications. 2024 Feb;79(1):15-33.
24. Wang S, Yang L. A Survey of Service Function Chain Orchestration Based on Neural Network. In2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall) 2023 Oct 10 (pp. 1-5). IEEE.
25. Hu Y, Guo Y. Blockchain-Enabled Service Function Chain in 6G Networks: A Survey. In2023 IEEE International Conference on Communications Workshops (ICC Workshops) 2023 May 28 (pp. 446-451). IEEE.

Pouya Khosravian Dehkordi received the B.Eng. degree from Islamic Azad University, Najafabad branch, Iran, in 2005 and the M.S. degree from Islamic Azad University, Arak branch, Iran, in 2008. Since 2009, he is a faculty member of Islamic Azad University, Shahrekord branch, Iran. Also in 2020, he received the Ph.D. from Islamic Azad University, Yazd branch, Iran. His Ph.D. thesis deals with Service Function Chaining. His current research interests include Software Defined Networks, Service Function Chaining, Natural Language Processing, and Automata Theory.

Journal of Optimization of Soft Computing (JOSC)

Vol. 2, Issue 1, pp: (37-43), June-2024

Journal homepage: https://sanad.iau.ir/journal/josc

Research paper

Comparison of Pre-Trained Models in Extractive Text Summarization of Mobile App User Reviews

Mehrdad Razavi Dehkordi [1], Hamid Rastegari [1,2*], Akbar Nabiollahi Najafabadi [1,2], Taghi Javdani Gandomani [3]

[1]. *Faculty of Computer Engineering, Najafabad Branch, Islamic Azad University, Najafabad, Iran*
[2]. *Big Data Research Center, Najafabad Branch, Islamic Azad University, Najafabad, Iran.*
[3]. *Department of Computer Science, Shahrekord University, Shahrekord, Iran.*

Article Info

Article History:
Received: 2024/03/11
Revised: -
Accepted: 2024/05/26

DOI:

Keywords:
Mobile applications, Summarization of User Reviews, Google Play Store Analysis, Pre-trained Model

* *Corresponding Author's Email Address: rastegari@iaun.ac.ir*

Abstract

Since the inception of mobile apps, user feedback has been extremely valuable to app developers as it contains users' feelings, bugs, and new requirements. Due to the large volume of reviews, summarizing them is very difficult and error-prone. So far, many works have been done in the field of extractive summarization of users' reviews; However, in most researches, old methods of machine learning or natural language processing have been used, or if a model has been trained for summarizing using transformers, it has not been determined whether this model is useful for summarizing the reviews of mobile users. No? In other words, the model for summarizing texts has been presented in a general-purpose form, and no investigation has been carried out for its use in special purpose summarization. In this article, first, 1000 reviews were randomly selected from the Kaggle database of user reviews, and then given to 4 pre-trained models bart_large_cnn, bart_large_xsum, mT5_multilingual_XLSum, and Falcon'sAI_ Text_Summrization for summarization, and the criteria Rouge1, Rouge2 and RoungL were calculated separately for each of the models and finally it was found that the pre-trained Falcon's AI model with a score of 0.6464 in the rouge1 criterion, a score of 0.6140 in the rouge2 criterion and a score of 0.6346 in rougeL The best model for summarizing users' reviews is the Play Store.

1. Introduction

According to the official statistics of IDC website[1], about 300.3 million smartphones were produced by manufacturers by the fourth quarter of 2022, 75.8% of which were smartphones with the Android operating system. The Android operating system has its own store, called Google Play Store, which includes all its produced apps by developers[2][1][2][3].

The apps available in the store are downloaded by many users and Google Play Store users could comment on the desired application. Studies have demonstrated the reviews made by users on apps

[1] IDC - Smartphone Market Share - Market Share. https://www.idc.com/promo/smartphone-marketshare

[2] Android Apps on Google Play. https://play.google.com/store/apps

contains important information, including bug reports, feature requests and user experience of working with the app[4][5][6]. Previous studies have revealed the reviews recorded by users could contribute to app development process and improve future app versions[7][8]. Moreover, reviews contain important information for app analysts, designers and developers[6][9][5]. Due to the high volume of reviews with important information, it was difficult to summarize them for handling by the development team manually, and as a result, having a tool to summarize and export the summarized reviews to the development team is very useful; Because having a tool or a model for summarizing reviews makes a summary list of requirements or bugs feed backed by users available to the development team and the development team does not waste much time reading each review and maintaining the software and providing More successful timely updates[10][11][12]. So far, many works have been done in summarizing the reviews of users of mobile applications, but in most of them, either natural language processing parameters have been used or machine learning methods have been used, which are old. Today, many pre-trained models are used. To summarize the reviews of users, using transformers are provided and all the models provided are general purpose and are not provided for a specific task. The purpose of this research is to compare and select the best pre-trained model in the extractive text summarization of user reviews of mobile applications in the Play Store. In this research, at first, 1000 reviews from the dataset including user reviews provided by Kaggle were randomly selected and given to 4 pre-trained models bart_large_cnn, bart_large_xsum, mT5_multilingual_XLSum and Falconsai. Finally, Rouge criteria have been measured for each model. The continuation of this article is organized as follows: in the second part, the concepts and works done in summarizing the reviews and their challenges are discussed, in the third part, the pre-trained models and their parameters are stated, in The fourth section discusses the dataset and evaluation criteria, the fifth section compares the 4 models presented in the summary, and finally, the sixth section provides conclusions and suggestions for future work.

2. Concepts, Literature Review and their Challenges

In this section, the concepts of text summarization are discussed first, and then the work done in the field of summarizing user reviews will be discussed, and finally, their challenges and problems will be discussed.

2.1 Concepts of text summarization

Text summarization was first introduced by Luhn in 1950 in the first IBM computers using the bag of words method[13]. In this method, the number of frequency of words that were used repeatedly in the text was counted, and based on that, a score was given to each sentence, and summarization was done based on this score. In the following, summarization methods were advanced by using linguistic parameters available in natural language processing. Then, new methods for converting sentences into vectors such as word2vec [14] and deep learning methods using LSTM architectures [15], RNN networks [16] and convolutional neural networks [17] were presented.
In general, there are 2 methods for summarizing texts:

A- Extractive summarization of reviews is done with the aim of identifying words and sentences and using them to create a summary of the text. In this method, the selection of words and sentences is based on their importance. This process includes three parts: separating the sentences and words, calculating the score and selecting the sentences and words with the highest score[18][19][20].

B- Abstractive summarization that has been developed and automated traditional methods. In this method, the key parts of the sentences and the main ideas of the sentence are processed using quoting. This method of summarizing includes the stages of analyzing sentences and quotations, which is done with two methods based on structure and based on meaning[21].

2.2 Work done in summarizing user Reviews

In this part, some of the works done in the field of summarizing reviews will be discussed according to the method used by them. Also, at the end, their challenges and problems will be discussed.

Table 1- Important works presented in the field of text summarization

Year/Reference	Main goal of Research	Challenges & Problems
2009/[22]	Investigating the problems in summarizing texts and providing a classification for summarizing methods	Lack of attention to methods based on extractive and abstractive text summarization, natural language processing, machine learning and deep learning
2014/[23]	Presenting a hybrid method based on extractive and abstractive summarization	In the described method, features based on natural language processing are not used
2014/[24]	Reviewing the work done from 2000 to 2013 and presenting a consolidated method based on statistics.	In this method, the cognitive features of language such as visualization have not been addressed, and its effect on summarization has not been measured
2016/[25]	Presenting two definitive methods for extractive and abstractive summarization of reviews	No testing has been done for the presented method
2017/[26]	A study based on automatic extraction of key words of texts and summarizing them	The method presented by them is not fully and clearly stated and the feature extraction part model is not stated
2017/[20]	Explain the advantages and disadvantages of topic-based, iteration count, and graph-based methods	The stated methods are not well explained.
2017/[27]	Processing related to extractive summarization methods is described in different languages	The exact idea about how to score features and how to extract them is not explained
2020/[28]	The method, processes, main structure, dataset and how to measure the efficiency of automatic summarization models are mentioned.	How to classify and extract features is not described in detail
2020/[29]	Summarizing a set of documents based on previous work	There is no explanation about the different methods

In Table 1, the important works done in the field of summarizing reviews by both extractive and abstractive methods are stated. The presented works have challenges and problems as follows:

- Summarization methods based on deep learning have not been addressed at all
- In some articles, a method for summarization is presented, but the presented method has not been tested with any dataset
- In some summarization methods, the proposed method is not described in full detail
- None of the presented methods are specific and all of them are general and introduced to summarize all the texts.
- The methods of summarizing texts using transformers have not been discussed.
- To summarize the reviews of users of mobile applications, no specific method has been stated

Considering the challenges and problems mentioned above, providing a method or searching for a high-performance method for summarizing the reviews of mobile application users is required.

RQ. Which of the pre-trained models based on extractive text summarization in terms of Rouge criteria is suitable for summarizing the reviews of mobile application users?

3. Pre-Trained Models in User Reviews Text Summarization

3.1 bart_large_cnn

It is a pre-trained model in English and fine-tuned with CNN newspaper news using 400 million parameters[30]. This model is available on the hunggingface.com website, which includes many pre-trained models for various tasks such as summarizing, categorizing, masking, sentiment analysis, searching for text keywords, etc. To use this model, it is sufficient to give the parameters max_length (maximum number of words of the input text) and min_length (minimum number of words of the input text) as input to the model along with the desired text[31].

3.2 bart_large_XSUM

The model was trained using 226 million BBC articles from 2010 to 2017 in the categories of politics, news, weather, sports, business, science, health, education and family, entertainment and arts.

3.3 mT5_multilingual_XLSum

This model is based on unsupervised learning method using different parameters for Amharic, Arabic, Azerbaijani, Bengali, Burmese, Chinese, English, French, Gujarati, Hausa, Hindi, Igbo, Indonesian, Japanese, Kirundi, Korean, Kyrgyz, Marathi, Nepali, Oromo, Pashto, Pidgin, Portuguese, Punjabi, Russian, Scottish, Serbian, Spanish, Thai, Turkish, Ukrainian, Uzbek, languages and etc.... It Performs tasks such as summarizing, translating, correcting words, language acceptance, etc[32].

3.4 Falcon's AI

This model is trained based on the original T5 model for the summarization task only, so that it can produce accurate and good results in extractive summarization. This model is trained to generate text summaries with higher efficiency than the base T5 model; In addition, this model is trained using a dataset based on summaries made by humans[33].

4. Experimental Design

In this section, the dataset used, the testing environment, the comparison criteria, and how the tests are performed are explained.

4.1 Test Environment

Python programming language version 3.10.11 and Visual Studio Code version 1.78.2 programming environment have been used to test the model. The reason for using this environment is the ease of Debug and compatibility with Microsoft Visual Studio.

To compare the model with other models, the computer of the Big Data Research Center located in the Islamic Azad University of Najafabad branch with an Intel Xeon E5-2650 v4 processor, 16 GB of DDR4 RAM, without a graphics card and Windows 10 was used.

4.2 Used Dataset

The database provided by Kaggle has been used to train the model. Tables 2 and 3 provide complete information about the dataset and its features.

Table 2 - Dataset used along with details

Provider	Number of Reviews	Number of Apps	Number of Categories
Kaggle	51000	10842	32

Table 3 - Features available for each application in the used dataset

Number	Feature
1	App Name
2	Category
3	Average app rating (0 to 5)
4	Number of Reviews
5	App Size in mb
6	Number of Installations
7	Free or paid
8	Price of app in case of not free
9	Age limit for using app
10	Date of last app update
11	Last Version of app

12	Minimum android version required for installing app
13	Reviews for app in text format
14	Reviews Label (Feature Request, Bugfix and Information Giving)

As can be seen in Table 2 and 3, the above dataset contains the play store reviews submitted by users for the application. From the mentioned database, 1000 reviews are randomly selected and given to 4 models for summarization.

4.3 Data Preprocessing

Before sending each review to the summarization models, we have performed text-related preprocessing operations such as removing special characters (e.g. #, * and ...), whitespace, and punctuation on the data; In addition, all the letters related to reviews have been converted to lower case. The reason for doing this is the ease of work for the review summarization system, which works on the basis of transformers [33] [34].

4.4 Rouge Evaluation criterion for pre-trained model evaluation

To answer the research question, our main goal is to compare pre-trained models to find the best model for extractive summarization of user reviews. For this purpose, the models are checked in terms of the F-Mesaure criterion with the Rouge evaluation criterion, which is specific for the evaluation of the summarization methods.
Rouge (Recall-Oriented Understudy for Gisting Evaluation) is a set of benchmarks and a software package specifically designed to evaluate machine summarization, but can also be used for machine translation. These criteria compare a summary or machine translation with reference summaries or translations (of high quality and produced by humans). The rouge criterion itself has subsets that are defined in different articles based on the number n of common tuples between sentences. The main and the summarized sentence are calculated. This means that the rouge1 criterion calculates the number of common 1s between two sentences, the rouge2 criterion calculates the number of 2s in common, and the rougeL criterion calculates the L number of common tuples between two sentences. Then, based on the degree of similarity, precision, recall and F-Measure are calculated[34] and finally, based on the F-Measure parameter, it will be decided whether the presented method is suitable for summarization or not.

4.5 How to Perform the Tests

RQ. To find the best model in summarizing user reviews, first, 1000 reviews are randomly selected from the mentioned dataset and then given to each of the models for summarization separately. Each test is repeated 10 times and summarized reviews are kept at each stage. Then, through the libraries available in the Python software, for each model, 1000 original texts along with 1000 summarized texts are given to Python, and then each review is compared with its summarized review separately, and the rouge measure is compared with F-Measure. It is calculated for that. In the following, the amount of this parameter is recorded, the next review along with the summary is prepared for processing. Finally, the average parameters of rouge1, rouge2 and rougeL are calculated for each review

5.Experiment Results

RQ. In this part, to answer the research question, the results of the tests related to the pre-trained models in summarizing reviews are calculated according to the rouge criterion.

Table 4 - Comparison of pre-trained models in summarizing 1000 reviews

Model	rouge1 Average	rouge2 Average	rougeL Average
bart_large_cnn	0.3801	0.3517	0.3521
bart_large_XSUM	0.1736	0.0518	0.1434
mT5_multilingual_XLSum	0.1976	0.0602	0.1695
Falcon'sAI	0.6464	0.6140	0.6346

As can be seen in Table 4, Falcon's AI model has a better result than other models in summarizing reviews.

- It should be noted that in none of the articles', pre-trained models have not been compared for the task of summarizing the reviews of mobile application users.
- On the other hand, because the Falcon's AI model has been trained with different datasets of reviews, texts, news, etc. for summarizing, it has been able to get better results in summarizing users' reviews.

6.Conclusion & Future Work

Due to the fact that the number of reviews submitted for applications is very large, summarizing them by the development team is a difficult and time-consuming task. If there is a method or a tool to summarize the reviews, it can save the time of the development team and help to implement new features in the application, fix their bugs and make the application successful. There are many pre-trained models for summarizing texts, but none of them have been specifically adjusted for summarizing reviews. In this article, 4 pre-trained models were compared in the extractive summarization of reviews according to rouge parameter in summarizing 1000 reviews from Kaggle dataset. Finally, it was found that the Falcon's AI method is a suitable method for the extractive summarization of reviews. By using the pre-trained model in summarizing reviews, the development team will easily have a summary list of reviews after categorizing the reviews, and will not waste time reading long reviews from the development team. Falcon's AI model was able to obtain a score of 0.6346 in the rougeL parameter due to the use of many parameters and precise adjustment using texts in different categories. This means that the degree of similarity of the summarized text with the original text is appropriate.

In the future, more pre-trained models can be examined and compared in summarizing reviews, if there is a dataset, a model can be presented for abstract summarization of users' reviews also in the field of Persian language, pre-trained models.

References

[1] M. R. Dehkordi, H. Seifzadeh, G. Beydoun, and M. H. Nadimi-Shahraki, "Success prediction of android applications in a novel repository using neural networks," *Complex Intell. Syst.*, vol. 6, no. 3, pp. 573–590, 2020, doi: 10.1007/s40747-020-00154-3.

[2] W. Martin, F. Sarro, Y. Jia, Y. Zhang, and M. Harman, "A survey of app store analysis for software engineering," *IEEE Trans. Softw. Eng.*, vol. 43, no. 9, pp. 817–847, 2017, doi: 10.1109/TSE.2016.2630689.

[3] M. Razavi, H. Rastegari, and A. Nabiollahi-najafabadi, "User Reviews Classification in Play Store Applications Using Deep Learning : An Empirical Study," pp. 43–57.

[4] E. Guzman and W. Maalej, "How Do Users Like This Feature? A Fine Grained Sentiment Analysis of App Reviews," in *2014 IEEE 22nd International Requirements Engineering Conference (RE)*, IEEE, Aug. 2014, pp. 153–162. doi: 10.1109/RE.2014.6912257.

[5] D. Pagano and W. Maalej, "User feedback in the appstore: An empirical study," in *2013 21st IEEE International Requirements Engineering Conference (RE)*, IEEE, Jul. 2013, pp. 125–134. doi: 10.1109/RE.2013.6636712.

[6] L. V. G. Carreno and K. Winbladh, "Analysis of user comments: An approach for software requirements evolution," *Proc. - Int. Conf. Softw. Eng.*, pp. 582–591, 2013, doi: 10.1109/ICSE.2013.6606604.

[7] W. Maalej and D. Pagano, "On the socialness of software," *Proc. - IEEE 9th Int. Conf. Dependable, Auton. Secur. Comput. DASC 2011*, pp. 864–871, 2011, doi: 10.1109/DASC.2011.146.

[8] N. Seyff, F. Graf, and N. Maiden, "Using mobile RE tools to give end-users their own voice," *Proc. 2010 18th IEEE Int. Requir. Eng. Conf. RE2010*, pp. 37–46, 2010, doi: 10.1109/RE.2010.15.

[9] A. Al-Subaihin *et al.*, "App store mining and analysis," in *Proceedings of the 3rd International Workshop on Software Development Lifecycle for Mobile*, New York, NY, USA: ACM, Aug. 2015, pp. 1–2. doi: 10.1145/2804345.2804346.

[10] N. Chen, J. Lin, S. C. H. Hoi, X. Xiao, and B. Zhang, "AR-miner: Mining informative reviews for developers from mobile app marketplace," in *Proceedings - International Conference on Software Engineering*, IEEE Computer Society, May 2014, pp. 767–778. doi: 10.1145/2568225.2568263.

[11] E. C. Groen, J. Doerr, and S. Adam, "Towards Crowd-Based Requirements Engineering A Research Preview," in *Requirements Engineering: Foundation for Software Quality*, S. A. Fricker and K. Schneider, Eds., Cham: Springer International Publishing, 2015, pp. 247–253.

[12] S. A. Licorish, B. T. R. Savarimuthu, and S. Keertipati, "Attributes that predict which features to fix: Lessons for app store mining," *ACM Int. Conf. Proceeding Ser.*, vol. Part F1286, pp. 108–117, 2017, doi: 10.1145/3084226.3084246.

[13] H. P. Luhn, "The Automatic Creation of Literature Abstracts," *IBM J. Res. Dev.*, vol. 2, no. 2, pp. 159–165, 2010, doi: 10.1147/rd.22.0159.

[14] K. S. Kalyan, A. Rajasekharan, and S. Sangeetha, "AMMUS : A Survey of Transformer-based Pretrained Models in Natural Language Processing," pp. 1–42, 2021, [Online]. Available: http://arxiv.org/abs/2108.05542

[15] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[16] A. M. and G. H. Alex Graves, "Speech Recognition with Deep Recurrent Neural Networks , Department of Computer Science, University of Toronto," *Dep. Comput. Sci. Univ. Toronto*, vol. 3, no. 3, pp. 45–49, 2013, [Online]. Available: https://ieeexplore.ieee.org/stampPDF/getPDF.jsp?tp=&arnumber=6638947&ref=aHR0cHM6Ly9pZWVleHBsb3JlLmllZWUub3JnL2Fic3RyYWN0L2RvY3VtZW50LzY2Mzg5NDc/Y2FzYV90b2tlbj1OQUo1VFJxWk5JRUFBQUFBOmtPZmdDbS00NGhqaGI2N3dMd2JrU3lSaEdJREhBWnpMSkxoT201Um5YMXR0S0poUDAtM2hkbT

[17] Y. Kim, "Convolutional Neural Networks for Sentence Classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2014, pp. 1746–1751. doi: 10.3115/v1/D14-1181.

[18] V. Gupta and G. S. Lehal, "A Survey of Text Summarization Extractive techniques," *J. Emerg. Technol. Web Intell.*, vol. 2, no. 3, pp. 258–268, 2010, doi: 10.4304/jetwi.2.3.258-268.

[19] N. Moratanch and S. Chitrakala, "A survey on extractive text summarization," *Int. Conf. Comput. Commun. Signal Process. Spec. Focus IoT, ICCCSP 2017*, no. November, 2017, doi: 10.1109/ICCCSP.2017.7944061.

[20] M. Allahyari *et al.*, "Text Summarization Techniques: A Brief Survey," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 10, 2017, doi: 10.14569/ijacsa.2017.081052.

[21] N. Moratanch and S. Chitrakala, "A survey on abstractive text summarization," *Proc. IEEE Int. Conf. Circuit, Power Comput. Technol. ICCPCT 2016*, no. November, 2016, doi: 10.1109/ICCPCT.2016.7530193.

[22] S. Gholamrezazadeh, M. A. Salehi, and B. Gholamzadeh, "A Comprehensive Survey on Text Summarization Systems." doi: 10.1109/CSA.2009.5404226.

[23] G. L. A. Babu and S. Badugu, "A Survey on Automatic Text Summarisation," *Lect. Notes Networks Syst.*, vol. 612, pp. 679–689, 2014, doi: 10.1007/978-981-19-9228-5_58.

[24] R. Mishra *et al.*, "Text summarization in the biomedical domain: a systematic review of recent research.," *J. Biomed. Inform.*, vol. 52, pp. 457–467, Dec. 2014, doi: 10.1016/j.jbi.2014.06.009.

[25] N. Andhale and L. A. Bewoor, "An overview of text summarization techniques," *Proc. - 2nd Int. Conf. Comput. Commun. Control Autom. ICCUBEA 2016*, no. May, 2017, doi: 10.1109/ICCUBEA.2016.7860024.

[26] J. R. Thomas, S. K. Bharti, and K. S. Babu, "Automatic keyword extraction for text summarization in e-newspapers," *ACM Int. Conf. Proceeding Ser.*, vol. 25-26-August-2016, 2016, doi: 10.1145/2980258.2980442.

[27] M. Gambhir and V. Gupta, "Recent automatic text summarization techniques: a survey," *Artif. Intell. Rev.*, vol. 47, 2017, doi: 10.1007/s10462-016-9475-9.

[28] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, "Automatic text summarization: A comprehensive survey," *Expert Syst. Appl.*, vol. 165, no. November 2021, 2021, doi: 10.1016/j.eswa.2020.113679.

[29] L. Abualigah, M. Q. Bashabsheh, H. Alabool, and M. Shehab, "Text Summarization: A Brief Review," *Stud. Comput. Intell.*, vol. 874, no. January, pp. 1–15, 2020, doi: 10.1007/978-3-030-34614-0_1.

[30] M. Lewis *et al.*, "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension," *CoRR*, vol. abs/1910.1, Oct. 2019, doi: https://doi.org/10.48550/arXiv.1910.13461.

[31] S. Narayan, S. B. Cohen, and M. Lapata, "Don't give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization," *Proc. 2018 Conf. Empir. Methods Nat. Lang. Process. EMNLP 2018*, pp. 1797–1807, 2018, doi: 10.18653/v1/d18-1206.

[32] T. Hasan *et al.*, "{XL}-Sum: Large-Scale Multilingual Abstractive Summarization for 44 Languages," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, Online: Association for Computational Linguistics, Aug. 2021, pp. 4693–4703.

[33] L. Basyal and M. Sanghvi, "Text Summarization Using Large Language Models: A Comparative Study of MPT-7b-instruct, Falcon-7b-instruct, and OpenAI Chat-GPT Models," 2023.

[34] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, 2004, pp. 74–81.