Celal Bayar University Journal of Science
Volume 22, Issue 2, 2026, p 354-361
Doi:10.18466/cbayarfbe.1740994
Research Article

Celal Bayar University Journal of Science

An Application of Bellman-Ford Algorithm in Hyperbolic Graph

İbrahim Gökcan[1]*

[1]Artvin Çoruh University, Faculty of Arts and Sciences, 08000 Artvin, Türkiye.
* gokcan@artvin.edu.tr
*Orcid No:0000-0002-6933-8494

Received: July 12, 2025
Accepted: April 8, 2026
DOI: 10.18466/cbayarfbe.1740994

Abstract

In this study, the classical Bellman-Ford algorithm, designed to operate on directed graphs with negative edge weights, is examined and accepted in a manner that allows each edge to be traversed only once. It is important to note that this study does not propose a new algorithm. Then, this algorithm is applied to the Farey graph F, defined on the hyperbolic plane $\mathbb{H}^2$ and constructed Farey fractions. The primary objective is to determine the shortest paths and their corresponding minimum weights from a designated source vertex to all other vertices in the graph. The Farey graph F is structured based on the orbits of the modular group $\mathrm{PSL}(2,\mathbb{Z})$, with edges determined by the group action. Each vertex represents a Farey fraction, and an edge exists between two vertices if the corresponding fractions are Farey neighbours. Edge weights are assigned in accordance with the modular group's influence on the respective suborbital structures, resulting in a graph composed of bidirectional edges. Then, upon applying the Bellman-Ford algorithm to the graph F, it was observed that all vertices had reached minimum values. Furthermore, using the mediant operation, Farey graphs with edges of $1,3,7,15$, and 31 were constructed. For each graph, numerical sequences were derived based on the minimum possible weights that vertices can attain. These sequences reflect the metric characteristics of graph expansions in the hyperbolic plane.

Keywords: Bellman-Ford algorithm, Farey graph, Shortest path

1. Introduction

Graph theory has been studied in mathematics for many years and is associated with different disciplines. Moreover, it is an important tool in the effective use of mathematics as a language of science. By systematizing problems that seem difficult and complex to solve, it makes solutions easier, and moreover, it provides solutions with data that can be obtained more quickly, thus increasing the interest in graph theory.

Graphs are essentially formed by vertices, and edges connecting vertices. Based on the concept of edge direction, graphs are applicable in many disciplines. For example, in addition to Euclidean geometry, one of the truths accepted in the world of science for many years, graphs are used in non-Euclidean geometries, which are relatively new.

Algorithms, the basis of computer science, accelerate the paper and pencil operations of graphs by bringing them to the electronic platform, and thus enhance interdisciplinary collaboration between algorithms and graph theory. In graph theory, one of the most investigated problems is the shortest path problem. Numerous algorithms have been developed for solving such problems. Bellman-Ford and Dijkstra algorithms are examples of algorithms that give shortest paths from a source to other vertices, and the Floyd-Warshall algorithm gives shortest paths from multiple source vertices to other vertices in a graph. The shortest path algorithms differ from each other by some operational constraints and properties. Some examples of these differences are runtime, negatively weighted edges, and shorter results time. In contrast to these algorithms, Breadth-First Search (BFS) algorithms are also available, which find the shortest paths from the source vertex to other vertices, regardless of the weight or length of the edges, assuming all edge values in the graph are equal.

In this study, the hyperbolic Farey graph serves as a bridge between hyperbolic geometry and number theory. Here, the Bellman-Ford algorithm is accepted in a manner that allows each edge to be traversed only once can be considered a graph-specific application

designed to compute shortest paths efficiently within the structure of this graph. Since the Farey graph has a tree-like structure, the algorithm yields faster and more efficient shortest path computations. Moreover, the algorithm's ability to operate on graphs with negatively weighted edges makes it one of the few applicable algorithms for the Farey graph. In particular, in applications where hyperbolic graphs, especially Farey graphs are employed, the algorithm can offer a practical solution for obtaining low-latency and cost-effective shortest paths.

1.1 Non-Euclidean Farey Graph F and Algorithm Applications

A graph mainly consists of vertices and edges connecting the vertices. There are some other basic concepts related to graphs. In a graph, in case an edge has directional information, all edges in the graph have directional information; otherwise, if an edge does not have directional information, no edge has directional information. The sequence of the edges that can be traversed between two vertices in a graph is called a path. A cycle is formed when a path ends at the vertex it started at and does not pass through a vertex twice. A graph is a tree if it contains no cycles. Edge weights influence the value of a graph as all edges have values that contribute to this determination. A graph can be considered simple when all edges have the same weight. For an introduction to the basic principles and developments in graph theory, see [1],[2],[3],[4],[5],[6],[7],[8],[9],[10],[11],[12].

In graph theory, using group theory and in particular modular groups, vertices and edges are associated with action of the group. In this context, the ideas of [13] can be given as examples. [13] described the action of G on $\Omega \times \Omega$ as $g:(\alpha,\beta) \to (g(\alpha), g(\beta))$ where $g \in G$ and $\alpha, \beta \in \Omega$. The resulting (G, Ω) is a permutative group. In addition, the suborbital graph $G(\alpha,\beta)$ is composed of suborbital's $O(\alpha,\beta)$. Thus, if (α,β) in the suborbital $O(\alpha,\beta)$, it represents an edge from α to β and is symbolically denoted by $\alpha \to \beta$. The resulting edge represents a geodesic edge in the hyperbolic plane. In addition to suborbital $O(\alpha,\beta)$, $O(\beta,\alpha)$ is also an orbital. These two orbitals can be equal or different. If they are equal, $G(\alpha,\beta) = G(\beta,\alpha)$. In these graphs, each edge consists of a bidirection. Hence, a bidirectional edge can be assumed as an undirected edge and a simple graph is formed. However, in case the orbital's are different, different suborbitals $G(\alpha,\beta)$ and $G(\beta,\alpha)$ are formed. In this case, the direction of one edge is reversed in the other graph. These are called paired suborbital graphs. [14] analyzed the case of $G = \Gamma$ and $\Omega = \widehat{\mathbb{Q}}$ in suborbital graph action and associated suborbital graphs with modular groups. In particular, suborbital graphs $G_{u,N}$ and $F_{u,N}$ for $(u,N) = 1, u \leq N$ via modular group Γ and its subgroups were defined in [14]. Moreover, graph elements formed by modular group Γ and its equivalence subgroup were studied. Furthermore, [14, Theorem 3.1 and 5.1] were proved for edge between vertices in suborbital graphs $G_{u,N}$ and $F_{u,N}$. That means, for $(u,N) = 1, u \leq N$, the condition for an edge from $\frac{r}{s}$ to $\frac{x}{y}$ in the suborbital graph $G_{u,N}$ was presented by $x \equiv ur\ (mod\ N)$, $y \equiv us\ (mod\ N)$ and $ry - sx = N$ or $x \equiv -ur\ (mod\ N)$, $y \equiv -us\ (mod\ N)$ and $ry - sx = -N$ depending on the direction of the edge. A similar condition was also given for $F_{u,N}$ since the disjoint union of the suborbital graphs $F_{u,N}$ forms the suborbital graph $G_{u,N}$. So, the condition for an edge from $\frac{r}{s}$ to $\frac{x}{y}$ in the suborbital graph $F_{u,N}$ was given as $x \equiv ur\ (mod\ N)$ and $ry - sx = N$ or $x \equiv -ur\ (mod\ N)$ and $ry - sx = -N$ with respect to the direction of the edge. Specifically, if $u = 1$ and $N = 1$, the Farey graph $G_{1,1} = F$ is obtained. In addition, the following Lemma describes the action between vertices in a hyperbolic Farey graph.

Lemma 1.1.1 Assume that $\frac{r}{s}$ and $\frac{x}{y}$ are reduced rational numbers in the Farey graph F. So, the following conditions are equivalent to each other.

i. $\frac{r}{s}$ and $\frac{x}{y}$ are consecutive vertices in Farey graph.
ii. $ry - sx = \pm 1$
iii. $\frac{r}{s}$ and $\frac{x}{y}$ are consecutive vertices in Farey sequence.

In this Lemma, the necessary and sufficient condition that $\frac{r}{s}$ and $\frac{x}{y}$ are neighboring vertices in Farey graph F was given as $ry - sx = \pm 1$. See [14, Lemma 4.1].

More detailed information on the formation of suborbital graphs via modular groups can be found in references [14], [15], [16]. In particular, reference [16] presents the vertices of suborbital graphs in terms of continued fractions and demonstrates that the condition of minimal distance between vertices is satisfied. Moreover, see [17], [18] for applications of shortest path algorithms on non-Euclidean graphs. In an interdisciplinary approach, graph theory makes complex problems easier by visualizing them. Algorithms obtain data by dividing these problems into steps and then present the gathered data as a meaningful whole, thus increasing the relation between these two disciplines.

1.2. Bellman-Ford Algorithm

The Bellman-Ford algorithm provides shortest paths from the source vertex to other vertices in a weighted graph. The most important difference from the other shortest path algorithms is that the Bellman-Ford algorithm can run on negative edge weights and can detect the presence of negative cycles. Due to this feature, unlike other shortest path algorithms, this algorithm has a wider range of applications. First, if there is a negative weight reachable from the source, or

Celal Bayar University Journal of Science
Volume 22, Issue 2, 2026, p 354-361
Doi:10.18466/cbayarfbe.1740994

if the sum of weights in the cycle is negative, it can be concluded that there is no solution. This is because with each revolution around the cycle we find a shorter path. Naturally, we have to try numerous times to find the shortest path. If a negative weighted cycle is reached from the source vertex, the shortest path algorithm is stopped.

In the Bellman-Ford algorithm, the number of iterative steps is based on the assumption that for a graph $G = (V, E)$, essentially a graph with vertices $|V|$, the shortest paths from the source vertex to the other vertices are obtained in iterative time $|V| - 1$. Since any path can be longer than a given threshold there might be negative weights if the length is still changing after obtaining the calculated value |V|. For the Bellman-Ford algorithm, the shortest path depends on the vertex that is previously traveled.

Assume that in a negative weighted graph, the path from vertex u to vertex v travels through at most vertices l at symbolize such path as follows:

$\delta_G^l(u, v)$=w: Shortest paths from vertex u to vertex v via at most vertices l

First, let us determine the values of vertices so that 0 for u and ∞ for all other vertices v, i.e,

$$\text{For } u \in V, \delta_G^0(u,u) = 0,$$
$$\forall v \neq u \in V, \delta_G^l(u,v) = \infty.$$

Hence, in order to determine $\delta_G^1(u, v)$ for $\forall v \neq u \in V$, the values given in the first step can be used. More generally, $\delta_G^{l+1}(u, v)$ is calculated from $\delta_G^l(u, v)$. For this, the shortest paths $l + 1$ from vertex u to vertex v has to be determined. In other words, it is obtained by finding shortest paths to each neighboring vertex of v and summing their weights. The following equation explains this situation.

$$\delta^{l+1}(v) = \min\left(\delta^l(v), \min_{v \in N(v)}\left(\delta^l(u)\right) + w(u,v)\right)$$

where $N(v)$ is a set of neighbouring vertices of v.
However, the pseudo-code of the Bellman-Ford algorithm can be given as follows:

Algorithm 1. Pseudocode for Bellman-Ford algorithm

```
1  For each v ∈ V
2    do d[v] ← ∞        } initialization
3       d[s] ← 0
4  for 1 ← k to |V| − 1
5  do for each edge (u, v) ∈ E
       do if d[v] > d[u] + w(u, v)
            then d[v] ← d[u] + w(u, v)   } relaxion step
   for each edge (u, v) ∈ E
   do if d[v] > d[u] + w(u, v)
      then a report that a negative
                − weight cycle exists
```

In a graph consisting of V vertices and E edges, the time complexity of the Bellman-Ford algorithm is given $O(V.E)$, and each edge is traversed is $|V| - 1$ times. However, in this study, since it is assumed that each edge is traversed only once, the time complexity can be given as $O(E)$. For more detailed information on the Bellman-Ford algorithm, see references [19],[20].

In study [21], the Act Colony Optimization algorithm was investigated within the context of fuzzy logic for addressing shortest path problems in networks characterized by various fuzzy-weighted structures. The weights of fuzzy paths comprising different types of fuzzy arcs were approximated, and a signed distance function was employed to enable a comparative evaluation of these paths. In study [22], the Bellman-Ford and Floyd-Warshall algorithms were analyzed with the aim of determining the shortest path in LPG distribution systems, and the findings demonstrated that the Floyd-Warshall algorithm was more effective in identifying the minimum-length path. In study [23], the shortest path problem was investigated in the setting of linear Diophantine fuzzy graphs, where inter-nodal distances and associated costs in the proposed network model were modeled using linear Diophantine fuzzy numbers, and the Bellman-Ford algorithm was applied to obtain the optimal solution. Furthermore, study [24] addressed the problem of reconstructing two shortest paths, presenting polynomial-time algorithms to solve practical applications related to network reconfiguration.

2. Main Results

The classical Bellman-Ford pseudo-code, organized based on the rule of passing only once on each edge, is presented below. The Bellman-Ford algorithm is studied on the non-Euclidean Farey graph.

Algorithm 2. Pseudocode for The Bellman-Ford accepted in a manner that allows each edge to be traversed only once

```
1  INPUT:
2  G = (V, E) → a directed weighted graph
3  source → the starting vertex
4  BEGIN:
5  For each vertex v in V:
                for k = 0 to |V| − 1
   distance[v][k] ← ∞
   distance[s][0] ← 0
   for k = 1 to |V| − 1
            For each edge (u, v, w) in E:
        If d[u][k − 1] + w < d[v][k] then:
            d[v][k] ← d[u][k − 1] + w
```

Celal Bayar University Journal of Science
Volume 22, Issue 2, 2026, p 354-361
Doi:10.18466/cbayarfbe.1740994

	$\min d[v] = \min(dist[v][0 \cdots \lvert V\rvert - 1])$ END

In addition, equation related to action between two vertices is taken as follows:

$$\delta^m(v) = \min\left(\delta^n(v), \min_{v \in N(v)}\left(\delta^n(u)\right) + w(u,v)\right)$$

Let m and n denote the number of edges to reach vertex u or v. In the previous equation, $\delta^{l+1}(v)$ defined the shortest path $l+1$ from the source vertex u to vertex v. Whereas in the equation $\delta^m(v)$, the number of edges reaching the vertices neighboring v in the previous step is taken into account and as a result, the same data will be obtained. The same edge cannot be used more than once a path in this study. This rule is consistent with the basic principle of the Bellman-Ford algorithm. Not only the vertices but also "k: the order of vertices" should be taken into account in this algorithm. However, the Bellman-Ford algorithm in given Algorithm 2 does not allow using of same edge again unlike the classical algorithm and requires more detailed tracking. In terms of iterations, the classical algorithm has the potential for infinite cycles, while the Bellman-Ford algorithm is used in this study has $|V|$ iterations and does not allow negative cycles. Moreover, this pseudo-code can be considered as a less qualified version of Bellman-Ford's classical pseudo-code. However, in terms of relaxing edges, performs like classical algorithm. Furthermore, it is useful to restrict the cycles possible in the graph. This algorithm is applied to the hyperbolic Farey graph defined in non-Euclidean plane. Additionally, the weights of the edges are associated to the edge condition in accordance with [14, Lemma 4.1]. Based on the direction of action, black colored edges represent right-directional actions and red colored edges represent left-directional actions. The weights of edges can be given as follows for $\frac{r}{s}$ and $\frac{x}{y}$ are neighboring vertices in Farey graph F:

$$\begin{matrix}1\\-1\end{matrix} = \begin{cases}\text{If the condition } ry - sx = +1 \text{ is satisfied,}\\ \text{the value of the black colored edges}\\ \text{If the condition } ry - sx = -1 \text{ is satisfied,}\\ \text{the value of the red colored edges}\end{cases}$$

First, the operations resulting when applying the Bellman-Ford algorithm to a hyperbolic Farey graph with one edge are given below, and the sequence obtained is given in the table:

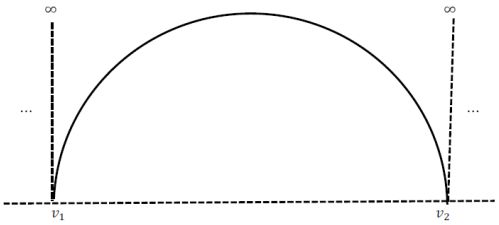


Figure 1. Farey graph with 1 edge

Let the vertices $v_1 = \frac{0}{1}$ and $v_2 = \frac{1}{1}$ in the Farey graph given in Figure 1. Then, the vertex values according to the shortest paths between vertices can be presented as follows. For $1 \le i \le 2$ and $v_i \in F$,

$$\delta_F^0(v_1, v_1) = 0, \delta_F^0(v_1, v_2) = \infty$$
$$\delta_F^1(v_1, v_2) = 1$$

The following table of values can be given here.

Table 1. Value table for Farey graph with 1 edge

Number of Steps	v_1	v_2
	0	∞
1	0	1

In Table 1, since v_1 is the source vertex and v_1 and v_2 are consecutive vertices, the values of vertices are found in 1 step.

Suppose a hyperbolic Farey graph with 3 edges giving the shortest paths is shown as in the following.

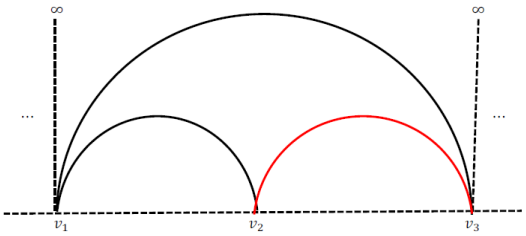


Figure 2. Farey graph with 3 edges

In the Farey graph given in Figure 2, let us assume the vertices $v_1 = \frac{0}{1}, v_2 = \frac{1}{2}$ and $v_3 = \frac{1}{1}$. Then, the vertex values based on the shortest paths can be found below. For $1 \le i \le 3$ and $v_i \in F$,

$$\delta_F^0(v_1, v_1) = 0, \delta_F^0(v_1, v_2) = \infty, \delta_F^0(v_1, v_3) = \infty$$
$$\delta_F^1(v_1, v_2) = 1, \delta_F^1(v_1, v_3) = 1$$
$$\delta^2(v_2) = \min\left(\delta^1(v_2), \min_{v_2 \in N(v_i)}\left(\delta^1(v_3)\right) + w(v_3, v_2)\right)$$
$$= \min(1,0) = 0$$

Celal Bayar University Journal of Science
Volume 22, Issue 2, 2026, p 354-361
Doi:10.18466/cbayarfbe.1740994

Table 2. Value table for Farey graph with 3 edges

Number of Steps	v_1	v_2	v_3
	0	∞	∞
1	0	1	1
2	0	0	1

In Table 2, since v_1 is the source vertex, v_2 and v_3 are obtained as 1. However, if the edge between v_2 and v_3 is assumed as negative weight, v_2 is achieved as 0.

Let the hyperbolic Farey graph with 7 edges according to the shortest path condition is given as below.

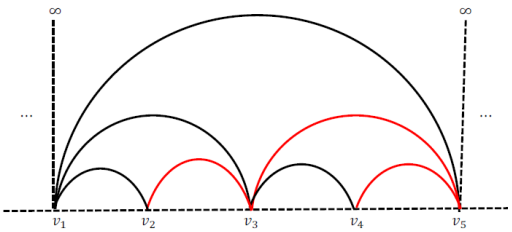


Figure 3. Farey graph with 7 edges

Let $v_1 = \frac{0}{1}, v_2 = \frac{1}{3}, v_3 = \frac{1}{2}, v_4 = \frac{2}{3}$ and $v_5 = \frac{1}{1}$ for the graph in Figure 3. From this, the vertex values for the shortest paths are obtained and given in the table below:

$$\delta_F^0(v_1, v_1) = 0, \delta_F^0(v_1, v_2) = \infty, \delta_F^0(v_1, v_3) = \infty, \delta_F^0(v_1, v_4) = \infty, \delta_F^0(v_1, v_5) = \infty$$

$$\delta_F^1(v_1, v_2) = 1, \delta_F^1(v_1, v_3) = 1, \delta_F^1(v_1, v_5) = 1$$
$$\delta_F^2(v_1, v_4) = 2$$
$$\delta^2(v_2) = \min\left(\delta^1(v_2), \min_{v_2 \in N(v_i)}\left(\delta^1(v_3)\right) + w(v_3, v_2)\right) = \min(1,0) = 0$$
$$\delta^2(v_3) = \min\left(\delta^1(v_3), \min_{v_3 \in N(v_i)}\left(\delta^1(v_5)\right) + w(v_5, v_3)\right) = \min(1,0) = 0$$
$$\delta^2(v_4) = \min\left(\delta^2(v_4), \min_{v_4 \in N(v_i)}\left(\delta^1(v_5)\right) + w(v_5, v_4)\right) = \min(2,0) = 0$$

Table 3: Value table for Farey graph with 7 edges

Number of Steps	v_1	v_2	v_3	v_4	v_5
	0	∞	∞	∞	∞
1	0	1	1	∞	1
2	0	1	1	∞	1
3	0	0	1	2	1
4	0	0	0	0	1

In table 3, the vertices that can be reached directly or via an edge from the source vertex v_1; since they can be reached from consecutive vertices via negatively weighted edge, only the value of v_5 is found as 1, while the other vertices are 0.

A hyperbolic Farey graph with 15 edges and shortest paths is given as follows.

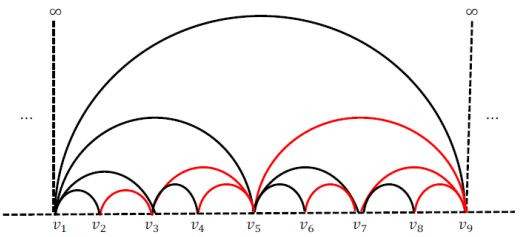


Figure 4. Farey graph with 15 edges

Let us assume that vertices $v_1 = \frac{0}{1}, v_2 = \frac{1}{4}, v_3 = \frac{1}{3}, v_4 = \frac{2}{5}, v_5 = \frac{1}{2}, v_6 = \frac{3}{5}, v_7 = \frac{2}{3}, v_8 = \frac{3}{4}$ and $v_9 = \frac{1}{1}$ on Farey graph. In this case, operations and table related to vertices in below:

$$\delta_F^0(v_1, v_1) = 0, \delta_F^0(v_1, v_2) = \infty, \delta_F^0(v_1, v_3) = \infty, \delta_F^0(v_1, v_4) = \infty, \delta_F^0(v_1, v_5) = \infty, \delta_F^0(v_1, v_6) = \infty, \delta_F^0(v_1, v_7) = \infty, \delta_F^0(v_1, v_8) = \infty, \delta_F^0(v_1, v_9) = \infty$$
$$\delta_F^1(v_1, v_2) = 1, \delta_F^1(v_1, v_3) = 1, \delta_F^1(v_1, v_5) = 1, \delta_F^1(v_1, v_9) = 1$$
$$\delta_F^2(v_1, v_4) = 2, \delta_F^2(v_1, v_6) = 2, \delta_F^2(v_1, v_7) = 2$$
$$\delta_F^3(v_1, v_8) = 3$$
$$\delta^2(v_2) = \min\left(\delta^1(v_2), \min_{v_2 \in N(v_i)}\left(\delta^1(v_3)\right) + w(v_3, v_2)\right) = \min(1,0) = 0$$
$$\delta^2(v_3) = \min\left(\delta^1(v_3), \min_{v_3 \in N(v_i)}\left(\delta^1(v_5)\right) + w(v_5, v_3)\right) = \min(1,0) = 0$$
$$\delta^2(v_4) = \min\left(\delta^2(v_4), \min_{v_4 \in N(v_i)}\left(\delta^1(v_5)\right) + w(v_5, v_4)\right) = \min(2,0) = 0$$
$$\delta^2(v_5) = \min\left(\delta^1(v_5), \min_{v_5 \in N(v_i)}\left(\delta^1(v_9)\right) + w(v_9, v_5)\right) = \min(1,0) = 0$$
$$\delta^3(v_6) = \min\left(\delta^2(v_6), \min_{v_6 \in N(v_i)}\left(\delta^2(v_7)\right) + w(v_7, v_6)\right) = \min(2,1) = 1$$
$$\delta^2(v_7) = \min\left(\delta^2(v_7), \min_{v_7 \in N(v_i)}\left(\delta^1(v_9)\right) + w(v_9, v_7)\right) = \min(2,0) = 0$$
$$\delta^2(v_8) = \min\left(\delta^3(v_8), \min_{v_8 \in N(v_i)}\left(\delta^1(v_9)\right) + w(v_9, v_8)\right) = \min(3,0) = 0$$

Since the vertices that received values in previous step in Table 4 could be reached via negatively weighted paths from consecutive vertices, only values of vertices v_6 and v_9 are found as 1, while the others were 0.

Celal Bayar University Journal of Science
Volume 22, Issue 2, 2026, p 354-361
Doi:10.18466/cbayarfbe.1740994

Table 4. Value table for Farey graph with 15 edges

Number of Steps	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9
	0	∞	∞	∞	∞	∞	∞	∞	∞
1	0	1	1	∞	1	∞	∞	∞	1
2	0	1	1	∞	1	∞	∞	∞	1
3	0	0	1	2	1	∞	∞	∞	1
4	0	0	1	2	1	∞	∞	∞	1
5	0	0	0	0	1	2	2	∞	1
6	0	0	0	0	1	2	2	∞	1
7	0	0	0	0	1	1	2	3	1
8	0	0	0	0	0	1	0	0	1

The hyperbolic Farey graph, consisting of 31 edges and providing the shortest paths and operations obtained for the vertex values, is presented below. The corresponding data are provided in the tables.

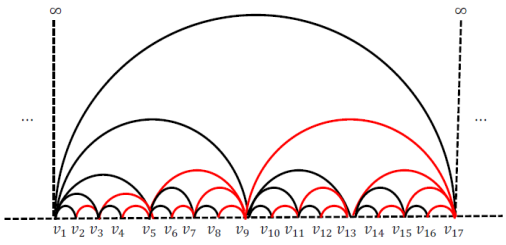


Figure 5. Farey graph with 31 edges

Let $v_1 = \frac{0}{1}, v_2 = \frac{1}{5}, v_3 = \frac{1}{4}, v_4 = \frac{2}{7}, v_5 = \frac{1}{3}, v_6 = \frac{3}{8}, v_7 = \frac{2}{5}, v_8 = \frac{3}{7}, v_9 = \frac{1}{2}, v_{10} = \frac{4}{7}, v_{11} = \frac{3}{5}, v_{12} = \frac{5}{8}, v_{13} = \frac{2}{3}, v_{14} = \frac{5}{7}, v_{15} = \frac{3}{4}, v_{16} = \frac{4}{5}$ and $v_{17} = \frac{1}{1}$ are vertices on Farey graph.

$$\delta_F^0(v_1, v_1) = 0, \delta_F^0(v_1, v_2) = \infty, \delta_F^0(v_1, v_3) = \infty, \delta_F^0(v_1, v_4) = \infty, \delta_F^0(v_1, v_5) = \infty,$$

$$\delta_F^0(v_1, v_6) = \infty, \delta_F^0(v_1, v_7) = \infty, \delta_F^0(v_1, v_8) = \infty, \delta_F^0(v_1, v_9) = \infty, \delta_F^0(v_1, v_{10}) = \infty,$$

$$\delta_F^0(v_1, v_{11}) = \infty, \delta_F^0(v_1, v_{12}) = \infty, \delta_F^0(v_1, v_{13}) = \infty, \delta_F^0(v_1, v_{14}) = \infty, \delta_F^0(v_1, v_{15}) = \infty,$$

$$\delta_F^0(v_1, v_{16}) = \infty, \delta_F^0(v_1, v_{17}) = \infty.$$

$$\delta_F^1(v_1, v_2) = 1, \delta_F^1(v_1, v_3) = 1, \delta_F^1(v_1, v_5) = 1, \delta_F^1(v_1, v_9) = 1, \delta_F^1(v_1, v_{17}) = 1$$

$$\delta_F^2(v_1, v_4) = 2, \delta_F^2(v_1, v_6) = 2, \delta_F^2(v_1, v_7) = 2, \delta_F^2(v_1, v_{10}) = 2, \delta_F^2(v_1, v_{11}) = 2, \delta_F^2(v_1, v_{13}) = 2$$

$$\delta_F^3(v_1, v_8) = 3, \delta_F^3(v_1, v_{12}) = 3, \delta_F^3(v_1, v_{14}) = 3, \delta_F^3(v_1, v_{15}) = 3$$

$$\delta_F^4(v_1, v_{16}) = 4$$

$$\delta^2(v_2) = \min\left(\delta^1(v_2), \min_{v_2 \in N(v_i)}\left(\delta^1(v_3)\right) + w(v_3, v_2)\right) = \min(1,0) = 0$$

$$\delta^2(v_3) = \min\left(\delta^1(v_3), \min_{v_3 \in N(v_i)}\left(\delta^1(v_5)\right) + w(v_5, v_3)\right) = \min(1,0) = 0$$

$$\delta^2(v_4) = \min\left(\delta^2(v_4), \min_{v_4 \in N(v_i)}\left(\delta^1(v_5)\right) + w(v_5, v_4)\right) = \min(2,0) = 0$$

$$\delta^2(v_5) = \min\left(\delta^1(v_5), \min_{v_5 \in N(v_i)}\left(\delta^1(v_9)\right) + w(v_9, v_5)\right) = \min(1,0) = 0$$

$$\delta^3(v_6) = \min\left(\delta^2(v_6), \min_{v_6 \in N(v_i)}\left(\delta^2(v_7)\right) + w(v_7, v_6)\right) = \min(2,1) = 1$$

$$\delta^2(v_8) = \min\left(\delta^3(v_8), \min_{v_8 \in N(v_i)}\left(\delta^1(v_9)\right) + w(v_9, v_8)\right) = \min(3,0) = 0$$

$$\delta^2(v_9) = \min\left(\delta^1(v_9), \min_{v_9 \in N(v_i)}\left(\delta^1(v_{17})\right) + w(v_{17}, v_9)\right) = \min(1,0) = 0$$

$$\delta^3(v_{10}) = \min\left(\delta^2(v_{10}), \min_{v_{10} \in N(v_i)}\left(\delta^2(v_{11})\right) + w(v_{11}, v_{10})\right) = \min(2,1) = 1$$

$$\delta^3(v_{11}) = \min\left(\delta^2(v_{11}), \min_{v_{11} \in N(v_i)}\left(\delta^2(v_{13})\right) + w(v_{13}, v_{11})\right) = \min(2,1) = 1$$

$$\delta^3(v_{12}) = \min\left(\delta^3(v_{12}), \min_{v_{12} \in N(v_i)}\left(\delta^2(v_{13})\right) + w(v_{13}, v_{12})\right) = \min(3,1) = 1$$

$$\delta^2(v_{13}) = \min\left(\delta^2(v_{13}), \min_{v_{13} \in N(v_i)}\left(\delta^1(v_{17})\right) + w(v_{17}, v_{13})\right) = \min(2,0) = 0$$

$$\delta^4(v_{14}) = \min\left(\delta^3(v_{14}), \min_{v_{14} \in N(v_i)}\left(\delta^3(v_{15})\right) + w(v_{15}, v_{14})\right) = \min(3,2) = 2$$

$$\delta^2(v_{15}) = \min\left(\delta^3(v_{15}), \min_{v_{15} \in N(v_i)}\left(\delta^1(v_{17})\right) + w(v_{17}, v_{15})\right) = \min(3,0) = 0$$

$$\delta^2(v_{16}) = \min\left(\delta^4(v_{16}), \min_{v_{16} \in N(v_i)}\left(\delta^1(v_{17})\right) + w(v_{17}, v_{16})\right) = \min(4,0) = 0$$

Table 5. Value table for Farey graph with 31 edges

Number of Steps	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	v_{12}	v_{13}	v_{14}	v_{15}	v_{16}	v_{17}
	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
1	0	1	1	∞	1	∞	∞	∞	1	∞	∞	∞	∞	∞	∞	∞	1
2	0	1	1	∞	1	∞	∞	∞	1	∞	∞	∞	∞	∞	∞	∞	1
3	0	0	1	2	1	∞	∞	∞	1	∞	∞	∞	∞	∞	∞	∞	1
4	0	0	1	2	1	∞	∞	∞	1	∞	∞	∞	∞	∞	∞	∞	1
5	0	0	0	0	1	2	2	∞	1	∞	∞	∞	∞	∞	∞	∞	1
6	0	0	0	0	1	2	2	∞	1	∞	∞	∞	∞	∞	∞	∞	1
7	0	0	0	0	1	1	2	3	1	∞	∞	∞	∞	∞	∞	∞	1
8	0	0	0	0	1	1	2	3	1	∞	∞	∞	∞	∞	∞	∞	1
9	0	0	0	0	0	1	0	0	1	2	2	∞	2	∞	∞	∞	1
10	0	0	0	0	0	1	0	0	1	2	2	∞	2	∞	∞	∞	1
11	0	0	0	0	0	1	0	0	1	1	2	3	2	∞	∞	∞	1
12	0	0	0	0	0	1	0	0	1	1	2	3	2	∞	∞	∞	1
13	0	0	0	0	0	1	0	0	1	1	1	1	2	3	3	∞	1
14	0	0	0	0	0	1	0	0	1	1	1	1	2	3	3	∞	1
15	0	0	0	0	0	1	0	0	1	1	1	1	2	2	3	4	1
16	0	0	0	0	0	1	0	0	1	1	1	1	2	2	3	4	1
17	0	0	0	0	0	1	0	0	0	1	1	1	0	2	0	0	1

Table 5 presents the values obtained directly from the source vertex or through consecutive vertices; also, the values that vertices can take when accessed is via negative weight paths.

3. Conclusion

The Farey graph is a mathematical structure in which rational numbers serve as vertices, and adjacent relations among them are defined through specific arithmetic rules. This graph can be naturally embedded in the hyperbolic plane, thereby necessitating analysis within a negatively curved geometric space rather than classical Euclidean geometry. An extended Farey graph $F = (V, E)$, when defined with bidirectional edges, can be treated as an undirected weighted graph. Given a weight function $w: E \to \mathbb{R}$ defined on $e \in E$, it becomes possible to analyze paths that may include negative weights. This feature is particularly important in contexts where classical shortest-path algorithms must account for potentially non-monotonic cost metrics. In this string, the Bellman-Ford algorithm emerges as a suitable method for computing the shortest paths from a source vertex $s \in V$ to all other vertices. While the standard form of the algorithm iteratively processes all edges $|V| - 1$ times to ensure global optimality, the Bellman-Ford algorithm in which each edge is relaxed only once can yield locally optimal or approximate solutions. This approximation is of particular interest in hyperbolic geometric contexts, where local minima may offer practically meaningful interpretations. The Bellman-Ford used in this study approach becomes especially relevant for hyperbolic networks characterized by directed edges and potentially negative weights. Its computational efficiency makes it an attractive tool for applications such as information diffusion, routing protocols, and geometric compression algorithms in the hyperbolic plane. Moreover, the interaction between the hyperbolic metric structure and the algorithm's behavior can provide insights that are not as easily observed in Euclidean settings. Therefore, applying the Bellman-Ford algorithm particularly on the extended Farey graph, offers new theoretical and practical perspectives in both computational geometry and hyperbolic network typologies.

Acknowledgement

The author appreciates the editor and the anonymous reviewers for their valuable time, insightful comments, and support in the evaluation of this manuscript.

Author's Contributions

The article is a single-authored article.

Ethics

There are no ethical issues for the publication of this manuscript.

References

[1]. Diestel, R. Graph Theory; Springer-Verlag Heidelberg: New York, 3. Edition, 2005.

[2]. Ruohonen, K. Graph Theory (Translation by Janne Tamminen, Kung-Chung Lee and Robert Piché);, 2013.

[3]. Gupta, C.B, Singh, S.R, Kumar, S. Advance Discrete Structure; I.K. International Publishing House Pvt. Ltd.: New Delhi, 2010.

[4]. Tremblay, J.P, Manohar, R. Discrete Mathematical Structures with Applications to Computer Science; Tata McGraw-Hill Edition: New Delhi, 2010.

[5]. Bal Gupta, S. Discrete Mathematics and Structures; University Science Press: New Delhi, 2008.

[6]. Das, M.K. Discrete Mathematical Structures for Computer Scientist and Engineers; Narosa Publishing House Pvt. Ltd.: New Delhi, 2008.

[7]. Veerarajan, T. Discrete Mathematics with Graph Theory and Combinatorics; Tata McGraw-Hill Edition: New Delhi, 2010.

[8]. Grimaldi, R.P. Discrete and Combinatorial Mathematics; Addision–Wesley, 1999.

[9]. Krishna Kumar, R. Discrete Mathematics; Laxmi Publications: New Delhi, 2006.

[10]. Somasundaram, R.M. Discrete Mathematical Structures; Prentice-Hall of India: New Delhi, 2007.

[11]. Aldous, J,M, Wilson, R.J. Graphs and Applications; Springer: London, 2000.

[12]. Cangül, İ.N. Graf Teori-I; Dora Basım-Yayın Dağıtım Ltd. Şti.: Bursa, 1. Baskı, 2017.

[13]. Sims, C.C. 1967. Graphs and finite permutation groups. *Mathematische. Zeitschrift;* 95, 76-86. (https://doi.org/10.1007/BF01117534)

[14]. Jones, G.A, Singerman, D, Wicks, K. 1991. The Modular group and generalized Farey graphs. *London Mathematical Society Lecture Note Series;* 160, 316-338. (https://doi.org/10.1017/CBO9780511661846.006)

[15]. Akbaş, M. 2001. On suborbital graphs for the Modular group, *Bulletin of the London Mathematical Society,* 33(6), 647-652. (https://doi.org/10.1112/S0024609301008311)

[16]. Deger, A.H. 2017. Vertices of Paths of Minimal Lengths on Suborbital Graphs, *Filomat,* 31(4), 913-923. (https://doi.org/10.2298/FIL1704913D)

[17]. Gökcan, İ. Application of Dijkstra Algorithm with Sequence F_m in $F_{u,N}$, In: Akgül, H, Aksoy, A (ed) *R*esearch&Reviews in Science and Mathematics, Gece Publishing, 2021, 1-19.

[18]. Gökcan, İ. 2025. Solution of Shortest Paths in Non-Euclidean Farey Graph with Floyd-Warshall Algorithm. *Süleyman Demirel University Faculty of Arts and Science Journal of Science;* 20(1), 63-74. (https://doi.org/10.29233/sdufeffd.1591711)

[19]. Ahuja, R.K, Magnanti, T.L, Orlin, J.B, Network Flows Theory, Algorithms and Applications; Prentice-Hall: New Jersey, 1993.

[20]. Cormen, T.H, Leiserson, C.E, Rivest, R.L., Stein, C. Intruction to Algorithms; The MIT Press: United States, 2009.

[21]. Caprio, D.D, Ebrahimnejad, A, Alrezaamiri, H, Santos-Arteaga, F. J. 2022. A Novel Ant Colony Algorithm for Solving Shortest Path Problems with Fuzzy Arc Weights, *Alexandria Engineering Journal,* 61, 3403–3415. (https://doi.org/10.1016/j.aej.2021.08.058)

[22]. Yusuf, M. R, Nurwan, N, Wungguli, D, Yahya, L. Implementation of the Floyd-Warshall Algorithm and Bellman-Ford Algorithm to Determine the Shortest Path in the Distribution of LPG Gas, Gorontalo, Indonesia, 2023, 1-6. (https://doi.org/10.1051/e3sconf/202340003004)

[23]. Kannan, V, Appasamy, S. 2023. Employing the Bellman-Ford Algorithm with Score Functions to Address the Linear Diophantine Fuzzy Shortest Path Problem in Network Analysis, *Mathematical Modelling of Engineering Problems;* 10(5), 1884-1892. (https://doi.org/10.18280/mmep.100542)

[24]. Gajjar, K, Jha, A.V, Kumar, M, Lahiri, A. 2024. Reconfiguring Shortest Paths in Graphs, *Algorithmica;* 86, 3309–3338. (https://doi.org/10.1007/s00453-024-01263-y)