

## TRANSPORT TARMOQLARIDA ENG QISQA YO‘LNI TOPISH USULLARI

***Farmonov Sherzodbek Raxmonjonovich***

*Farg‘ona davlat universiteti amaliy matematika va  
informatika kafedrası katta o‘qituvchisi  
[farmonovsh@gmail.com](mailto:farmonovsh@gmail.com)*

**Suyarov Ulug‘bek G‘iyosiddin o‘g‘li**

*Farg‘ona davlat universiteti talabasi  
[suyarov1857@gmail.com](mailto:suyarov1857@gmail.com)*

**Anotatsiya.** Transport tarmoqlarini samarali boshqarish va optimallashtirish murakkab grafiklar asosida eng qisqa yo‘lni topish masalalari bilan bog‘liqdir. Ushbu maqolada transport tizimlarida Ford-Bellman algoritmining amaliy qo‘llanilishi va uning samaradorligi tahlil qilinadi. Algoritmning asosiy tamoyillari, jumladan, manfiy og‘irlikli qirralar va tsikllar bilan ishlash qobiliyati, transport tarmoqlarida marshrutlarni rejalashtirishda tutgan o‘rni keng yoritiladi. Shuningdek, algoritmnining transport tizimlarida optimal yo‘lni aniqlashdagi afzalliklari va cheklovlari ko‘rib chiqiladi. Tadqiqot natijalari algoritmnining real sohalarda, jumladan, jamoat transporti tizimlari, logistika va telekommunikatsiya tarmoqlarida qo‘llanilish imkoniyatlarini ochib beradi.

**Kalit so‘zlar:** transport tarmoqlari, eng qisqa yo‘l, Ford-Bellman algoritmi, manfiy tsikllar, grafik modellashtirish, marshrut optimallashtirish, logistika, jamoat transporti, telekommunikatsiya.

**Аннотация.** Управление и оптимизация транспортных сетей тесно связаны с задачами поиска кратчайшего пути в сложных графах. В данной статье рассматривается практическое применение и эффективность алгоритма Форда-Беллмана в транспортных системах. Изучаются основные принципы работы алгоритма, включая его способность обрабатывать рёбра и циклы с отрицательным весом, а также его роль в планировании маршрутов в транспортных сетях. Кроме того, анализируются преимущества и ограничения алгоритма в задачах определения оптимального маршрута. Результаты исследования демонстрируют возможности применения алгоритма в реальных областях, таких как системы общественного транспорта, логистика и телекоммуникационные сети.

**Ключевые слова:** транспортные сети, кратчайший путь, алгоритм Форда-Беллмана, отрицательные циклы, графовое моделирование, оптимизация маршрутов, логистика, общественный транспорт, телекоммуникации.

**Annotation.** The management and optimization of transportation networks are closely related to solving shortest path problems in complex graphs. This article examines the practical application and efficiency of the Ford-Bellman algorithm in transportation systems. The study explores the fundamental principles of the algorithm, including its ability to handle edges and cycles with negative weights, as well as its role in route planning within transportation networks. Additionally, the advantages and limitations of the algorithm in determining optimal paths are analyzed. The results of the research demonstrate the algorithm's applicability in real-world domains such as public transport systems, logistics, and telecommunication networks.

**Keywords:** transportation networks, shortest path, Ford-Bellman algorithm, negative cycles, graph modeling, route optimization, logistics, public transport, telecommunication.

Transport tarmoqlari murakkab grafiklar shaklida ifodalanadi, bunda tugunlar — shaharlari yoki bekatlarni, qirralar esa ularning o‘zaro bog‘liqligini (masofa, vaqt yoki xarajat kabi) aks ettiradi. Ushbu tarmoqlarda eng qisqa yo‘lni aniqlash — transport tizimlarini samarali boshqarish va optimallashtirishning asosiy masalalaridan biridir.

Ammo, ushbu masalaning murakkabligi transport grafiklarida manfiy og‘irlikli qirralarning mavjudligi yoki katta hajmdagi ma’lumotlarni qayta ishlash ehtiyoji bilan oshadi. Masalan, transport tarmoqlarida uzoq masofali yo‘nalishlarni rejalashtirishda xarajatlarni kamaytirish va vaqtni tejash kabi talablar ustuvor hisoblanadi.

Bu muammo nafaqat jamoat transporti tizimlarini optimallashtirishda, balki logistika, ta’minot zanjirlari va telekommunikatsiya tarmoqlarida ham dolzarb hisoblanadi. Ford-Bellman algoritmi ushbu murakkabliklarni hal qilishda muhim vosita bo‘lib, uning asosiy vazifasi grafiklarda manfiy tsikllar mavjud bo‘lsa ham eng qisqa yo‘lni aniqlashdan iboratdir.

Mazkur maqolada ushbu algoritmning asosiy tamoyillari, amaliy qo‘llanilish imkoniyatlari va samaradorligi haqida fikr yuritiladi. Bu, ayniqsa, transport tarmoqlarini innovatsion boshqarish va optimallashtirish uchun muhim ahamiyatga ega.

Ford-Bellman algoritmi murakkab tarmoqlarda eng qisqa yo‘lni aniqlashda samarali bo‘lsa-da, uning cheklovlarini bartaraf etish va qo‘llanilish imkoniyatlarini kengaytirish uchun quyidagi yo‘nalishlarda rivojlantirish amalga oshirilishi mumkin:

### ***1. Parallel dasturlash texnologiyalari***

- **Muammo:** Ford-Bellman algoritmi barcha qirralarni ketma-ket tahlil qiladi, bu esa katta grafiklarda hisoblash vaqtini oshiradi.
- **Imkoniyat:** Parallel dasturlash yondashuvlarini qo‘llash orqali qirralar va tugunlar orasidagi hisob-kitoblarni bir vaqtda bajarish mumkin. Bu algoritmning katta grafiklarda ishlash tezligini sezilarli darajada oshiradi.

- **Amaliy tatbiq:** Zamonaviy superkompyuterlar va GPU yordamida algoritmni tezlashtirish.

## 2. *Gibrid algoritmlarni integratsiya qilish*

- **Muammo:** Ford-Bellman algoritmi katta grafiklarda samarali bo'lsa-da, kichik grafiklarda ba'zi hollarda Dijkstra kabi algoritmlardan sekinroq ishlaydi.
- **Imkoniyat:** Dijkstra va A\* algoritmlari bilan gibrid yondashuvni qo'llash orqali turli grafiklarda samaradorlikni oshirish mumkin. Bu yondashuv kichik va katta grafiklarda optimal ishlashni ta'minlaydi.

## 3. *Dinamik tarmoqlarda qo'llash*

- **Muammo:** Hozirgi algoritmlar statik grafiklar uchun mo'ljallangan, ya'ni grafikdagi tugunlar yoki qirralar o'zgarishida algoritmni qayta ishga tushirish talab etiladi.
- **Imkoniyat:** Dinamik grafiklar uchun algoritmni moslashtirish orqali tarmoqdagi real vaqtli o'zgarishlarga moslashish mumkin. Bu transport va telekommunikatsiya tarmoqlarini boshqarishda katta afzallik beradi.

## 4. *Sun'iy intellekt va mashinani o'qitish yondashuvlari*

- **Muammo:** Algoritm murakkab grafiklar uchun o'z-o'zidan optimallashtirilmagan.
- **Imkoniyat:** Sun'iy intellekt va mashinani o'qitish yondashuvlarini qo'shish orqali grafikni boshqarish va manfiy tsikllarni oldindan aniqlash imkoniyati yaratiladi. Neyron tarmoqlar yoki kuchaytirish orqali o'qitish usullari algoritmni murakkab holatlarda samarali ishlashga moslashishi mumkin.

## 5. *Ixtisoslashgan tarmoqlarda qo'llash*

- **Transport Tarmoqlari:** Marshrutlarni yanada optimallashtirish, masalan, real vaqtli transport tarmoqlarida tezkor yo'nalishlarni belgilash.
- **Telekommunikatsiya:** Ma'lumotlar oqimini samarali boshqarish va qirqimlarning to'g'ri taqsimlanishini ta'minlash.
- **Moliyaviy Tahlil:** Xavfli yo'nalishlarni avtomatik aniqlash va qaror qabul qilishni avtomatlashtirish.

## 6. *Energiya sarfini optimizatsiya qilish*

- **Muammo:** Algoritmning murakkabligi uni yuqori resurs talab qiladi.
- **Imkoniyat:** Algoritmni resursni kam talab qiladigan versiyalari orqali IoT (Internet of Things) va sensor tarmoqlari kabi kam quvvatli tizimlarda qo'llash.

Ford-Bellman algoritmi murakkab grafiklarda eng qisqa yo'lni aniqlash va manfiy og'irlikli qirralar mavjud bo'lgan tarmoqlarni boshqarish uchun samarali vosita hisoblanadi. Uning natijalari va samaradorligini quyidagi mezonlar asosida baholash mumkin:

### 1. Natijalar

- **Eng qisqa yo'lni topish.** Ford-Bellman algoritmi barcha tugunlar o'rtasidagi masofalarni aniqlashda optimal natijalar beradi. U musbat va manfiy og'irlikli qirralar bilan ishlash qobiliyatiga ega bo'lgan kam sonli algoritmlardan biridir.
- **Manfiy tsikllarni aniqlash.** Algoritm tarmoqdagi manfiy tsikllarni aniqlay oladi. Bu xususiyat, ayniqsa, transport yoki moliyaviy tarmoqlarda xavfli yo'nalishlarni boshqarishda foydali hisoblanadi.
- **Keng qo'llanilish imkoniyati.** Tadqiqotlar shuni ko'rsatadiki, algoritm transport tarmoqlari, jamoat transporti marshrutlari va logistika tizimlarini optimallashtirishda keng qo'llaniladi.

### 2. Samaradorlik

- **Hisoblash murakkabligi.** Algoritmning hisoblash murakkabligi  $O(|V| \times |E|)$  bo'lib, bu har bir qirra uchun barcha tugunlarni ketma-ket tahlil qilishni talab qiladi. Bu katta grafiklarda resurslarni talab qiluvchi jarayon bo'lishi mumkin.
- **Manfiy tsikllar mavjudligi.** Manfiy tsikllar algoritmi noto'g'ri yo'l natijalariga olib kelmasdan, bu tsikllarni samarali aniqlash imkonini beradi. Bu xususiyat boshqa algoritmlardan muhim farq qiladi.
- **Katta grafiklarda ishlash.** Algoritm katta grafiklarda samarali natijalar beradi, ammo ishlash tezligi Dijkstra kabi algoritmlardan sekinroq bo'lishi mumkin. Shu sababli, uning samaradorligini oshirish uchun parallel dasturlash texnologiyalarini qo'llash tavsiya etiladi.
- **Amaliy samaradorlik.** Algoritm tarmoqdagi barcha yo'llarni ketma-ket tahlil qilishni talab qiladi, bu esa kichik va o'rta hajmdagi grafiklarda samarali, katta grafiklarda esa hisoblash quvvatini oshirishni talab qiladi.

Masala: Transport tarmoqlarida eng qisqa yo'lni topish

#### *Masala Tavsifi*

Bir shaharda bir nechta bekatlar mavjud va ular orasidagi masofalar tasodifiy qirralar og'irliklari orqali ifodalangan. Bekatlar orasida manfiy og'irlikli yo'llar mavjud bo'lishi mumkin. Har bir bekatning eng qisqa yo'lini aniqlash va barcha bekatlar uchun optimal yo'nalishni hisoblash kerak. Shuningdek, manfiy tsikllar mavjud bo'lsa, bu haqida xabar berilishi lozim.

#### *Masalaning Kirish Ma'lumotlari*

1. **Tugunlar (bekatlar):** A, B, C, D, E.
2. **Qirralar (yo'llar):** Og'irliklar (masofa yoki vaqt) bilan berilgan. Masalan:

- $A \rightarrow B: 4$
- $A \rightarrow C: 2$
- $B \rightarrow C: -3$
- $B \rightarrow D: 5$
- $C \rightarrow D: 2$
- $D \rightarrow E: 1$
- $E \rightarrow B: -6$

### **Masalaning Chiqish Ma'lumotlari**

1. Har bir bekatning boshlang'ich tugundan eng qisqa masofasi.
2. Manfiy tsikllar mavjud bo'lsa, ularga oid xabar.

### **Algoritm**

Masalani yechish uchun **Ford-Bellman algoritmi** quyidagi qadamlar orqali amalga oshiriladi:

1. **Boshlang'ich tugundan masofani aniqlash:**
  - Barcha tugunlar uchun boshlang'ich masofa  $\infty$  (cheksiz) qilib o'rnatiladi, faqat boshlang'ich tugun uchun masofa 0 bo'ladi.
2. **Barcha qirralarni iteratsiya qilish:**
  - Har bir qirra bo'yicha barcha tugunlarning masofasi yangilanadi:
    - Agar  $\text{masofa}[u] + \text{og'irlilik}(u, v) < \text{masofa}[v]$  bo'lsa,  $\text{masofa}[v]$  yangilanadi.
3. **Manfiy tsikllarni tekshirish:**
  - Oxirgi iteratsiyadan so'ng, agar masofa yangilansa, manfiy tsikl mavjud.
4. **Natijani chiqarish:**
  - Har bir tugunning boshlang'ich tugundan masofasi.
  - Manfiy tsikl bor yoki yo'qligi.

### **Ford-Belman algoritmi C# kodida:**

```
using System;
```

```
using System.Collections.Generic;
```

```
class Program
```

```
{
```

```
    class Edge
```

```
    {
```

```
        public int Source, Destination, Weight;
```

```
        public Edge(int src, int dest, int weight)
```

```
        {
```

```
            Source = src;
```

```
    Destination = dest;  
    Weight = weight;  
  }  
}
```

```
static void BellmanFord(int vertices, int edges, List<Edge> edgeList, int src)  
{  
    int[] distance = new int[vertices];  
    for (int i = 0; i < vertices; i++)  
        distance[i] = int.MaxValue;  
    distance[src] = 0;  
  
    // Qadam 2: Masofalarni yangilash  
    for (int i = 1; i < vertices; i++)  
    {  
        foreach (var edge in edgeList)  
        {  
            if (distance[edge.Source] != int.MaxValue && distance[edge.Source] +  
edge.Weight < distance[edge.Destination])  
            {  
                distance[edge.Destination] = distance[edge.Source] + edge.Weight;  
            }  
        }  
    }  
  
    // Qadam 3: Manfiy tsikllarni tekshirish  
    foreach (var edge in edgeList)  
    {  
        if (distance[edge.Source] != int.MaxValue && distance[edge.Source] +  
edge.Weight < distance[edge.Destination])  
        {  
            Console.WriteLine("Manfiy tsikl mavjud!");  
            return;  
        }  
    }  
  
    // Natijalarni chiqarish  
    Console.WriteLine("Tugunlarga masofalar:");
```

```
for (int i = 0; i < vertices; i++)
    Console.WriteLine($"Tugun {i}: {distance[i]}");
}

static void Main()
{
    int vertices = 5; // Tugunlar soni
    int edges = 7; // Qirralar soni
    var edgeList = new List<Edge>
    {
        new Edge(0, 1, 4),
        new Edge(0, 2, 2),
        new Edge(1, 2, -3),
        new Edge(1, 3, 5),
        new Edge(2, 3, 2),
        new Edge(3, 4, 1),
        new Edge(4, 1, -6)
    };

    int src = 0; // Boshlang'ich tugun
    BellmanFord(vertices, edges, edgeList, src);
}
}
```

### Natija:

Agar yuqoridagi kodni ishga tushirsangiz:

1. **Masofa Natijalari:** Har bir tugun uchun eng qisqa yo‘l masofasi.
2. **Manfiy Tsikl:** Agar grafikda mavjud bo‘lsa, bu haqida xabar beradi.

Transport tarmoqlarida eng qisqa yo‘lni topish masalasi murakkab grafiklar va tarmoq modellashtirishning muhim qismi hisoblanadi. Ushbu maqolada Ford-Bellman algoritmining transport tizimlarida qo‘llanilishi tahlil qilindi. Algoritmning asosiy afzalliklari — manfiy og‘irlikli qirralar va tsikllar bilan ishlash qobiliyati, keng ko‘lamli grafiklarda qo‘llanilishi va aniqligi — ushbu muammoni hal qilishda uning ahamiyatini ko‘rsatdi.

Shuningdek, algoritmnining transport tarmoqlari, logistika, va telekommunikatsiya tizimlarini optimallashtirishdagi o‘rni alohida ta’kidlandi. Ammo hisoblash murakkabligi kabi ba’zi cheklovlarni bartaraf etish uchun zamonaviy texnologiyalar, jumladan, parallel dasturlash yondashuvlaridan foydalanish tavsiya etiladi.

Ford-Bellman algoritmi transport tarmoqlarini samarali boshqarish, xarajatlarni kamaytirish va tarmoq samaradorligini oshirishda muhim vosita bo'lib qolmoqda. Kelajakda algoritmi yanada takomillashtirish va amaliy qo'llanilish imkoniyatlarini kengaytirish zarur.

### **Foydalanilgan Adabiyotlar**

1. Bellman, R. (1958). On a Routing Problem. *Quarterly of Applied Mathematics*.
2. Ford, L. R. (1956). *Network Flow Theory*. Princeton University Press.
3. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms*. MIT Press.
4. Ahuja, R. K., Magnanti, T. L., & Orlin, J. B. (1993). *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall.
5. Gibbons, A. (1985). *Algorithmic Graph Theory*. Cambridge University Press.
6. Sedgewick, R., & Wayne, K. (2011). *Algorithms (4th Edition)*. Addison-Wesley.
7. Tarjan, R. E. (1983). *Data Structures and Network Algorithms*. Society for Industrial and Applied Mathematics.
8. Dijkstra, E. W. (1959). A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*.
9. Russel, S., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach*. Pearson.
10. Newman, M. (2010). *Networks: An Introduction*. Oxford University Press.
11. Marcin Jamro. *C# Data Structures and Algorithms*. Second Edition. Published by Packt Publishing Ltd., in Birmingham, UK. 2024. – 349 p.
12. Дж.Эриксон. *АЛГОРИТМЫ.* – М.: " ДМК Пресс ", 2023. – 528 с.
13. Hemant Jain. *Data Structures & Algorithms using Kotlin*. Second Edition. in India. 2022. – 572 p.
14. Н. А. Тюкачев, В. Г. Хлебостроев. *C#. Алгоритмы и структуры данных: учебное пособие для СПО.* – СПб.: Лань, 2021. – 232 с.
15. Mykel J. Kochenderfer. Tim A. Wheeler. *Algorithms for Optimization*. Published by The MIT Press., in London, England. 2019. – 500 p.
16. Рафгарден Тим. *Совершенный алгоритм. Графовые алгоритмы и структуры данных.* – СПб.: Питер, 2019. - 256 с.
17. Ахо Альфред В., Ульман Джеффри Д., Хопкрофт Джон Э. *Структуры данных и алгоритмы.* – М.: Вильямс, 2018. – 400 с.
18. Дж.Хайнеман, Г.Поллис, С.Стэнли. *Алгоритмы. Справочник с примерами на C, C++, Java и Python, 2-е изд.: Пер. с англ.* — СПб.: ООО "Альфа-книга", 2017. — 432 с.



19. Farmonov, S., & Nazirov, A. (2023). C# DASTURLASH TILIDA GRAY KODI BILAN ISHLASH. В CENTRAL ASIAN JOURNAL OF EDUCATION AND INNOVATION (Т. 2, Выпуск 12, сс. 71–74). Zenodo.
20. Farmonov, S., & Toirov, S. (2023). NETDA DASTURLASHNING ZAMONAVIY TEXNOLOGIYALARINI O'RGANISH. Theoretical aspects in the formation of pedagogical sciences, 2(22), 90-96
21. Raxmonjonovich, F. S. (2023). Array ma'lumotlar tizimini talabalarga o'qitishda Blockchain metodidan foydalanish. Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari, 2(2), 541-547.
22. Raxmonjonovich, F. S. (2023). Dasturlashda interfeyslardan foydalanishning ahamiyati. Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari, 2(2), 425-429.
23. Raxmonjonovich, F. S. (2023). Dasturlashda obyektga yo'naltirilgan dasturlashning ahamiyati. Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari, 2(2), 434-438.
24. Raxmonjonovich, F. S. (2023). Dasturlash tillarida fayllar bilan ishlash mavzusini Blended Learning metodi yordamida o'qitish. Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari, 2(2), 464-469.
25. Raxmonjonovich, F. S. (2023). DASTURLASHDA ISTISNOLARNING AHAMIYATI. Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari, 2(2), 475-481.
26. Raxmonjonovich, F. S. (2023). Dasturlashda abstraksiyaning o'rni. Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari, 2(2), 482-486.
27. Farmonov Sherzodbek Raxmonjonovich, & Rustamova Humoraxon Sultonbek qizi. (2024). C# DASTURLASH TILIDA TO'PLAMLAR BILAN ISHLASH. Ta'lim Innovatsiyasi Va Integratsiyasi, 11(10), 210–214. Retrieved from <http://web-journal.ru/index.php/ilmiy/article/view/2480>.
28. Raxmonjonovich, F. S., & Ravshanbek o'g'li, A. A. (2023). Zamonaviy dasturlash tillarining qiyosiy tahlili. Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari, 2(2), 430-433.
29. Farmonov, S., & Rasuljonova, Z. (2024). OB'EKTGA YO'NALTIRILGAN DASTURLASH ZAMONAVIY DASTURLASHNING ASOSI SIFATIDA. Центральноеазиатский журнал образования и инноваций, 3(1), 83-86.