



BIOLOGIYA VA GENETIKAGA OID MASALALARNI TARMOQLANISH VA BOG'LASH ALGORITMLARI YORDAMIDA YECHISH

Farmonov Sherzodbek Raxmonjonovich

Farg'ona davlat universiteti amaliy matematika va informatika

kafedrası katta o'qituvchisi, farmonovsh@gmail.com

Omonjonova Mavludaxon Kamoliddin qizi

Farg'ona davlat universiteti 2-kurs talabasi

ozodbekomonjonov203@gmail.com

<https://doi.org/10.5281/zenodo.14249159>

ARTICLE INFO

Received: 19 th November 2024

Accepted: 20th November 2024

Published: 29th November 2024

KEYWORDS

Tarmoqlanish algoritmi,
Bog'lash algoritmi, C#
dasturlash, Dijkstra algoritmi,
Kruskal algoritmi, Graf
nazariyasi, Minimum spanning
tree (MST), Qisqa yo'l masalasi,
Tarmoq optimallashtirish,
Resurslarni taqsimlash,
Telekommunikatsiya
tarmoqlari, Sun'iy intellekt,
Tarmoq yuklamasini
boshqarish, Samaradorlik
optimallashtirish, Algoritmlar,
Ma'lumotlar oqimi.

ABSTRACT

Ushbu maqolada tarmoqlanish va bog'lash algoritmlari C# dasturlash tilida qanday amalga oshirilishi va ularning amaliy qo'llanilishi ko'rib chiqiladi. Tarmoqlanish algoritmlari, masalan, Dijkstra algoritmi yordamida tizimlar o'rtasidagi eng qisqa yo'lni topishda ishlatiladi, bog'lash algoritmlari esa, Kruskal algoritmi kabi metodlar yordamida tizimlar o'rtasidagi eng kam xarajatli bog'lanishlarni aniqlaydi.

Kirish: Tarmoqlanish va bog'lash algoritmlari (minimal bog'lanish daraxti algoritmlari) graf nazariyasida muhim tushunchalardir. Ular grafdagi nuqtalar (tugunlar) o'rtasida minimal og'irlikka ega bo'lgan bog'lanish (bog'lanish daraxti)ni topishda qo'llaniladi, ma'lumotlarni tahlil qilish va ulardan foydalanish uchun ishlatiladi. Graf — bu tugunlar (vertices) va ularni bog'laydigan qirralardan (edges) iborat bo'lgan matematik modeldir.

Tarmoqlanish va bog'lash algoritmlari kompyuter ilmida ma'lumotlarni tashkil etish va ularga kirish usullarini belgilovchi muhim tushunchalardir. Ular asosan ma'lumotlar tuzilmalari va ularni boshqarish uchun ishlatiladi.

Bog'lash algoritmlari esa ma'lumotlar tuzilmalarini bir-biriga bog'lash yoki ularga murojaat qilish jarayonlarini boshqaradi. Ular ko'pincha ma'lumotlarni saqlash va ularga kirishni optimallashtirishda qo'llaniladi.

Tarmoqlanish va bog'lash algoritmlari ko'plab sohalarda qo'llaniladi. Ularning qo'llanilish sohasidan ba'zilar quyidagilar:

1. Kompyuter tarmoqlari

Tarmoqlanish algoritmlari ma'lumotlarni uzatish uchun eng optimal yo'llarni aniqlashda, tarmoq topologiyasini boshqarishda va tarmoqning samaradorligini oshirishda qo'llaniladi.

2. Ma'lumotlar bazalari:

Bog'lash algoritmlari ma'lumotlar bazasida ma'lumotlarni izlash va bog'lash, shuningdek, ma'lumotlar to'plamlarini birlashtirish uchun ishlatiladi.

3. Sun'iy intellekt va mashinani o'rganish:

Tarmoqlanish algoritmlari sun'iy neyron tarmoqlarni yaratishda va ularni o'qitishda qo'llaniladi. Bog'lash algoritmlari esa ma'lumotlarni tasniflash va klasterlashda yordam beradi.

4. Grafika va kompyuter grafikasida:

Tarmoqlanish algoritmlari grafik ob'ektlar orasidagi bog'lanishlarni aniqlash va ularni boshqarishda ishlatiladi.

5. Biologiya:

Genomika va biologik tahlillarda bog'lash algoritmlari genlar va proteinlar o'rtasidagi bog'lanishlarni aniqlashda qo'llaniladi.

6. Iqtisodiyot va moliya:

Tarmoqlanish algoritmlaridan iqtisodiy modellarda, masalan, bozorlar orasidagi bog'lanishlarni aniqlashda foydalaniladi.

Biologiya va genetika sohasida tarmoqlanish va bog'lash algoritmlari bir qator muhim masalalarni hal qilishda qo'llaniladi. Quyida ularning asosiy qo'llanilish joylari keltirilgan:

1. Genomni tahlil qilish:

Tarmoqlanish algoritmlari genomni tahlil qilishda, ya'ni genetik ma'lumotlarni o'rganishda va ularni tasniflashda qo'llaniladi. Bu algoritmlar genomning tuzilishini aniqlash, genlar o'rtasidagi bog'lanishlarni o'rganish va genetik variantlarni aniqlashda yordam beradi.

2. Genetik aloqalar va filogenetika:

Bog'lash algoritmlari turli organizmlar orasidagi genetik aloqalarni aniqlashda, evolyutsion daraxtlarni qurishda va turli xil organizmlar o'rtasidagi munosabatlarni o'rganishda qo'llaniladi. Bu algoritmlar yordamida turli xil genlar yoki proteinlar o'rtasidagi evolyutsion bog'lanishlar aniqlanadi.

3. Proteomika:

Proteinlar o'rtasidagi bog'lanishlarni va ularning funksiyalarini aniqlashda bog'lash algoritmlari qo'llaniladi. Bu, masalan, protein-protein o'zaro ta'sirlarini o'rganishda yoki proteinlarning funksional tarmoqlarini aniqlashda muhimdir.

4. Ma'lumotlar integratsiyasi:

Tarmoqlanish va bog'lash algoritmlari turli manbalardan olingan genetik ma'lumotlarni birlashtirishda va integratsiyalashda ishlatiladi. Bu, masalan, ommaviy ma'lumotlar bazalaridan (masalan, GenBank yoki Ensembl) ma'lumotlarni olishda va ularni birlashtirishda qo'llaniladi.

5. Genetik kasalliklarni aniqlash:

Genetik kasalliklarni aniqlash va ularning sabablarini tushunishda bog'lash algoritmlari yordamida genetik variantlar va kasalliklar o'rtasidagi bog'lanishlar o'rganiladi. Bu, shuningdek, individual genetik xususiyatlar bilan kasalliklar o'rtasidagi munosabatlarni aniqlashda ham qo'llaniladi.

Biologiya va genetikada tarmoqlanish va bog'lash algoritmlaridan foydalanish ko'plab qiziqarli masalalarni yechishda qo'llaniladi. Quyida biologiya va genetikaga doir masala keltirilgan:

Masala:

Bir biologik tadqiqotda, 5 ta turli populyatsiyadan olingan namunalar mavjud: A, B, C, D, E. Har bir populyatsiya orasida genetik aloqalar kuchi (masalan, genetik o'xshashlik koeffitsienti) berilgan. Ushbu ma'lumotlar asosida populyatsiyalar o'rtasida minimal bog'lanish daraxtini toping.

Bu daraxt orqali genetik aloqalarni vizualizatsiya qilish va tahlil qilish mumkin.

Berilgan genetik o'xshashlik koeffitsientlari:

- A - B: 0.9
- A - C: 0.8
- B - C: 0.6
- B - D: 0.7
- C - D: 0.5
- C - E: 0.4
- D - E: 0.3

Berilgan masalani tarmoqlanish va bog'lash algoritmlari yordamida yechamiz:

Yechim:

1. Grafik tuziladi: Har bir populyatsiya nuqtasi sifatida qabul qilinadi va berilgan o'xshashlik koeffitsientlari qirralar sifatida ko'rsatiladi.
2. Minimal bog'lanish daraxti topiladi: Prim yoki Kruskal algoritmlaridan biri tanlanadi va berilgan ma'lumotlar asosida minimal bog'lanish daraxti hisoblanadi.

✓ Prim algoritmi yordamida yechim:

Berilgan masalani prim algoritmi yordamida yechish jarayoni quyidagi ketma-ketlikda bajariladi:

Prim Algoritmi yordamida minimal bog'lanish daraxtini topish

Boshlang'ich ma'lumotlar

Berilgan genetik o'xshashlik koeffitsientlari:

- A - B: 0.9
- A - C: 0.8
- B - C: 0.6
- B - D: 0.7
- C - D: 0.5
- C - E: 0.4
- D - E: 0.3

Ketma-ketlik

1. Boshlang'ich nuqta tanlanadi:

A nuqtasi boshlang'ich nuqta sifatida tanlanadi.

Bog'lanish daraxti (MST) boshlanishi uchun bo'sh to'plam yaratiladi.

2. Qirralarni aniqlash:

A dan chiqadigan barcha qirralar aniqlanadi:

A - B (0.9)

A - C (0.8)

3. Eng yuqori o'xshashlik tanlanadi:

A dan chiqadigan eng yuqori o'xshashlikka ega qirra tanlanadi:

Tanlangan qirra: A - B (0.9)

MST ga qo'shiladi: $MST = \{A - B\}$

4. Yangi nuqtalarni qo'shish:

Endi B nuqtasidan chiqadigan qirralar aniqlanadi:

B - C (0.6)

B - D (0.7)

Qo'shilgan qirralar: $\{A - B, B - C, B - D\}$

5. Eng yuqori o'xshashlik tanlanadi:

Hozirgi eng yuqori o'xshashlik tanlanadi:

Tanlangan qirra: B - D (0.7)

MST ga qo'shiladi: $MST = \{A - B, B - D\}$

6. Yangi nuqtalarni qo'shish:

Endi D nuqtasidan chiqadigan qirralar aniqlanadi:

D - E (0.3)

Qo'shilgan qirralar: $\{A - B, B - D, D - E\}$

7. Eng yuqori o'xshashlik tanlanadi:

Hozirgi eng yuqori o'xshashlik tanlanadi:

Tanlangan qirra: D - E (0.3)

MST ga qo'shiladi: $MST = \{A - B, B - D, D - E\}$

8. Yana yangi nuqtalarni qo'shish:

E nuqtasidan chiqadigan qirralar aniqlanadi:

E dan boshqa nuqtalar bilan bog'lanish yo'q.

9. Tugallash:

Hozirda barcha nuqtalar (A, B, D, E) MST ga kiritilgan, shuning uchun algoritm tugaydi.

Natija:

Minimal bog'lanish daraxti (MST) quyidagi qirralardan iborat bo'ladi:

A - B (0.9)

B - D (0.7)

D - E (0.3).

Quyida, yuqoridagi masalaning C# dasturlash tilida prim algoritmini amalga oshiruvchi dastur keltirilgan. Bu kodlar minimal bog'lanish daraxtini topishda yordam beradi.

➤ Prim Algoritmi

```
using System;
```

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

```
class Program
```

```
{  
    static int V = 5; // Grafdagi nuqtalar soni
```

```
    static int MinKey(int[] key, bool[] mstSet)  
    {  
        int min = int.MaxValue, minIndex = -1;
```

```
        for (int v = 0; v < V; v++)  
        {  
            if (!mstSet[v] && key[v] < min)  
            {  
                min = key[v];  
                minIndex = v;  
            }  
        }  
    }
```

```
        return minIndex;
```

```
}
```

```
static void PrimMST(int[,] graph)
```

```
{  
    int[] parent = new int[V];  
    int[] key = new int[V];
```

```
bool[] mstSet = new bool[V];

for (int i = 0; i < V; i++)
{
    key[i] = int.MaxValue;
    mstSet[i] = false;
}

key[0] = 0; // Birinchi nuqtani boshlang'ich qilib olamiz
parent[0] = -1; // Birinchi nuqtada ota yo'q

for (int count = 0; count < V - 1; count++)
{
    int u = MinKey(key, mstSet);
    mstSet[u] = true;

    for (int v = 0; v < V; v++)
    {
        if (graph[u, v] != 0 && !mstSet[v] && graph[u, v] < key[v])
        {
            parent[v] = u;
            key[v] = graph[u, v];
        }
    }
}
```

```
Console.WriteLine("Minimal bog'lanish daraxti:");
for (int i = 1; i < V; i++)
{
    Console.WriteLine($"{parent[i]} - {i} t {graph[i, parent[i]]}");
}
}
```

```
static void Main()
{
    int[,] graph = new int[,]
    {
        { 0, 2, 0, 6, 0 },
        { 2, 0, 3, 8, 5 },
        { 0, 3, 0, 0, 7 },
        { 6, 8, 0, 0, 9 },
        { 0, 5, 7, 9, 0 }
    };
}
```

```
PrimMST(graph);
}
```

Xulosa

Bu minimal bog'lanish daraxti orqali turli populyatsiyalar o'rtasidagi genetik aloqalarni vizualizatsiya qilish va tahlil qilishi mumkin. Bu tadqiqotlar natijasida, masalan, evolyutsion aloqalar, genetik diversifikatsiya va boshqa biologik jarayonlar haqida muhim

ma'lumotlar olish mumkin. Shuningdek, tarmoqlanish va bog'lash algoritmlari biologik tadqiqotlarda genetik aloqalarni o'rganishda foydali vosita bo'lishi mumkin.

Foydalanilgan adabiyotlar ro'yhati

1. "Graph Algorithms", Robert S. W. McEliece, 1998
2. "Network Flow and Monotropic Optimization", A. K. Jain, R. K. Ahuja, and J. B. Orlin, 1993
3. "Algorithms on Trees and Graphs", Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, 2009
4. "Graph Theory with Applications", Bondy, J.A., and Murty, U.S.R., 1976
5. "Introduction to Graph Theory", Richard J. Wilson, 2010
6. Farmonov, S., & Jo'rayeva, M. (2023, December). DASTURLASHDA POLIMORFIZMNING AHAMIYATI. In Международная конференция академических наук (Vol. 2, No. 13, pp. 5-8).
7. Farmonov, S., & Karimova, M. (2023). MODERN METHODS TO DEVELOP MATHEMATICAL THINKING IN SCHOOLCHILDREN. Бюллетень педагогов нового Узбекистана, 1(6 Part 2), 28-38.
8. Farmonov, S., & Rahmatjonov, M. (2023). FLUTTER YORDAMIDA PLATFORMALARARO KUTUBXONANI ISHLAB CHIQISH TEXNOLOGIYASI. Центральноеазиатский журнал образования и инноваций, 2(6 Part 2), 124-127.
9. 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).
10. Tojiyev, T., Boynazarov, A., & Farmonov, S. (2022). PHARMACOKINETICS IS A DESCRIPTION OF DRUGS AND THEIR BEHAVIOR IN THE HUMAN BODY BY BUILDING A MATHEMATICAL MODEL. Евразийский журнал медицинских и естественных наук, 2(13), 146-149.
11. 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.
12. Raxmonjonovich, F. S. (2023). Dasturlashda interfeyslardan foydalanishning ahamiyati. Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari, 2(2), 425-429.
13. Raxmonjonovich, F. S. (2023). Dasturlashda obyektga yo'naltirilgan dasturlashning ahamiyati. Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari, 2(2), 434-438.
14. 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.
15. Raxmonjonovich, F. S. (2023). DASTURLASHDA ISTISNOLARNING AHAMIYATI. Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari, 2(2), 475-481.
16. Raxmonjonovich, F. S. (2023). Dasturlashda abstraksiyaning o'rni. Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari, 2(2), 482-486.
17. 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.
18. Raxmonjonovich, F. S. (2023). C# dasturlash tilida fayl operatsiyalari qo'llashning qulayliklari haqida. Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari, 2(2), 439-446.
19. Raxmonjonovich, F. S. (2023). C# tilida ArrayList bilan ishlashning afzalliklari. Yangi O'zbekiston taraqqiyotida tadqiqotlarni o'rni va rivojlanish omillari, 2(2), 470-474.
20. 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>.

21. 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.

22. Farmonov, S. R., & qizi Xomidova, M. A. (2024). C# VA JAVA DASTURLASH TILLARIDA FAYLLAR BILAN ISHLASHNING TURLI USULLARINING SAMARADORLIGI HAQIDA. *Zamonaviy fan va ta'lim yangiliklari xalqaro ilmiy jurnal*, 1(9), 45-51.

23. Raxmonjonovich, F. S. (2024). C# VA MASHINA TILI. *Ta'lim innovatsiyasi va integratsiyasi*, 12(1), 59-62.

