

TRANSPORT YO‘LLARIDA MINIMAL MASOFALARNI HISOBLASHDA DEYKSTRI ALGORITMINING AHAMIYATI

Farmonov Sherzodbek Raxmonjonovich

*Farg‘ona davlat universiteti amaliy matematika va
informatika kafedrasida katta o‘qituvchisi*

farmonovsh@gmail.com

Abduvaxobova Musilmaxon Baxtiyor qizi

Farg‘ona davlat universiteti talabasi

abduvaxobovazaytuna@gmail.com

Annotatsiya: Ushbu maqolada matematik algoritmlar va hisoblash usullarining dastur tizimlarining samaradorligini oshirishdagi roli tahlil qilinadi. Algoritmik yondashuvlar, jumladan, optimallashtirish algoritmlari, raqamli hisoblash usullari, hisoblash tezligini oshirish va parallel hisoblash texnikalari haqida batafsil ma'lumot beriladi. Shuningdek, maqolada matematik modellarni amaliy dasturlashga joriy etish, hisoblash jarayonlarini takomillashtirish va tizimlarning aniqligini yaxshilash imkoniyatlari muhokama qilinadi. Maqola dasturchilarga o'z dasturlarining samaradorligini oshirish uchun matematik algoritmlardan qanday foydalanishni, ularning imkoniyatlari va chegaralarini tushunishga yordam beradi. Matematika va algoritmlar orqali hisoblash samaradorligini optimallashtirishga qaratilgan turli usullar, ularning dasturlash jarayonidagi o'rni va rivojlanish tendensiyalari ko'rib chiqiladi.

Kalit so‘zlar: Algoritmlar, matematik hisoblash, dastur samaradorligi, optimallashtirish algoritmlari, raqamli usullar, parallel hisoblash, hisoblash aniqligi, hisoblash tezligi, matematik modellar, dastur optimizatsiyasi, matematik yondashuvlar, dasturlash texnikalari, tizim ishlashi, hisoblash texnologiyalari, tezlik va aniqlikni oshirish, amaliy dasturlash, tizimlar samaradorligi.

Annotation: This article explores the role of mathematical algorithms and computational methods in improving the efficiency of software systems. It provides a detailed analysis of algorithmic approaches, including optimization algorithms, numerical methods, techniques to enhance computational speed, and parallel computing. The article also discusses the integration of mathematical models into practical programming, ways to improve computational processes, and possibilities to enhance the accuracy of systems. It helps developers understand how to apply mathematical algorithms to improve the efficiency of their programs and systems, while examining the potential and limitations of these algorithms. Different methods of optimizing computational performance through mathematics and algorithms, their role in the development process, and emerging trends are explored.

Keywords: Algorithms, mathematical computation, software efficiency, optimization algorithms, numerical methods, parallel computing, computational accuracy, computational speed, mathematical models, program optimization, mathematical approaches, programming techniques, system performance, computational technologies, speed and accuracy enhancement, practical programming, system efficiency.

Аннотация : В этой статье рассматривается роль математических алгоритмов и методов вычислений в повышении эффективности программных систем. Приводится подробный анализ алгоритмических подходов, включая алгоритмы оптимизации, численные методы, способы повышения вычислительной скорости и технологии параллельных вычислений. Также рассматривается внедрение математических моделей в практическое программирование, улучшение вычислительных процессов и возможности повышения точности систем. Статья помогает разработчикам понять, как применить математические алгоритмы для повышения

эффективности своих программ и систем, а также анализирует потенциал и ограничения этих алгоритмов.

Ключевые слова: Алгоритмы, математические вычисления, эффективность программного обеспечения, алгоритмы оптимизации, численные методы, параллельные вычисления, вычислительная точность, вычислительная скорость, математические модели, оптимизация программ, математические подходы, техники программирования,

Algoritmlar va matematik hisoblash inson tafakkuri va texnologiyalar rivojining ajralmas qismiga aylangan. Bugungi kunning eng mashhur matematiklaridan biri, Jorj Poliyaning soʻzlari bilan aytganda, “Har qanday muammo oʻz yechimini izlash san’ati orqali topiladi”. Dasturlash va hisoblash sohasida bu san’at algoritmik yondashuvlar orqali namoyon boʻladi. Algoritmlar nafaqat matematik modellarni tushunish, balki real hayotdagi masalalarni yechishda samarali vosita sifatida ham xizmat qiladi.

Axborot texnologiyalari dunyosi kundan-kunga rivojlanib bormoqda va bu jarayonda matematik algoritmlarning ahamiyati beqiyos. Dasturlashning rivojlanishi faqatgina kod yozishni emas, balki murakkab tizimlarni boshqarish, ularga yuqori aniqlik bilan tezkor yechimlar topishni ham oʻz ichiga oladi. Masalan, Donald Knutning “The Art of Computer Programming” kitobida algoritmlar bilimning ustun nuqtasi sifatida taʼriflanadi va ularning tasnifidan tortib, samaradorlikni oshirish yoʻllarigacha batafsil tushuntirilgan.

Algoritmlar inson faoliyatining turli sohalarda chuqur ildiz otgan. Transport tarmoqlaridan tortib, telekommunikatsiya tizimlarigacha, bank operatsiyalaridan sunʼiy intellekt texnologiyalarigacha barcha sohalarda matematik modellar yordamida masalalar yechim topmoqda. Bugungi tezkor dunyoda matematik asosga ega boʻlgan algoritmlarsiz samaradorlikni taʼminlash, tezkor yechimlar

ishlab chiqishning deyarli imkoni yo'q. Shu sababli, algoritmlar va matematik hisoblash sohasida yangiliklarni izlash va ularni amaliyotga joriy etish dolzarb masala bo'lib qolmoqda.

Asosiy Qism:

Matematik algoritmlar va hisoblash usullari zamonaviy texnologiyalarning asosini tashkil etadi. Lekin ulardan foydalanish faqat hisoblash tezligini oshirish yoki ma'lumotni qayta ishlashda qo'llaniladigan oddiy vosita sifatida cheklanishi kerakmi? Aslida, algoritmlar murakkab tizimlarni boshqarish va hayotimizning turli jabhalarida strategik qarorlar qabul qilishda asosiy o'rin tutadi. Masalan, transport tarmoqlaridagi yo'l-yo'riqni optimallashtirish yoki tibbiyotda kasalliklarning oldini olish uchun sun'iy intellekt yordamida algoritmik modellarni qo'llash bizni o'z ishimizni yanada samarali qilishga undaydi.

Amaliy dasturlashda matematik algoritmlar ko'pincha intuitiv qarorlarni qo'llab-quvvatlovchi vosita sifatida ishlatiladi. Masalan, tibbiyotda diagnostik jarayonlarda yoki moliya sohasida xavf-xatarlarni bashorat qilishda algoritmlar inson qarorlarini qanday boyitadi? Sun'iy intellekt va mashina o'qitish algoritmlari bu jarayonlarni avtomatlashtirsa ham, ular ma'lumotlarga tayanadi. Bu yerda savol tug'iladi: ma'lumotlarning sifatiga va hajmiga qanchalik bog'liqmiz? Algoritmlar ma'lumotlar to'g'ri bo'lmaganida ham samarali ishlay oladimi?

Matematik modellarni real hayotga tatbiq etishda yana bir muhim jihat – bu algoritmlarning murakkabligi va ulardan foydalanishning qulayligi. Agar algoritmlar juda murakkab bo'lsa, ular bilan ishlashni oddiy dasturchilar uchun qanday osonlashtirish mumkin? Shu nuqtai nazardan, vizual dasturlash muhitlari va foydalanuvchilarga qulay interfeyslar yaratilishi tobora dolzarb bo'lib bormoqda. Zamonaviy texnologiyalar algoritmlarni murakkablikdan qutqarib, ularni turli sohalar uchun moslashtirish imkonini beradimi?

Masala:

Transport tarmog'ida bir shaharni boshqasiga bog'lovchi eng qisqa yo'lni topish kerak. Har bir yo'lning masofasi berilgan va shaharlardan qaysi biriga qanday yo'nalishda borish mumkinligi grafik ko'rinishida ifodalangan. Eng qisqa yo'lni aniqlash uchun **Dijkstra algoritmini** qo'llang.

Kod (C#)

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

class Program
{
    static void Main(string[] args)
    {
        // Grafni yaratish (shaharlar va ularni bog'lovchi yo'llar)
        Dictionary<string, Dictionary<string, int>> graph = new Dictionary<string,
Dictionary<string, int>>
        {
            { "A", new Dictionary<string, int> { { "B", 1 }, { "C", 4 } } },
            { "B", new Dictionary<string, int> { { "C", 2 }, { "D", 6 } } },
            { "C", new Dictionary<string, int> { { "D", 3 } } },
            { "D", new Dictionary<string, int>() }
        };

        // Boshlang'ich va maqsad shaharlar
        string start = "A";
        string destination = "D";
```

```
// Dijkstra algoritmini ishlatish
var result = DijkstraAlgorithm(graph, start, destination);

// Natijani chiqarish
if (result != null)
{
    Console.WriteLine("Eng qisqa yo'l:");
    Console.WriteLine(string.Join(" -> ", result.Path));
    Console.WriteLine($"Umumiy masofa: {result.Distance}");
}
else
{
    Console.WriteLine("Yo'l topilmadi.");
}
}

public static Result DijkstraAlgorithm(Dictionary<string, Dictionary<string,
int>> graph, string start, string destination)
{
    var distances = new Dictionary<string, int>();
    var previous = new Dictionary<string, string>();
    var unvisited = new HashSet<string>(graph.Keys);

    foreach (var node in graph.Keys)
    {
        distances[node] = int.MaxValue; // Boshlanishida barcha masofalar cheksiz
    }
    distances[start] = 0;
```

```
while (unvisited.Count > 0)
{
    string current = null;

    // Eng yaqin shaharni topish
    foreach (var node in unvisited)
    {
        if (current == null || distances[node] < distances[current])
        {
            current = node;
        }
    }

    if (current == destination)
    {
        var path = new List<string>();
        while (current != null)
        {
            path.Insert(0, current);
            previous.TryGetValue(current, out current);
        }

        return new Result
        {
            Path = path,
            Distance = distances[destination]
        };
    }
}
```

```
    }

    unvisited.Remove(current);

    foreach (var neighbor in graph[current])
    {
        int alt = distances[current] + neighbor.Value;
        if (alt < distances[neighbor.Key])
        {
            distances[neighbor.Key] = alt;
            previous[neighbor.Key] = current;
        }
    }
}

return null; // Yo'l topilmadi
}

public class Result
{
    public List<string> Path { get; set; }
    public int Distance { get; set; }
}
}
```

Izoh:

1. Graf tuzilishi: Graf shaharlardan (A, B, C, D) iborat va har bir shahar boshqa shaharga o'tish yo'llariga ega. Har bir yo'lning masofasi berilgan.

2. Dijkstra algoritmi: Har bir shahar uchun minimal masofani topish uchun ishlatiladi.

Natija: Eng qisqa yo'lni va umumiy masofani konsolda chiqaradi.

Misol Natijasi:

Agar yuqoridagi grafik ma'lumotlari ishlatilsa:

Eng qisqa yo'l:

A -> B -> C -> D

Umumiy masofa: 6

Zamonaviy texnologiyalarning murakkab muammolariga yechim izlashda matematik algoritmlar va hisoblash usullarining o'rni beqiyosdir. Ular faqatgina hisoblash va ma'lumotlarni qayta ishlash vositasi sifatida emas, balki strategik qarorlar qabul qilish, resurslarni boshqarish va kelajakda yuzaga keladigan ehtimoliy holatlarni tahlil qilishda muhim rol o'ynaydi. Ushbu maqola davomida algoritmlarning samaradorligi, optimallashtirish imkoniyatlari va ularning amaliyotga tatbiq qilinishidagi asosiy jihatlari ko'rib chiqildi, ammo bu mavzu yanada kengroq tahlilni talab qiladi.

Adabiyotlar

1. Knuth, Donald E. *The Art of Computer Programming, Volume 1: Fundamental Algorithms*. Addison-Wesley, 1997.
2. Cormen, Thomas H., Leiserson, Charles E., Rivest, Ronald L., Stein, Clifford. *Introduction to Algorithms*. MIT Press, 2009.
3. Hemant Jain. *Data Structures & Algorithms using Kotlin*. Second Edition. in India. 2022. – 572 p.

4. Н. А. Тюкачев, В. Г. Хлебостроев. С#. Алгоритмы и структуры данных: учебное пособие для СПО. – СПб.: Лань, 2021. – 232 с.
5. Mykel J. Kochenderfer. Tim A. Wheeler. Algorithms for Optimization. Published by The MIT Press., in London, England. 2019. – 500 p.
6. Рафгарден Тим. Совершенный алгоритм. Графовые алгоритмы и структуры данных. – СПб.: Питер, 2019. - 256 с.
7. [Ахо Альфред В., Ульман Джеффри Д., Хопкрофт Джон Э.](#) Структуры данных и алгоритмы. – М.: [Вильямс](#), 2018. – 400 с.
8. 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.
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). Zenodo.
10. Farmonov, S., & Toirov, S. (2023). NETDA DASTURLASHNING ZAMONAVIY TEXNOLOGIYALARINI O'RGANISH. *Theoretical aspects in the formation of pedagogical sciences*, 2(22), 90-96
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.