



JOHNSON ALGORITMI

Onarkulov Maksadjon Karimberdiyevich

*Farg'ona davlat univarsiteti Amaliy matematika va
informatika kafedrası dotsenti (PhD)*

maxmaqsad@gmail.com

Sahobiddinov Adhamjon Vohidjon o'g'li

Farg'ona davlat universiteti 2-bosqich talabasi

sahobiddinovadhamjon@gmail.com

Anatatsiya:

Johnson algoritmi barcha juftlik eng qisqa yo'llar (All-Pairs Shortest Path) masalasini hal qilish uchun mo'ljallangan samarali algoritm hisoblanadi. U 1977-yilda Donald B. Johnson tomonidan ishlab chiqilgan bo'lib, siyrak (sparse) graflarda Dijkstra va Bellman-Ford algoritmilarining kombinatsiyasini ishlatadi. Johnson algoritmi manfiy qirrali graflar bilan ishlash imkoniyatiga ega bo'lib, Floyd-Warshall algoritmiga nisbatan ko'plab hollarda yuqori samaradorlik ko'rsatadi. Algoritm avval Bellman-Ford yordamida manfiy tsikllarni aniqlaydi va qirra og'irliklarini qayta o'zgartiradi, so'ngra har bir tugun uchun Dijkstra algoritmini qo'llaydi. Ushbu maqolada Johnson algoritmining ishlash prinsipi, asosiy bosqichlari, murakkablik tahlili va boshqa algoritmlarga nisbatan afzalliklari ko'rib chiqiladi.

Kalit so'zlar: *Johnson algoritmi, eng qisqa yo'l, barcha juftlik, graf nazariyasi, Dijkstra algoritmi, Bellman-Ford algoritmi, manfiy qirralar, siyrak graf, algoritmik murakkablik, optimizatsiya, yo'l topish, graf algoritmlari, dasturlash, ma'lumotlar strukturasi*

Annotation:

Johnson's algorithm is an efficient algorithm designed to solve the All-Pairs Shortest Path problem. Developed by Donald B. Johnson in 1977, it uses a combination of Dijkstra's and Bellman-Ford algorithms on sparse graphs. Johnson's algorithm can handle graphs with negative edge weights and often shows superior performance compared to the Floyd-Warshall algorithm. The



algorithm first uses Bellman-Ford to detect negative cycles and reweight edges, then applies Dijkstra's algorithm for each vertex. This paper examines the working principle of Johnson's algorithm, its main stages, complexity analysis, and advantages over other algorithms.

Keywords: Johnson algorithm, shortest path, all-pairs, graph theory, Dijkstra algorithm, Bellman-Ford algorithm, negative edges, sparse graph, algorithmic complexity, optimization, pathfinding, graph algorithms, programming, data structures

Аннотация:

Алгоритм Джонсона является эффективным алгоритмом для решения задачи поиска кратчайших путей между всеми парами вершин (All-Pairs Shortest Path). Разработанный Дональдом Б. Джонсоном в 1977 году, он использует комбинацию алгоритмов Дейкстры и Беллмана-Форда для работы с разреженными графами. Алгоритм Джонсона может работать с графами, содержащими рёбра с отрицательными весами, и часто показывает превосходную производительность по сравнению с алгоритмом Флойда-Уоршелла. В данной статье рассматриваются принципы работы алгоритма Джонсона, его основные этапы, анализ сложности и преимущества перед другими алгоритмами.

Ключевые слова: алгоритм Джонсона, кратчайший путь, все пары, теория графов, алгоритм Дейкстры, алгоритм Беллмана-Форда, отрицательные рёбра, разреженный граф, алгоритмическая сложность, оптимизация, поиск пути, алгоритмы на графах, программирование, структуры данных

Graf nazariyasi va algoritmlarning rivojlanishi bilan birga, turli xil masalalarni samarali hal qilish zarurati ham ortib bormoqda. Eng qisqa yo'llarni topish masalasi graf nazariyasining eng muhim va keng qo'llaniladigan masalalaridan biri hisoblanadi. Bu masala transport tizimlarida, tarmoq marshrut tanlashda, loyiha boshqaruvida, ijtimoiy tarmoq tahlilida va ko'plab boshqa



sohalarda qo'llaniladi. Barcha juftlik eng qisqa yo'llar (All-Pairs Shortest Path - APSP) masalasi grafning har bir juft tugun o'rtasidagi eng qisqa yo'lni topishni talab qiladi. Ushbu masalani hal qilish uchun bir nechta algoritmlar ishlab chiqilgan bo'lib, ularning har biri turli xil vaziyatlar va graf turlariga mos keladi.

Johnson algoritmi 1977-yilda Donald B. Johnson tomonidan ishlab chiqilgan bo'lib, u APSP masalasini hal qilishda innovatsion yondashuvni taqdim etadi. Bu algoritmi ayniqsa siyrak (sparse) graflarda yuqori samaradorlik ko'rsatadi va manfiy qirrali graflar bilan ishlash qobiliyatiga ega. Johnson algoritmi ikki mashhur algoritmning - Bellman-Ford va Dijkstra algoritmlarining - kuchli jihatlari birlashtiradi. Bellman-Ford algoritmi manfiy qirralar bilan ishlash imkoniyatini bersa, Dijkstra algoritmi ijobiy qirrali graflarda yuqori tezlikni ta'minlaydi. Johnson algoritmi ushbu ikki algoritmni aqlli tarzda kombinatsiyalab, manfiy qirrali graflarda ham tez ishlash imkoniyatini yaratadi.

Algoritmnining asosiy g'oyasi qirra og'irliklarini qayta o'zgartirish (reweighting) texnikasida yotadi. Avval barcha qirra og'irliklari manfiy bo'lmagan qiymatlarga aylantiriladi, so'ngra har bir tugun uchun Dijkstra algoritmi qo'llaniladi. Bu yondashuv Floyd-Warshall algoritmiga nisbatan ko'plab hollarda yuqori samaradorlik beradi, ayniqsa siyrak graflarda. Johnson algoritmi $O(V^2 \log V + VE)$ vaqt murakkabligiga ega bo'lib, bu siyrak graflarda Floyd-Warshallning $O(V^3)$ murakkabligidan sezilarli darajada yaxshiroqdir.

Hozirgi kunda Johnson algoritmi nafaqat nazariy jihatdan, balki amaliy dasturlashda ham keng qo'llaniladi. U navigatsiya tizimlari, tarmoq protokollari, operatsion tadqiqotlar va optimallashtirish masalalarida muhim rol o'ynaydi. Algoritmnining nisbatan murakkab tuzilishiga qaramay, uning samaradorligi va ko'p qirraliligi uni graf algoritmlari orasida alohida o'rin egallashga olib keladi. Ushbu maqolada Johnson algoritmining chuqur tahlili, uning ishlash mexanizmi, afzalliklari va kamchiliklari hamda boshqa algoritmlar bilan taqqoslash beriladi.

Johnson algoritmining ishlash prinsipi uning ikki bosqichli tuzilmasiga asoslanadi. Birinchi bosqichda algoritmi grafga yangi yordamchi tugun qo'shadi va barcha mavjud tugunlarga nol og'irlikdagi qirralar orqali bog'laydi. So'ngra



ushbu kengaytirilgan grafda Bellman-Ford algoritmi qo'llaniladi. Bellman-Ford algoritmi ikki muhim vazifani bajaradi: birinchidan, grafdagi manfiy tsikllarni aniqlaydi (agar mavjud bo'lsa), ikkinchidan, yangi qo'shilgan tugunden barcha boshqa tugunlarga bo'lgan eng qisqa masofalarni hisoblab chiqadi. Agar manfiy tsikl aniqlansa, algoritm ishini to'xtatadi, chunki bunday holda ba'zi tugunlar juftligi uchun eng qisqa yo'l mavjud emas.

Manfiy tsikl topilmagan taqdirda, ikkinchi bosqich boshlanadi. Bu bosqichda qirra og'irliklarini qayta o'zgartirish (reweighting) jarayoni amalga oshiriladi. Har bir qirra (u,v) uchun yangi og'irlik $w'(u,v) = w(u,v) + h(u) - h(v)$ formulasi orqali hisoblanadi, bu yerda $h(u)$ - Bellman-Ford algoritmi orqali topilgan u tugungacha bo'lgan eng qisqa masofa. Bu o'zgartirish natijasida barcha qirra og'irliklari manfiy bo'lmagan qiymatlarga aylanadi, lekin eng qisqa yo'llarning nisbiy tartibi saqlanib qoladi. Ushbu matematik xususiyat Johnson algoritmining asosiy innovatsiyasidir.

Uchinchi bosqichda esa har bir tugun uchun alohida Dijkstra algoritmi ishga tushiriladi. Qayta o'zgartirilgan qirra og'irliklari manfiy bo'lmagani uchun, Dijkstra algoritmi to'g'ri ishlaydi va har bir tugunden boshqa barcha tugunlarga bo'lgan eng qisqa masofalarni topadi. Nihoyat, topilgan masofalar dastlabki qirra og'irliklariga mos ravishda qayta o'zgartiriladi va yakuniy natija olinadi. Bu jarayon orqali Johnson algoritmi barcha juftlik eng qisqa yo'llar masalasini samarali hal qiladi.

Algoritmning bu ikki bosqichli yondashuvi uning asosiy kuchini tashkil etadi. Bellman-Ford algoritmi manfiy qirralar bilan ishlash imkoniyatini bersa, Dijkstra algoritmi tezlikni ta'minlaydi. Ushbu kombinatsiya Johnson algoritmini ko'plab amaliy vaziyatlarda optimal tanlovga aylantiradi, ayniqsa siyrak va o'rtacha zichlikdagi graflarda yuqori samaradorlik ko'rsatadi.

Johnson algoritmi zamonaviy graf algoritmlarining eng samarali namunalaridan biri sifatida bir qator muhim afzalliklarga ega. Birinchi va asosiy afzalligidan biri uning **siyrak graflarda yuqori samaradorligidir**. Floyd-Warshall algoritmi barcha holatlarda $O(V^3)$ vaqt murakkabligiga ega bo'lsa,



Johnson algoritmi $O(V^2 \log V + VE)$ murakkablikka ega bo'lib, qirra soni kam bo'lgan graflarda sezilarli tezlik ustunligini beradi. Masalan, qirra soni $O(V)$ tartibida bo'lgan graflarda Johnson algoritmi Floyd-Warshall dan bir necha marta tezroq ishlaydi.

Ikkinchi muhim afzalligi - algoritmning **manfiy qirrali graflar bilan ishlash qobiliyati**dir. Ko'plab amaliy masalalarda qirra og'irliklarining manfiy bo'lishi tabiiy holat hisoblanadi, masalan, arbitraj savdosida, loyiha boshqaruvida yoki resurs taqsimotida. Johnson algoritmi bunday graflarda ham ishonchli natija beradi, faqat manfiy tsikllar mavjud bo'lmasligi kerak. Uchinchidan, algoritm **xotira samaradorligi** jihatidan ham afzallikka ega. Floyd-Warshall algoritmi barcha juftlik masofalari uchun $O(V^2)$ xotira talab qilsa, Johnson algoritmi faqat joriy hisoblashlar uchun kerakli xotirani ishlatadi va natijalarni bosqichma-bosqich chiqarishi mumkin.

Shuningdek, Johnson algoritmi **modulli tuzilmaga** ega bo'lib, uning tarkibiy qismlari (Bellman-Ford va Dijkstra) mustaqil ravishda optimizatsiyalanishi va almashtirilishi mumkin. Bu xususiyat algoritmni turli xil amaliy vaziyatlarga moslashtirishga imkon beradi. **Parallellashtirish imkoniyati** ham algoritmning muhim afzalliklaridan biri hisoblanadi - har bir tugun uchun Dijkstra algoritmi mustaqil ravishda bajarilishi mumkin bo'lgani uchun, ko'p yadroli tizimlarda samaradorlikni oshirish imkoniyati mavjud.

Biroq Johnson algoritmining ba'zi **kamchiliklari** ham mavjud. Birinchidan, uning **murakkab tuzilishi** oddiy algoritmlarcha oson tushunilmaydi va amalga oshirilmaydi. Ikkinchidan, **zichlik yuqori graflarda** (qirra soni V^2 ga yaqin bo'lganda) Floyd-Warshall algoritmi Johnson algoritmidan tezroq ishlashi mumkin. Uchinchidan, algoritmning ikki bosqichli tuzilishi **kichik graflarda ortiqcha murakkablik** keltirib chiqarishi mumkin, bu holda oddiy algoritmlar samaraliroq bo'ladi.

Shuningdek, **manfiy tsikllar mavjudligini tekshirish** qo'shimcha vaqt talab qiladi va bu ba'zi hollarda keraksiz bo'lishi mumkin. **Xotira fragmentatsiyasi** ham muammo bo'lishi mumkin, chunki algoritm turli xil



ma'lumot strukturalarini bir vaqtda ishlatadi. Umuman olganda, Johnson algoritmi o'zining murakkabligi va spetsifikligiga qaramay, ko'plab amaliy vaziyatlarda optimal yechim bo'lib, ayniqsa o'rta va katta hajmdagi siyrak graflarda ustun samaradorlik ko'rsatadi.

Johnson algoritmi zamonaviy dasturlash va graf nazariyasining amaliy qo'llanishida keng o'rin olgan. Uning eng muhim qo'llanish sohalaridan biri **transport va logistika tizimlaridir**. Yirik shahar transport tarmoqlarida, aviakompaniyalar marshrut tarmog'ida yoki yuk tashish tizimlarida barcha nuqtalar orasidagi eng qisqa yo'llarni topish uchun Johnson algoritmi qo'llaniladi. Bu sohalarda manfiy og'irliklar ham uchraydi, masalan, yo'l haqi chegirmalari yoki yoqilg'i tejash imkoniyatlari ko'rinishida.

Tarmoq protokollari va internet marshrutlashda ham Johnson algoritmi muhim rol o'ynaydi. BGP (Border Gateway Protocol) va OSPF (Open Shortest Path First) kabi protokollarda barcha router juftliklari orasidagi optimal yo'llarni topish uchun ushbu algoritm yoki uning modifikatsiyalari ishlatiladi. **Ijtimoiy tarmoq tahlilida** ham algoritm keng qo'llaniladi - foydalanuvchilar orasidagi eng qisqa aloqa zanjirlarini topish, ta'sir ko'rsatish darajasini aniqlash va ijtimoiy graflarni tahlil qilishda foydalaniladi.

Moliya va investitsiya sohasida Johnson algoritmi arbitraj imkoniyatlarini topish, portfel optimizatsiyasi va risk tahlilida qo'llaniladi. Valyuta ayirboshlash kurslarini graf sifatida tasvirlab, manfiy logarifmik og'irliklar yordamida arbitraj imkoniyatlarini aniqlash mumkin. **O'yin nazariyasi va strategik rejalashtirishda** ham algoritm foydali - turli strategiyalar orasidagi optimal o'tish yo'llarini topish va Nash muvozanatini aniqlashda ishlatiladi.

Dasturlash tillarida Johnson algoritmi turli kutubxonalar orqali amalga oshirilgan. Python'da NetworkX kutubxonasi, Java'da JGraphT, C++da Boost Graph Library kabi kutubxonalarda algoritm mavjud. **Ma'lumotlar tahlili va mashinali o'rganishda** ham algoritm qo'llaniladi - xususiyatlar orasidagi bog'liqliklarni tahlil qilish, klasterlash algoritmlari uchun masofa matritsalarini hisoblash va grafik neural tarmoqlarda yo'l topish masalalarida foydalaniladi.



Amaliy amalga oshirishda Johnson algoritmi ko'pincha optimizatsiyalar bilan birgalikda ishlatiladi: Fibonacci heap yordamida Dijkstra algoritmi tezligini oshirish, parallellashtirish texnikalari va kesh optimizatsiyasi. Bu yondashuvlar algoritmni real vaqt rejimida ishlashga imkon beradi va katta hajmdagi graflarda ham samarali natija olishga yordam beradi.

Johnson algoritmi hozirgi zamonaviy dasturlash va graf nazariyasida muhim o'rin egallab, barcha juftlik eng qisqa yo'llar masalasini hal qilishda samarali va ishonchli yechim taqdim etadi. Uning hibrid yondashuvi - Bellman-Ford va Dijkstra algoritmilarining kombinatsiyasi - manfiy qirrali graflarda ham yuqori samaradorlikni ta'minlaydi. Algoritmning $O(V^2 \log V + VE)$ murakkabligi uni siyrak graflarda Floyd-Warshall algoritmiga nisbatan afzal tanlovga aylantiradi. Qirra og'irliklarini qayta o'zgartirish texnikasi Johnson algoritmining asosiy innovatsiyasi bo'lib, ushbu yondashuv graf nazariyasining nazariy asoslarini ham boyitgan.

Kelajakda Johnson algoritmini yanada takomillashtirish yo'nalishlari mavjud. Parallel hisoblash imkoniyatlaridan kengroq foydalanish, kvant hisoblash texnologiyalari bilan integratsiya va sun'iy intellekt yondashuvlari bilan kombinatsiyalash istiqbolli yo'nalishlar hisoblanadi. Shuningdek, algoritm Internet of Things (IoT), blokchejn texnologiyalari va katta ma'lumotlar tahlilida yangi qo'llanish sohaslarini topmoqda.

Tavsiya sifatida, dasturlash va graf algoritmilari bilan shug'ullanuvchi mutaxassislar Johnson algoritmini nafaqat nazariy jihatdan, balki amaliy loyihalarda ham chuqur o'rganishlari zarur. Algoritmni turli dasturlash tillarida amalga oshirish, uning samaradorligini boshqa algoritmilar bilan solishtirib ko'rish va real ma'lumotlar ustida sinovdan o'tkazish muhim amaliy ko'nikmalarni rivojlantiradi. Algoritm ustida tadqiqot olib borish, uni optimallashtirish va yangi qo'llanish sohaslarini topish graf nazariyasi va amaliy dasturlash rivojiga muhim hissa qo'shishi mumkin.

Xulosa qilib aytganda, Johnson algoritmi nafaqat samarali graf algoritmi, balki algoritmik fikrlashning mukammal namunasi sifatida ham qadrlanadi.



Uning o'rganilishi dasturchilar va tadqiqotchilarga kompleks masalalarni hal qilishda kreativ yondashuvlar qo'llash va turli algoritmlarning kuchli jihatlarini birlashtirish usullarini o'rgatadi.

Foydalanilgan adabiyotlar

1. Johnson, Donald B. "Efficient algorithms for shortest paths in sparse networks". Journal of the ACM, 1977.
2. Thomas H. Cormen et al. *Introduction to Algorithms*. MIT Press, 4th Edition.
3. Robert Sedgewick, Kevin Wayne. *Algorithms*. Addison-Wesley, 4th Edition.
4. Ahuja, Ravindra K., et al. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall.
5. Kleinberg, Jon; Tardos, Éva. *Algorithm Design*. Addison-Wesley.
6. Bang-Jensen, Jørgen; Gutin, Gregory. *Digraphs: Theory, Algorithms and Applications*. Springer.
7. West, Douglas B. *Introduction to Graph Theory*. Prentice Hall, 2nd Edition.
8. GeeksforGeeks. *Johnson's Algorithm for All-pairs Shortest Paths*.
<https://www.geeksforgeeks.org>
9. Wikipedia. *Johnson's Algorithm*.
https://en.wikipedia.org/wiki/Johnson%27s_algorithm
10. LeetCode Discussions. *All-Pairs Shortest Path Algorithms Comparison*.
11. Stack Overflow. *Implementation of Johnson's Algorithm*.
12. NetworkX Documentation. *Shortest Path Algorithms in Python*.
13. Boost Graph Library Documentation. *Johnson All-Pairs Shortest Paths*.
14. Algorithm Visualizations. *Johnson's Algorithm Step-by-Step*.
15. Lecture Notes on Graph Algorithms. *Advanced Shortest Path Techniques*. Computer Science Department, MIT.