Bellman - Ford vs Dijkstra

Matthew Ellis
Gavin Kerr
Shefali Patel
11/16/15

INTRODUCTION

Dijkstra

- Computes shortest path
- Does not allow negative weight edges
- Time Complexity: O(V^2)

Bellman-Ford

- Computes shortest path
- Allows negative weight edges
- Time Complexity: O(VE)

Scope

- Compare the various implementation of Dijkstra’s and Bellman-Ford’s algorithms to determine case usage and optimization
- Implement various forms and test against various working sets
- Analyze the results of each tests to determine behavior from each implementation that may not be apparent
- Find best case situations for each form

RESEARCH

A Distributed Power Routing Method between Regional Markets based on Bellman-Ford Algorithm [1]

- Implementation of the Bellman-Ford Algorithm for practical application in the real world
- Focused the problem of routing power through regional markets that have been deregulated
- The cheapest route needed to be selected for power exchange
- The Bellman-Ford Algorithm was implemented to determine the cheapest route and to develop the convergence analysis

Extended Dijkstra algorithm and Moore-Bellman-Ford algorithm [2]

- Discusses extended versions of Dijkstra's and Bellman-Ford's algorithms for general single-source shortest path problems
- Shows that the extended Dijkstra's algorithm is O(n^2)M(n)
- Shows that the extended Bellman-Ford's algorithm is O(mn)M(n)
- Some applications of these implementation are also explored

Randomized Speedup of the Bellman-Ford Algorithm [3]

- While using the Bellman-Ford Algorithm, they randomly permute the vertices
- This implementation works to reduce the worst-case for the number of relaxation steps
- Additionally, adds in negative cycle detection

Optimizing Dijkstra for real-world performance [4]

- Introduces an alternative queue design to the original Fibonacci Heap to speed up the algorithm
- The prototype implementation is twice as fast as the “Boost implementation”
- The code for this specific implementation is publicly available

SHORTEST PATH WITH DYNAMIC WEIGHT IMPLEMENTATION USING DIJKSTRA'S ALGORITHM [5]

- Used street routes as a graph to apply the Dijkstra algorithm
- They demonstrated both one-way and two-way graphs
- Concludes that Dijkstra's algorithm is not suitable to solve two-way graphs with dynamic weights
- Claims no performance drawbacks from recalculating the shortest path at every change

Bellman-Ford: Brilliant.com [6]

- Designed to find the shortest path between two points on a graph
- Does not work if graph contains a negative cycle
  - However, the algorithm can be used to detect negative cycles
- Uses principle of relaxation
  - Continuously shortens the distance between vertices
  - Bad ordering can cause exponential relaxations

Use Cases of Bellman-Ford: Programiz.com [7]

- B-F can be used to solve problems in cashflow, movement of heat, etc.
- The algorithm works by overestimating the path length, then iteratively relaxing the estimates by finding shorter paths
- Similar to Dijkstra’s algorithm, but BF goes through every each edge in every iteration

Comparisons: IJPTT: “A survey paper…”[8]

- Bellman-Ford has a built-in functionality to find out if there is no solution to a shortest path problem
  - Done by detection of negative cycles, which continually decrease distance to negative infinity
- The simplest implementation of Bellman-Ford is O(|V| - |E|)
- Dijkstra’s algorithm can only be used if no negative distances exist within the graph
- The simplest implementation of Dijkstra is O(|V|^2)

CMU Lecture [9]

- Dijkstra’s algorithm does not work with negative edges because it does not correctly measure distances
- Bellman-Ford solves this issue by adding the vertices one by one arbitrarily

Application: IRJET [10]

- By using Bellman-Ford or Dijkstra's algorithm, it is possible to find the best route to take given a GPS position and a map containing a set of roads that act as edges, and intersections that act as vertices
- As real-world distance is non-negative, both algorithms are valid
- Ultimately, Dijkstra's algorithm appeared to be more efficient in this case than Bellman-Ford

Modified Dijkstra Algorithm with Invention Hierarchies Applied to a Conic Graph [11]

- Goes over a modified version of the Dijkstra Algorithm
- Utilizes an “inventive contraction hierarchy” to increase efficiency
- Creates new path using edge difference for the minimum edge pair and its neighbour

Dynamic Scope-Based Dijkstra's Algorithm [12]

- Focus on scope for dynamic applications of Dijkstra’s Algorithm
- Looks at road networks that my be dynamic in nature (they change over time)
- Also takes into consideration the size of the network when considering the effectiveness of the algorithm

An Investigation of Dijkstra and Floyd Algorithms in National City Traffic Advisory Procedures [13]

- Focuses on the implementation of the Dijkstra algorithm for city traffic control
- The goal was to produce efficient and optimal traffic rules and procedures
- Analyzed time and cost parameters across cities in China

Anapplication of Dijkstra’s Algorithm to shortest route problem. [14]

- Focuses on an application of the Dijkstra’s Algorithm in the transportation of goods and products
- An incredibly practical example of the application of the Dijkstra’s Algorithm for a real word company
- Presents exact solution to the present problem

Dijkstra’s algorithm revisited: the dynamic programming connexion [15]

- Attempts to remove the idea that Dijkstra's algorithm is a “greedy” algorithm
- Brings in a dynamic framework to make this argument
- They imply that, through this argument, Dijkstra’s algorithm can be added to the dynamic programming index

CONCLUSION

Deliverables

- Coded implementation of various forms of the two algorithms we have found
- Data pertaining to the results of each implementation across various working sets
- 10 page paper analyzing the data that we have produced

References

[1] T. Xue, S. Hongbin, Z. Li, and G. Qinglai, “A Distributed Power Routing Method between Regional Markets based on Bellman-Ford Algorithm,” *[astro-ph/0005112] A Determination of the Hubble Constant from Cepheid Distances and a Model of the Local Peculiar Velocity Field*, 07-Nov-2017. [Online]. Available: https://arxiv.org/abs/1711.02470. [Accessed: 15-Nov-2018].

[2] C. Cong-Dian, “Extended Dijkstra algorithm and Moore-Bellman-Ford algorithm,” *[astro-ph/0005112] A Determination of the Hubble Constant from Cepheid Distances and a Model of the Local Peculiar Velocity Field*, 14-Aug-2017. [Online]. Available: https://arxiv.org/abs/1708.04541. [Accessed: 15-Nov-2018].

[3] M. J. Bannister and D. Eppstein, “Randomized Speedup of the Bellman-Ford Algorithm,” *[astro-ph/0005112] A Determination of the Hubble Constant from Cepheid Distances and a Model of the Local Peculiar Velocity Field*, 23-Nov-2011. [Online]. Available: https://arxiv.org/abs/1111.5414. [Accessed: 15-Nov-2018].

References

[4] N. Aviram and Y. Shavitt, “Optimizing Dijkstra for real-world performance,” *[astro-ph/0005112] A Determination of the Hubble Constant from Cepheid Distances and a Model of the Local Peculiar Velocity Field*, 19-May-2015. [Online]. Available: https://arxiv.org/abs/1505.05033. [Accessed: 15-Nov-2018].

[5] E. N. Tamatjita and A. W. Mahastama, “SHORTEST PATH WITH DYNAMIC WEIGHT IMPLEMENTATION USING DIJKSTRA’S ALGORITHM,” *Informing Science The International Journal of an Emerging Transdiscipline*, 01-Sep-2016. [Online]. Available: https://doaj.org/article/dc41e0f818f44d649c14b83594c2c3ea. [Accessed: 15-Nov-2018].

[6] Patel, V. (2018). *A survey paper of Bellman - ford algorithm and Dijkstra algorithm for finding shortest path in GIS application*. [online] Ijpttjournal.org. Available at: http://www.ijpttjournal.org/volume-5/IJPTT-V5P401.pdf [Accessed 16 Nov. 2018].

References

[7] Chumbly, A., Moore, K. and Ross, E. (2018). *Bellman Ford's Algorithm*. [online] Brilliant. Available at: https://brilliant.org/wiki/bellman-ford-algorithm/ [Accessed 16 Nov. 2018].

[8] Programiz.com. (2018). *Bellman Ford's Algorithm*. [online] Available at: https://www.programiz.com/dsa/bellman-ford-algorithm [Accessed 16 Nov. 2018].

[9] Acar, U. (2018). *Lecture 13 — Shortest Weighted Paths II*. [online] Cs.cmu.edu. Available at: https://www.cs.cmu.edu/afs/cs/academic/class/15210-s13/www/lectures/lecture13.pdf [Accessed 16 Nov. 2018].

[10] Kalpana and Tyagi, A. (2018). *Bellman Ford Shortest Path Algorithm using Global Positioning System*. [online] Irjet.net. Available at: https://www.irjet.net/archives/V4/i4/IRJET-V4I4624.pdf [Accessed 16 Nov. 2018].

References

[11] U. A. Okengwu, E. O. Nwachukwu, and E. N. Osegi, “Modified Dijkstra Algorithm with Invention Hierarchies Applied to a Conic Graph,” *[astro-ph/0005112] A Determination of the Hubble Constant from Cepheid Distances and a Model of the Local Peculiar Velocity Field*, 19-Mar-2015. [Online]. Available: https://arxiv.org/abs/1503.02517. [Accessed: 23-Nov-2018].

[12] P. Hlineny and O. Moris, “Dynamic Scope-Based Dijkstra's Algorithm,” *[astro-ph/0005112] A Determination of the Hubble Constant from Cepheid Distances and a Model of the Local Peculiar Velocity Field*, 10-Jan-2012. [Online]. Available: https://arxiv.org/abs/1201.2000. [Accessed: 23-Nov-2018].

[13] A. K. Sangaiah, M. Han, and S. Zhang, “An Investigation of Dijkstra and Floyd Algorithms in National City Traffic Advisory Procedures,” *International Journal of Computer Science and Mobile Computing*, Feb-2014. [Online]. Available: https://ijcsmc.com/docs/papers/February2014/V3I2201442.pdf. [Accessed: 23-Nov-2018].

References

[14] Ojekudo, N. Akpofure, Akpan, and N. Paul, “Anapplication of Dijkstra’s Algorithm to shortest route problem.,” *IOSR Journal of Mathematics (IOSR-JM)* , May-2017. [Online]. Available: http://www.iosrjournals.org/iosr-jm/papers/Vol13-issue3/Version-1/C1303012032.pdf. [Accessed: 23-Nov-2018].

[15] M. Sniedovich, “Dijkstra’s algorithm revisited: the dynamic programming connexion,” *Control and Cybernetics*, 2006. [Online]. Available: http://matwbn.icm.edu.pl/ksiazki/cc/cc35/cc3536.pdf. [Accessed: 23-Nov-2018].