DHBW
Duale Hochschule
Baden-Württemberg
Karlsruhe

Runtime complexity analysis of Johnson's algorithm

Exercise as part of the lecture Wissenschaftliches Arbeiten I

as part of the exam for

Bachelor of Science (B.Sc.)

of the course Informatik

at the Dualen Hochschule Baden-Württemberg Karlsruhe

from

Oliver Schirmer

Submission date:	20. Juli 2021
Processing period:	13.07.2021 - 20.07.2021
Matriculation number, class:	7059763, TINF20B2
Training company:	SAP SE Dietmar-Hopp-Allee 16 69190 Walldorf, Deutschland
Trainter at training company:	Simon Nichterlein
Expert from the cooperative university:	Heinrich Braun

Statutory declaration

I hereby acknowledge, that my Exercise as part of the lecture Wissenschaftliches Arbeiten I with the topic:

Runtime complexity analysis of Johnson's algorithm

according to § 5 of the „Studien- und Prüfungsordnung DHBW Technik“ from 29th of September 2017 has been independently written and no other sources or aids other than those specified have been used. The work has not yet been submitted to any other examination authority and has not been published.

I also assure that the submitted electronic version matches the printed version.

Karlsruhe, the 14. Juli 2021

Signed Oliver Schirmer
Schirmer, Oliver

Equality notice

For better readability, gender-specific duplications are not used.

Abstract

There are many algorithms which solve the All-pairs shortest path (APSP). By far the most renowned is Floyd-Warshall algorithm. Most of the algorithms have a pretty bad runtime complexity as for example Floyd-Warshall algorithm with $O(|V|^3)$.

The aim of this work is to take a closer look at Johnson's algorithm. This algorithm poses an alternative to Floyd-Warshall algorithm with the goal of reducing runtime complexity.

Inhaltsverzeichnis

Abkürzungsverzeichnis

APSP	All-pairs shortest path
DP	Dynamic Programming
SSSP	Single source shortest path

1 Dijkstra

Dijkstra algorithm is a greedy algorithm for finding the shortest paths between one vertex and all other vertices in a given graph. Therefore it solves the Single source shortest path (SSSP). One of the major constraints of Dijkstra algorithm is that it can only be used on graphs with strictly positive edge weigths. [1]

Despite the mentioned limitations Dijkstra is one of the fastest algorithms to solve the SSSP. The respective complexity depends on the implementation of the min priority queue. Using a fibonacci heap results in a runtime complexity of $O(|V| * log|V| + |E|)$. [2]

2 Bellman-Ford

Bellman-Ford is the most renowned algorithm for solving the APSP with Dynamic Programming (DP). As opposed to Dijkstra it also works with graphs which contain negative edge weights. It also is able to detect negative cycles in graphs. [1, 2]

On the other hand the Bellman-Ford algorithm has a runtime complexity of $O(|V| * |E|)$ which is not to be neglected. [1, 2]

3 Johnson's algorithm

Johnson's algorithm combines the previously considered algorithms Dijkstra & Bellman-Ford to solve the APSP with a faster runtime complexity than Floyd-Warshall algorithm, which runs in $O(|V|^3)$. [1, 2]

To do so the limitations of Dijkstra have to be considered. As Dijkstra doesn't support negative edge weights Bellman-Ford is used to transform the graph into a graph with non-negative edge weights (or raises detected negative cycles). After that Dijkstra is applied to every vertex in the graph to solve the SSSP for every vertex. The final shortest distances can be computed by a combination of the values computed by Bellman-Ford and Dijkstra. [2]

The runtime complexity of this algorithm consists of the different stages of the algorithm. The first stage - transformation via Bellman-Ford - is the plain application of the Bellman-Ford algorithm and therefore has a runtime complexity of $O(|V| * |E|)$. The second stage - computation of shortest paths for all pairs of vertices via Dijkstra - adds a factor of $|V|$ to the runtime complexity as Dijkstra is applied to every vertex. Therefore this stage has a runtime complexity of $O(|V|^2 * log|V|)$. The last stage can be computed in constant time as it just accesses previously computed values. [1, 2]
Putting together the single parts Johnson's algorithm exhibits a runtime complexity of $O(|V|^2 * log|V| + |V| * |E|)$, which is considerably faster than $O(|V|^3)$. Therefore we have found a faster alternative to Floyd-Warshall algorithm.

Literaturverzeichnis

[1] Tamimi, A. A. „Comparison Studies for Different Shortest path Algorithms“. In: *INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY* 14.8 (2015), S. 5979–5986. URL: https://www.researchgate.net/profile/hanaa-abu-ryash/publication/283666925_comparison_studies_for_different_shortest_path_algorithms.

[2] Kiruthika, R./ Umarani, R. „Shortest path algorithms: A comparative analysis“. In: *International Journal of Managment, IT and Engineering* 2.4 (2012), S. 55–62. URL: https://www.indianjournals.com/ijor.aspx?target=ijor:ijmie&volume=2&issue=4&article=005.