ECM 3428 Coursework – Boruvka's Algorithm

Introduction

Boruvka's algorithm is an algorithm that solves the minimum spanning tree problem. The minimum spanning tree problem is where the aim is to connect all the node in a graph with a tree which has the smallest possible sum of all edges. It is important to note that a tree is a subgraph where there are no cycles. The key principle of the algorithm is that every node is connected to its nearest neighbour and then the resulting trees are interconnected. Boruvka's algorithm is a greedy algorithm, which means that it will always choose the best option available to it at any one time. Consequently, if some information becomes available such as a negative weight which will change the best answer, the algorithm will not find an optimal solution [1].

Why is it Influential?

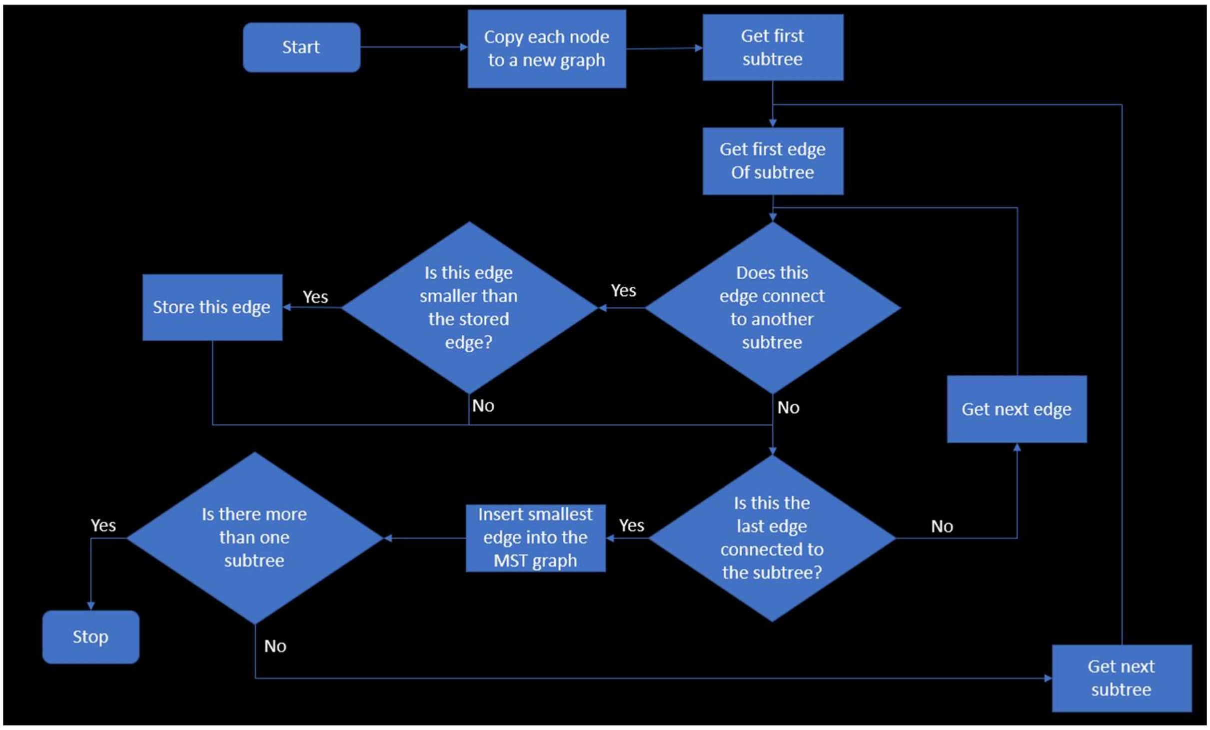
Boruvka's algorithm was the first algorithm which solved the minimum spanning tree problem[2]. It was created in 1926 by Otakar Boruvka, approximately 27 years earlier than more common algorithms which are used to solve the minimum spanning tree problem such as Prim's and Kruskal's algorithms [3].

Solving minimum spanning tree problems was influential, as finding the minimum spanning tree directly relates to the cost of connecting some networks. For example, when Boruvka originally developed the algorithm, he was tasked with developing the cheapest electrical network in Western Moravia (a provice in the Czech Republic) [3]. The minimum spanning tree was suitable here as the aim was to minimise the cost of the network, most effectively done by minimising the amount of cable used for the network. This can be modelled as a minimum spanning tree as each connection point in the town of Moravia could be modelled as a node and the distance between them could be represented as an edge in the graph. A minimum spanning tree is needed as each node must have a path to the power source for all the connection points to have access to electricity.

Limitations

Unfortunately, Boruvka's algorithm is not really implemented in modern computer science due to more modern algorithms outperforming the algorithm such as the aforementioned Prim's and Kruskal's. Additionally, Boruvka's algorithm may not find an optimal solution if there are negative weights in the graph. A possible method of dealing with this limitation may be to simply add the absolute value of the lowest edge in the graph to each node. This would ensure all edges have a weight of 0 or larger. Because of this the Boruvka's would now be able to find the minimum spanning tree.

Pseudocode



Complexity

The time complexity of Boruvka's Algorithm is O(m log n). Where m is the number of edges and n is the number of nodes [1]. This is explained below.

When the algorithm first runs, there is only a set of unconnected nodes where each edge of each node must be looped through to find the closest node. This creates a forest where there is maximum of n/2 trees (as each node must be connected to another node). Since the algorithm is only interested in the edges which connect the trees in the forest, in theory each tree could be treated as a single node, repeating the first step until only one tree exists which could be treated as a graph with a single node in the context of this algorithm. Therefore, since the maximum number of trees remaining halves each time the minimum spanning tree will be reached in log n iterations. Since all the edges must be evaluated for each step the time complexity of a single step is the time to iterate through each of the edges. Consequently, the total time complexity is the time to evaluate all edges multiplied by the number of steps and therefore the time complexity is O(m log n).

References

[1] C. F. Bazlamaçc and K. S. Hindi, "Minimum-weight spanning tree algorithms a survey and empirical study," *Computers & Operations Research*, vol. 28, no. 8, pp. 767–785, 2001.

[2] F. Marpaung and others, "Comparative of prim's and boruvka's algorithm to solve minimum spanning tree problems," in *Journal of Physics: Conference Series*, 2020, vol. 1462, no. 1, p. 12043.

[3] H. Durnová, "Otakar Boruvka (1899-1995) and the Minimum Spanning Tree," 1998.

[4] A. Mariano, A. Proenca, and C. D. S. Sousa, "A generic and highly efficient parallel variant of boruvka's algorithm," in *2015 23rd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*, 2015, pp. 610–617.