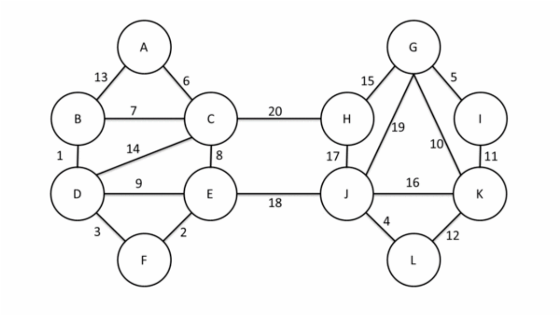
Borůvka’s algorithm



Animation of Boruvka’s algorithm

Borůvka’s algorithm is an algorithm for finding a minimum spanning tree in a graph for which all edge weights are distinct.

It was first published in 1926 by Otakar Borůvka as a method of constructing an efficient electricity network for Moravia.[1][2][3] The algorithm was rediscovered by Choquet in 1938;[4] again by Florek, Łukasiewicz, Perkal, Steinhaus, and Zubrzycki[5] in 1951; and again by Sollin [6] in 1965. Because Sollin was the only computer scientist in this list living in an English speaking country, this algorithm is frequently called **Sollin’s algorithm**, especially in the parallel computing literature.

The algorithm begins by first examining each vertex and adding the cheapest edge from that vertex to another in the graph, without regard to already added edges, and continues joining these groupings in a like manner until a tree spanning all vertices is completed.

1 Pseudocode

Designating each vertex or set of connected vertices a "component", pseudocode for Borůvka’s algorithm is:

```
Input: A connected graph G whose edges have distinct weights
Initialize a forest T to be a set of one-vertex trees, one for each vertex of the graph.
While T has more than one component:
    For each component C of T:
        Begin with an empty set of edges S
        For each vertex v in C:
            Find the cheapest edge from v to a vertex outside of C, and add it to S
        Add the cheapest edge in S to T
    Combine trees connected by edges to form bigger components
Output: T is the minimum spanning tree of G.
```

As in Kruskal’s algorithm, tracking components of T can be done efficiently using a disjoint-set data structure. In graphs where edges have identical weights, edges with equal weights can be ordered based on the lexicographic order of their endpoints.

2 Complexity

Borůvka’s algorithm can be shown to take O(log V) iterations of the outer loop until it terminates, and therefore to run in time O(E log V), where E is the number of edges, and V is the number of vertices in G. In planar graphs, and more generally in families of graphs closed under graph minor operations, it can be made to run in linear time, by removing all but the cheapest edge between each pair of components after each stage of the algorithm.[7]

3 Example

4 Other algorithms

Other algorithms for this problem include Prim’s algorithm and Kruskal’s algorithm. Fast parallel algorithms can be obtained by combining Prim’s algorithm with Borůvka’s.[8]

A faster randomized minimum spanning tree algorithm based in part on Borůvka’s algorithm due to Karger, Klein, and Tarjan runs in expected O(E) time.[9] The best known[*update needed*] (deterministic) minimum spanning tree algorithm by Bernard Chazelle is also based in part on Borůvka’s and runs in O(E α(E,V)) time, where α is the inverse of the Ackermann function.[10] These randomized and deterministic algorithms combine steps of Borůvka’s algorithm, reducing the number of components that remain to be connected, with steps of a different type that reduce the number of edges between pairs of components.

5 Notes

[1] Borůvka, Otakar (1926). “O jistém problému minimálním” [About a certain minimal problem]. *Práce mor. přírodověd. spol. v Brně III* (in Czech and German). **3**: 37–58.

[2] Borůvka, Otakar (1926). “Příspěvek k řešení otázky ekonomické stavby elektrovodních sítí (Contribution to the

solution of a problem of economical construction of electrical networks)". *Elektronický Obzor* (in Czech). **15**: 153–154.

[3] Nešetřil, Jaroslav; Milková, Eva; Nešetřilová, Helena (2001). "Otakar Borůvka on minimum spanning tree problem: translation of both the 1926 papers, comments, history". *Discrete Mathematics*. **233** (1–3): 3–36. doi:10.1016/S0012-365X(00)00224-7. MR 1825599.

[4] Choquet, Gustave (1938). "Étude de certains réseaux de routes". *Comptes-rendus de l'Académie des Sciences* (in French). **206**: 310–313.

[5] Florek, Kazimierz (1951). "Sur la liaison et la division des points d'un ensemble fini". *Colloquium Mathematicum* (in French). **2**: 282–285.

[6] Sollin, M. (1965). "Le tracé de canalisation". *Programming, Games, and Transportation Networks* (in French).

[7] Eppstein, David (1999). "Spanning trees and spanners". In Sack, J.-R.; Urrutia, J. *Handbook of Computational Geometry*. Elsevier. pp. 425–461.; Mareš, Martin (2004). "Two linear time algorithms for MST on minor closed graph classes" (PDF). *Archivum mathematicum*. **40** (3): 315–320..

[8] Bader, David A.; Cong, Guojing (2006). "Fast shared-memory algorithms for computing the minimum spanning forest of sparse graphs". *Journal of Parallel and Distributed Computing*. **66** (11): 1366–1378. doi:10.1016/j.jpdc.2006.06.001.

[9] Karger, David R.; Klein, Philip N.; Tarjan, Robert E. (1995). "A randomized linear-time algorithm to find minimum spanning trees". *Journal of the ACM*. **42** (2): 321–328. doi:10.1145/201019.201022.

[10] Chazelle, Bernard (2000). "A minimum spanning tree algorithm with inverse-Ackermann type complexity" (PDF). *J. ACM*. **47** (6): 1028–1047. doi:10.1145/355541.355562.

6 Text and image sources, contributors, and licenses

6.1 Text

- **Borůvka’s algorithm** *Source:* https://en.wikipedia.org/wiki/Bor%C5%AFvka’{}s_algorithm?oldid=773835173 *Contributors:* PierreAbbat, Davitf, Hike395, Charles Matthews, Dcoetzee, Dysprosia, Jogloran, Grendelkhan, Jaredwf, Altenmann, Captain Segfault, Giftlite, Rich Farmbrough, Zaslav, Oleg Alexandrov, Angr, Qwertyus, Rjwilmsi, YurikBot, Michael Slone, Toncek, Cedar101, SmackBot, Nkojuharov, Mgreenbe, GoodDay, Alieseraj, Kendrick7, Agamir, Headbomb, Rps, David Eppstein, Cobi, Adam Zivner, Falcongl, AlleborgoBot, Alexbot, PixelBot, Aurinegro~enwiki, Addbot, Lightbot, Luckas-bot, TaBOT-zerem, Senitiel, Citation bot 1, Kaeso, Trappist the monk, EmausBot, Elanguescence, WikitanvirBot, ZéroBot, Swfung8, Jk2q3jrklse, Electriccatfish2, JeffAEdmonds and Anonymous: 21

6.2 Images

- **File:Boruvka’{}s_algorithm_(Sollin’{}s_algorithm)_Anim.gif** *Source:* https://upload.wikimedia.org/wikipedia/commons/2/2e/Boruvka%27s_algorithm_%28Sollin%27s_algorithm%29_Anim.gif *License:* CC BY-SA 3.0 *Contributors:* Own work *Original artist:* Alieseraj
- **File:Borůvka_Algorithm_1.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/9/9c/Bor%C5%AFvka_Algorithm_1.svg *License:* CC BY-SA 3.0 *Contributors:* Own work, based on File:Prim Algorithm 0.svg *Original artist:* User:Dcoetzee, User:Maksim, User:Alexander Drichel
- **File:Borůvka_Algorithm_2.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/5/5c/Bor%C5%AFvka_Algorithm_2.svg *License:* CC BY-SA 3.0 *Contributors:* Own work, based on File:Prim Algorithm 0.svg *Original artist:* User:Dcoetzee, User:Maksim, User:Alexander Drichel
- **File:Borůvka_Algorithm_3.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/8/86/Bor%C5%AFvka_Algorithm_3.svg *License:* CC BY-SA 3.0 *Contributors:* Own work, based on File:Prim Algorithm 0.svg *Original artist:* User:Dcoetzee, User:Maksim, User:Alexander Drichel

6.3 Content license

- Creative Commons Attribution-Share Alike 3.0