BEST Theorem

Special Mathematics Lecture (Graph Theory)
Nagoya University, Spring 2024

YANG Ming-Tian
The University of Hong Kong

5th June 2024

1 Introduction

This report provides an overview of the BEST theorem, named in honour of its discoverers: N. G. de Bruijn, T. van Aardenne-Ehrenfest, C. A. B. Smith, and W. T. Tutte. The theorem offers a formula to determine the number of Eulerian tours in Eulerian graphs. It is rooted in an extension of Kirchhoff's matrix-tree theorem, whose introduction and proof can be found in Oleh's prior report [5].

To determine the number of Eulerian tours present in a connected, directed and finite graph $G = (V, E)$, it is important to first confirm that the graph is indeed Eulerian. Otherwise, no Eulerian tours will exist. According to Theorem 6.2(i) in the lecture notes [7], a connected, directed and finite graph G is Eulerian if and only if $\deg_{\text{in}}(v) = \deg_{\text{out}}(v)$ for any $v \in V$.

The BEST theorem can be stated as follows [9]:

Theorem 1 (BEST theorem)**.** *Given a connected, directed and finite Eulerian graph $G = (V, E)$, the number of its Eulerian tours is*

$$\operatorname{et}(G) = t_s(G) \cdot \prod_{v \in V} (\deg(v) - 1)!,$$

where $\deg(v)$ is the out-degree (or in-degree) of vertex v and $t_s(G)$ is the number of spanning anti-arborescences of G rooted at vertex s.

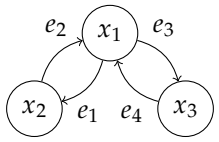
To calculate $t_s(G)$, we can use Kirchhoff's matrix-tree theorem for directed graphs, which can be easily generalized from the undirected version. Also, as demonstrated in Oleh's report [5], the selection of root s does not matter.

The theorem can also be written in an alternative form as follows:

Theorem 2 (BEST theorem (alternative form))**.** *Given a connected, directed and finite Eulerian graph $G = (V, E)$, the number of its Eulerian tours starting from s is*

$$\operatorname{et}_s(G) = t_s(G) \cdot \deg(s)! \cdot \prod_{v \in V \setminus \{s\}} (\deg(v) - 1)!.$$

The difference between $\mathrm{et}(G)$ and $\mathrm{et}_s(G)$ is that, a tour in $\mathrm{et}(G)$ has no starting vertex, while a tour in $\mathrm{et}_s(G)$ has a starting vertex s. For example, for the following graph, $\tau_1 = \langle e_1, e_2, e_3, e_4 \rangle$ and $\tau_2 = \langle e_3, e_4, e_1, e_2 \rangle$ are counted as different tours for $\mathrm{et}_s(G)$ but the same tour for $\mathrm{et}(G)$.



2 Computational Complexity

An algorithm to compute $\mathrm{et}_s(G)$ can thus be easily implemented using any programming language. The time complexity is determined by Gaussian elimination, so it is typically $O(|V|^3)$. A sample implementation in C++ can be found in my blog [10].

For special graphs, we also have more efficient implementations. For example, when $|E|$ is small, most of the elements in the Laplacian matrix are -1, and we can use a competitive programming technique (not widely) known as 'black box linear algebra' to speed up the calculation [2] [3] [8]. A sample implementation in C++ can be found here [4]. Alternatively, we may try wisely (or just randomly) swapping several rows during Gaussian elimination to create more columns of all 0s [6]. A sample implementation by me in C++ can be found here [11].

3 Proof of Correctness

However, for a course in mathematics, the proof is more important than the algorithm implementation. The following proof is partially inspired by [1].

Choose an arbitrary vertex s as starting vertex for all Eulerian tours. We are going to find a bijection between anti-arborescences T converging to s and sets $\mathcal{T}_T$ of Eulerian tours. Each set $\mathcal{T}_T$ will have size

$$d := \deg(s)! \cdot \prod_{v \in V \setminus \{s\}} (\deg(v) - 1)!.$$

- We can construct d Eulerian tours $\tau \in \mathcal{T}_T$ from an anti-arborescence T, and different anti-arborescences cannot generate the same Eulerian tour.

  Proof. For every vertex v, we number its $\deg(v)$ outgoing edges, such that the tree edge receives the highest number. For each vertex v ($v \neq s$), exactly one of its outgoing edges is a tree edge, and therefore we have $(\deg(v)-1)!$ ways to number them. For the root s, all of its outgoing edges must be non-tree edges, and therefore we have $\deg(s)!$ ways to number

them. Therefore, given a root s, we have in total d valid ways to number all the non-tree edges.

We can construct the Eulerian tour τ using the numbering as follows: Start with trail $\tau_1 = \langle s \rangle$. For a given trail $\tau_i = \langle s, \ldots, v_i \rangle$, select the smallest numbered unused outgoing edge from vertex v_i. Continue this process until all edges have been used, and denote the resulting trail as τ.

Since the graph is Eulerian, we will not encounter any dead ends until we eventually return to s. In other words, τ must be a circuit. Thus, to show τ is an Eulerian tour, we only need to show τ contains all edges.

For every vertex u ($u \neq s$), if there is an edge (u, v) that has not been used, since the in-degree of v is equal to the out-degree, then v also has an outgoing edge that has not been used, so the tree edge of v has not been used either... The tree edges finally lead us to s, then s also has an outgoing edge that has not been visited, which contradicts the termination condition. Therefore, no such edge is missed, and thus τ is indeed an Eulerian tour.

Since different numberings lead to different Eulerian tours, we can construct d Eulerian tours from an anti-arborescence, i.e., $|\mathcal{T}_T| = d$. Also, since different anti-arborescences lead to different numberings, they cannot generate the same Eulerian tour. □

- We can construct an anti-arborescence T_τ from an Eulerian tour τ, such that $\tau \in \mathcal{T}_{T_\tau}$.

  Proof. Each Eulerian tour τ corresponds to a unique numbering of edges. Thus, we can construct the anti-arborescence T_τ by selecting the outgoing edge with the largest number for every vertex v ($v \neq s$).

  Apparently, there are $|V| - 1$ tree edges. Thus, to show T_τ is an anti-arborescence, we only need to show that starting from any vertex $u (u \neq s)$, the selected edges will finally lead us to s.

  Suppose there is a selected edge (u, v). As this is the last outgoing edge from vertex u and $u \neq s$, the edge cannot form a loop, i.e., $v \neq u$. If $v = s$, we have shown that from vertex u, the edges in T_τ can lead us to s. If $v \neq s$, the tree edge (v, w) must have a larger number than (u, v) and $w \neq u$. At each step, the tree edge either leads us to s or to an unvisited vertex. Eventually, we will reach s.

  Hence, the selected edges form a spanning anti-arborescence.

  Since T_τ is constructed based on the numbering of edges in τ, it is always possible to reconstruct τ from T_τ by correctly numbering the non-tree edges. That is, $\tau \in \mathcal{T}_{T_\tau}$. □

Now, we have shown the correctness of Theorem 2. For Theorem 1, the original version, we only need to decide the first outgoing edge from s to avoid

repeated tours like τ_1 and τ_2. In other words, we can obtain $\mathrm{et}(G)$ from $\mathrm{et}_s(G)$ with

$$\mathrm{et}(G) = \frac{\mathrm{et}_s(G)}{\deg(s)}.$$

References

[1] Aigner, M. (2007). *A course in enumeration*. Springer.

[2] anta. (2015). 解説 No.310 2文字しりとり - yukicoder. Retrieved 2024-05-30, from <https://yukicoder.me/problems/no/310/editorial>.

[3] anta. (2015). Black box linear algebra Wiki - yukicoder. Retrieved 2024-05-30, from <https://yukicoder.me/wiki/black_box_linear_algebra>.

[4] anta. (2015). #62132 (C++11) No.310 2文字しりとり - yukicoder. Retrieved 2024-05-30, from <https://yukicoder.me/submissions/62132>.

[5] Dmytruk, O. (2024). Kirchhoff's matrix-tree theorem. Retrieved 2024-05-30, from <https://www.math.nagoya-u.ac.jp/~richard/teaching/s2024/SML_Oleh_2.pdf>.

[6] kmjp. (2016). yukicoder : No.310 2文字しりとり - kmjp's blog. Retrieved 2024-05-30, from <https://kmjp.hatenablog.jp/entry/2016/01/06/0900>.

[7] Richard, S. (2024). Graph theory. Retrieved 2024-05-30, from <http://www.math.nagoya-u.ac.jp/~richard/teaching/s2024/Graph.pdf>.

[8] Wiedemann, D. (1986). Solving sparse linear equations over finite fields. *IEEE Transactions on Information Theory, 32*(1), 54–62.

[9] Wikipedia. (2024). BEST theorem - Wikipedia. Retrieved 2024-05-30, from <https://en.wikipedia.org/wiki/BEST_theorem>.

[10] Yang, M.-T. (2018). [BZOJ3659]Which Dreamed It - skylee. Retrieved 2024-05-30, from <https://www.cnblogs.com/skylee03/p/10150343.html>.

[11] Yang, M.-T. (2018). #305581 (C++11) No.310 2文字しりとり - yukicoder. Retrieved 2024-05-30, from <https://yukicoder.me/submissions/305581>.