Cyclic factorization of complete graphs into spanning trees with an Euler trail

Alex Horn

Abstract

A simple graph algorithm is presented which decomposes an n-complete graph of even order into edge-disjoint, isomorphic copies of a spanning tree whose edges form an Euler trail of length $n-1$. While more general methods of such spanning tree factorizations have been known, we emphasize here its simple algorithmic treatment with an efficient data structure.

1 Introduction

This document is written mainly for fun and it concerns a graph decomposition algorithm which computes the cyclic spanning tree factorization of complete graphs of even order. This algorithm could find application in ad hoc network designs in which link failures need to be restored quickly by creating a new isomorphic yet link-disjoint network structure.

We consider only simple and finite graphs. Let G be such a graph. A **decomposition** of G is a set of pairwise edge-disjoint subgraphs $H_1, H_2, \dots H_k$ such that every edge of G belongs to exactly one subgraph H_i for $1 \leq i \leq k$. If every such subgraph is isomorphic to a graph H, then G is said to have an H**-decomposition**. An H-decomposition into $H_1, H_2, \dots H_k$ is called **cyclic** if there exists an ordering $(x_1, x_2, \dots, x_n)$ of the vertices of G together with graph isomorphisms $\phi_i : H_1 \to H_i$, for all $1 < i \leq k$, such that $\phi_i(x_j) = x_{(i+j) \bmod n}$ for all $1 \leq j \leq n$. If H has the same order as G and none of the vertices in H are isolated, then H is a connected factor and the decomposition is called an H**-factorization**. In particular, if H is a spanning tree, we say that graph G has a **spanning tree factorization**. Figure 1 (p. 2) shows cyclic decompositions of the 6-complete graph and 8-complete graph which can be verified by "rotating" the subgraphs twice. Note that only the centre and most right decompositions are spanning tree factorizations.

Many real-life applications such as network designs and combinatorial problems come in the disguise of complete graph decompositions. Graph theory studies their underlying properties. For example, when is the decomposition of a complete graph into cycles of length m possible? Since the existence of such a decomposition requires the degree of every vertex to be even, the complete graph must be of odd degree. Otherwise, a 1-factor can be removed from the

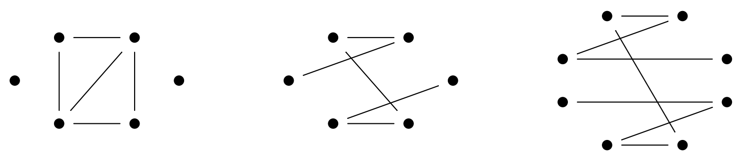


Figure 1: Cyclic decomposition of K_6 into a subgraph whose edges form an Euler trail (left). Similarly, K_6 and K_8 have also a cyclic spanning tree factorization whose edges form an Euler trail (centre, right).

$2n$-complete graph. Then, an m-cycle decomposition exists if and only if the length of the cycle, m, divides the number of edges in the graph [1], [7].

In contrast, no necessary and sufficient condition is known for a tree to admit a factorization of complete graphs [6]. Rather, most factorizations are subject to graph labeling techniques (see below). In general, a label is a function from the set of vertices into the set of natural numbers. Informally, these labels are chosen such that every edge appears in exactly one factor. In particular, Rosa-type labelings (see [2] for a recent survey) are being used for cyclic decompositions of complete graphs into spanning trees [3], [5], [4].

In the sequel, we automate the labeling of the vertices in an Euler trail such that it admits a cyclic decomposition into spanning trees of a complete graph. In fact, cyclic spanning tree factorizations can be completely characterized in terms of their symmetrical structure [3], [4]. Before we can make this characterization precise, we must agree on the following definitions.

Definition 1.1. Define a **labeling** of a graph G with m edges to be an injective function $\lambda : V(G) \to L$ where $L = \{0, 1, \ldots, 2m\}$. The **length** of an edge $(x, y) \in E(G)$ is defined by $\ell(x, y) := min\{|\lambda(x) - \lambda(y)|, 2m + 1 - |\lambda(x) - \lambda(y)|\}$. If $\{\ell(x, y) \mid (x, y) \in E(G)\} = \{1, 2, \ldots, m\}$, then λ is called a **ρ-labeling**; if the image of λ is moreover a subset of $\{0, 1, \ldots, m\}$, then λ is a **graceful labeling**. A graph which admits a graceful labeling is said to be **graceful**.

Definition 1.2. A connected graph G is **symmetric** if it has a bridge (x, y) and there exists an automorphism ϕ such that $\phi(x) = y$ and $\phi(y) = x$. The isomorphic connected components of $G - (x, y)$ are called **banks** and denoted by H and H' respectively. A labeling of a symmetric graph G with $2n - 1$ edges and banks H and H' is **ρ-symmetric graceful** if H has a ρ-labeling and $\lambda(\phi(x)) = \lambda(x) + n$ for each vertex $x \in V(H)$. A labeling of a symmetric graph with $2n - 1$ edges is said to be **symmetric graceful** if it is ρ-symmetric graceful and the bank H is moreover graceful.

Theorem 1.3 ([3],[4]). *Let G be a symmetric graph with $2n - 1$ edges. Then, K_{2n} has a cyclic G-decomposition if and only if G is ρ-symmetric graceful.*

In the next section, we present a polynomial time graph algorithm which constructs the symmetric graceful spanning tree of K_{2n} with the additional constraint that this spanning tree must have an Euler trail. Clearly, the length of this Euler trail is bound to be odd $(2n - 1)$. Moreover, the spanning tree which contains the Euler trail is unique up to isomorphism.

2 Algorithm

Given the $2n$-complete graph, we aim at finding the spanning tree S with an Euler trail such that K_{2n} has a cyclic S-decomposition. This spanning tree factorization is computed by Algorithm 1 and Algorithm 2 (see p. 4). More precisely, Algorithm 1 computes the symmetric labeling of the Euler trail such that a cyclic spanning tree factorization exists by theorem (1.3). Then, Algorithm 2 creates the cyclic permutations of this Euler trail.

Algorithm 1 Finds the spanning tree S with an Euler trail such that K_{2n} has a cyclic S-decomposition. The Euler trail in S is formed by the edges incident to the vertex labeled k and vertex labeled $trails[k]$ where $0 \leq k < 2n$. The trail can be traversed by starting at the vertex labeled 0 and stops when $trail[k] = k$ for some $0 \leq k < 2n$.

```
Require: $N$ to be even and $N \geq 2$
Require: $trail[k] = k$ for all $0 \leq k < N$
  $i \leftarrow 0$
  $j \leftarrow N/2 - 1$
  $x \leftarrow N - 1$
  $y \leftarrow N/2$
  loop
    $trail[i] \leftarrow j$
    $trail[x] \leftarrow y$
    $i \leftarrow i + 1$
    $y \leftarrow y + 1$
    if $i = j$ then
      exit loop
    end if
    $trail[j] \leftarrow i$
    $trail[y] \leftarrow x$
    $j \leftarrow j - 1$
    $x \leftarrow x - 1$
    if $i = j$ then
      exit loop
    end if
  end loop
  if $N > 2$ then
    $trail[i] \leftarrow y$
  end if
```

The following runtime complexity result is obvious:

Proposition 2.1. *Algorithm 1 is $O(n)$ given the $2n$-complete graph.*

Proof. Given K_{2n}, the loop iterates n times plus a final constant operation. ■

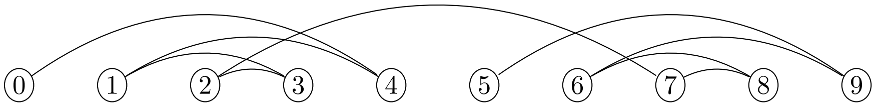
Algorithm 2 Computes the cyclic permutations of the symmetric spanning tree of K_{2n} (see Algorithm 1). For each $0 \le s < n$, the tree traversal starts at the vertex labeled $trail[s][s]$ and stops when $trail[s][t] = t$ for some $0 \le t < 2n$.

```
Require: $N$ to be even and $N \geq 2$
Require: $trail[s][k] = k$ for all $0 \leq s < \frac{1}{2}N$ and $0 \leq k < N$
  Store result of Algorithm 1 into $trails[0]$
  for $i = 1$ to $\frac{1}{2}N - 1$ do
    for $j = 0$ to $N - 1$ do
      $v \leftarrow (trails[i-1][j] + 1) \bmod N$
      $trails[i][(j+1) \bmod N] \leftarrow v$
    end for
  end for
```

Proposition 2.2. *Algorithm 2 is $O(n^2)$ given the $2n$-complete graph.*

Proof. By proposition (2.1), Algorithm 1 is $O(n)$. Afterwards, the innermost loop executes $2n$ times. The outermost loop executes this loop $n-1$ times counting a total of $2(n^2 - n)$ iterations. Therefore, Algorithm 2 is $O(n^2)$. ■

Both algorithms feature an efficient key-indexed array data structure which represents vertex labels and supports the traversal of the spanning trees.



v	0	1	2	3	4	5	6	7	8	9
$trail[v]$	4	3	7	2	1	5	9	8	6	5

Figure 2: The diagrammatic and key-indexed array representation of the spanning tree with an Euler trail for the 10-complete graph. This Euler trail is formed by joining the vertices labeled as $0, 4, 1, 3, 2, 7, 8, 6, 9, 5$. More generally, notice that $trail[v]$ is the label of the vertex adjacent to the vertex labeled v. If $trail[v] = v$, then the end of the trail has been reached. According to definition (1.2), the spanning tree is symmetric because the subgraphs formed by the edges of the trail $0, 4, 1, 3, 2$ and $7, 8, 6, 9, 5$ are two isomorphic banks joined by the bridge between the vertices labeled 2 and 7. Since the first bank is graceful, we conclude that the entire spanning tree is symmetric graceful.

v	0	1	2	3	4	5	6	7	8	9
$trail[1][v]$	6	5	4	8	3	2	6	0	9	7
$trail[2][v]$	8	7	6	5	9	4	3	7	1	0
$trail[3][v]$	1	9	8	7	6	0	5	4	8	2
$trail[4][v]$	3	2	0	9	8	7	1	6	5	9

Figure 3: Given the trail from Figure 2 (p. 4), Algorithm 2 (p. 4) computes the remaining four trails of length nine by applying cyclic permutations.

Example 2.3. Figure 2 (p. 4) illustrates the spanning tree labeling produced by Algorithm 1 (p. 3) of the 10-complete graph. This labeling is symmetric graceful by definition (1.2). Therefore, by theorem (1.3), the algorithm produced a cyclic spanning tree factorization of the 10-complete graph. Algorithm 2, in turn, creates the cyclic permutations of the trail to obtain the remaining four Euler trails of equal length which partition the edge set accordingly.

3 Conclusion

In summary, by relying on more recent graph theoretical results based on ρ-labelings [3], [4], a simple polynomial time graph algorithm was presented that constructs a spanning tree factorization of the $2n$-complete graph such that the spanning tree has an Euler trail.

References

[1] B. Brian Alspach and H. Gavlas. Cycle decompositions of K_n and $K_n - I$. *Journal of Combinatorial Theory*, 81(1):77–99, 2001.

[2] S. El-Zanati and C. Eynden. On rosa-type labelings and cyclic graph decompositions. *Mathematica Slovaca*, 59(1):1–18, 2009.

[3] P. Eldergill. Decompositions of the complete graph with an even number of vertices. Master's thesis, McMaster University, Hamilton, 1997.

[4] D. Fronček. Cyclic decompositions of complete graphs into spanning trees. *Discussiones Mathematicae. Graph Theory*, 24(2):345 – 353, 2004.

[5] D. Fronček and M. Kubesa. Factorizations of complete graphs into spanning trees. *Congressus Numerantium*, 154:125–134, 2002.

[6] P. Kovář. Decompositions and factorizations of complete graphs. In M. Dehmer, editor, *Structural Analysis of Complex Networks*, pages 169–196. Birkhäuser Boston, 2011.

[7] M. Šajna. Cycle decompositions III: Complete graphs and fixed length cycles. *Journal of Combinatorial Designs*, 10(1):27–78, 2002.