Partition of an Eulerian circuit search problem for the complete graph of order 15

Shuji Jimbo
Guest Researcher, Graduate School of Natural Science and Technology, Okayama University

E-mail address: jimbo-s@okayama-u.ac.jp

Abstract

For odd integers n greater than or equal to 15, it is known how to construct an Eulerian circuit of the complete graph of order n whose shortest subcycle length is $n-4$. Furthermore, the author and others have proved that there is no Eulerian circuit of the complete graph of order n whose shortest subcycle length is greater than $n-2$. The author and others conjecture that, for every odd integer n greater than or equal to 15, there is no Eulerian circuit of the complete graph of order n whose shortest subcycle length is $n-3$. As part of the proof of the conjecture, the author and others aim to prove that there is no Eulerian circuit of a complete graph of order 15 whose shortest subcycle length is 12. Currently, we expect that the conjecture above for $n = 15$ can be proved through large-scale distributed processing. For distributed processing to be effective, the size of each divided subproblem must be small enough to fit into the main memory. In this report, we describe the methods used to achieve this goal and discuss the possibility of applying these methods to complete the proof.

Keywords. Eulerian circuit, computer experiment, search space, distributed processing.

1 Introduction

Some of the terms used in this paper are described in Section 2. For other terms related to graph theory, please refer to Wilson's graph theory textbook[3].

The shortest subcycle length $s(C)$ for an Eulerian circuit C of an Eulerian graph is defined as

$$s(C) = \min\{(j-i) \bmod m \mid v_i = v_j\}.$$

Let G be an Eulerian graph. The maximum length of the shortest subcycle of G is called the Eulerian recurrence length of G and is denoted by $e(G)$. Expression K_n denotes the complete graph of order n, that is the complete graph consisting of n vertices. For odd integers n greater than or equal to 15, it is known how to construct an Eulerian circuit of K_n, whose shortest subcycle length is $n-4$[1]. Furthermore, the author and others have proved that there is no Eulerian circuit of K_n whose shortest subcycle length is greater than $n-2$[1].

The authors conjecture that, for every odd integer n greater than or equal to 15, there is no Eulerian circuit of K_n whose shortest subcycle length is $n-3$. If this conjecture holds, then $e(K_n)$, the Eulerian recurrence length of K_n is $n-4$ for every odd integer n greater than or equal to 15. As part of the proof of the conjecture, the authors aim to prove that there is no Eulerian circuit of K_{15} whose shortest subcycle length is 12. To this end, at last year's workshop, we proposed a method to reduce the search space by adding constraints on trails of complete graphs. However, no usefulness of such a method has been found at this time[2].

Currently, we expect that the above proofs can be achieved through large-scale distributed processing. For distributed processing to be effective, the size of each divided subproblem must be small enough to fit into main memory. In this report, we describe the methods used to achieve this goal and discuss the possibility of applying these methods to complete the proof.

2 Preliminaries

Let W be a walk of K_{15}. We say that W satisfies *condition* P_{12}, if for any sub-walk $v_0 \to v_1 \to v_2 \to \cdots \to v_{11}$ of W of length 11, v_0, v_1, $\ldots$, and v_{11} are distinct. A walk that satisfies condition P_{12} is called a P_{12} walk.

Hereafter, the vertex set of K_{15} is described as $V = \{0, 1, 2, \ldots, 14\}$. If there is an Eulerian circuit of K_{15} whose shortest subcycle length is 12, then the following trail must occur by applying some permutation on the vertex set of K_{15}:

$$I(15) = 13 \to 0 \to 1 \to 2 \to \cdots \to 11 \to 0 \to 12.$$

The trail $I(15)$ above is referred to as the *initial trail*. In the trail obtained by extending the initial trail, the position of the last vertex 12 on the initial trail is defined as 0, and, for any positive integer k, the position of the vertex reached from the vertex at position 0 forward through k edges is defined as k.

For each vertex $v \in V$, $\chi(v)$ denotes the set consisting of all vertices w in V such that w is not adjacent to v in the initial trail. For example, $\chi(14) = V - \{14\}$, $\chi(0) = V - \{1, 11, 12, 14\}$ holds.

3 Division of the search space

Consider dividing the entire search space consisting of P_{12} trails into a large number of mutually disjoint sets of trails to process the search in a distributed manner. Hereafter, each set of divided P_{12} trails is referred to as a *segmented set*. Determine the set S of invariant vertices by the mapping φ for the definition of a segmented set. Sets $\{0, 6, 14\}$, $\{0, 2, 4, 6, 8, 10, 14\}$, etc. can be considered as the set S above. The vertices belonging to the set S are hereinafter referred to as *pivot vertices*. The set consisting of all pivot vertices is denoted by S_P. Set S_P must be chosen so that the maximum size of a set in the segmented sets is sufficiently small. Initially, the computer experiment proceeds as $S_P = \{0, 2, 6, 10, 14\}$, and when the need arises to reduce the maximum size of a set in the segmented sets to a smaller size, $S_P = \{0, 2, 4, 6, 8, 10, 14\}$.

A segmented set is defined as the set of trails divided by the combination of whether each vertex adjacent to pivot vertices appears in the first half or the second half of an Eulerian circuit. In the definition of a segmented set, the combination of the set of vertices adjacent to each pivot vertex that appears in the first half of the Eulerian circuit is called the *segmentation parameter* of the segmented set. A combination parameter is described as follows:

$$P = \{(v_0, S_0), (v_1, S_1), \dots, (v_k, S_k)\}. \tag{1}$$

In the above expression, the set consisting of all pivot vertices is $S = \{v_1, v_2, \dots, v_k\}$, and each S_i is the set consisting of all vertices adjacent to the pivot vertex v_i that appear in the first half, where the size of S_i is even. Let $p(P, v_i)$ denote the S_i in the expression above. In the expression above, a pivot vertex may belong to an S_i. The segmented set consisting of all P_{12} trails of length m subject to segmentation parameter P is denoted by $W(P, m)$.

Let P be a segmentation parameter. For two pivot vertices a and b, we say that the mutual inclusion relations of a and b are different when the truth-values of the propositions $a \in p(P, b)$ and $b \in p(P, a)$ are different. A segmentation parameter for which there is exactly one pair of pivot vertices with different mutual inclusion relations is called a singular parameter. The vertices at positions 45 and 46 on the P_{12} trail W subject to the singular parameter P must both be specific pivot vertices. For example, if pivot vertex b is adjacent to pivot vertex a and pivot vertex b is not adjacent to pivot vertex a in the first half of the Eulerian circuit, then vertex a must appear at position 45 and vertex b must appear at position 46.

Let P be a segmentation parameter. Assume that there is an Eulerian circuit that satisfies the condition P_{12}. Let $W \in W(P, 59)$ denote the first half of C, and W' the sub-trail of length 60 in C whose initial vertex is the final vertex of W, namely the second half of C. Then the segmentation parameter of the segmented set to which $\varphi(W')$ belongs must be

$$\rho(P) = \{(\varphi(v_0), \chi(\varphi(v_0)) - \varphi(S_0)), (\varphi(v_1), \chi(\varphi(v_1)) - \varphi(S_1)), \dots, (\varphi(v_k), \chi(\varphi(v_k)) - \varphi(S_k))\}. \tag{2}$$

Note that $\rho(\rho(P)) = P$ holds. Also, if either 0 or 14 is a pivot vertex, then $\rho(P) = P$ does not hold for any segmentation parameter P, since, for both vertices $v \in \{0, 14\}$, the numbers of vertices adjacent to vertex v in the first half and second half of the Eulerian circuit are different.

Let Y be $E(K_{15}) - E(I(15))$. Set Y consists of all the edges of K_{15} that are not included in the initial trail $I(15)$, and $|Y| = 91$. Assign a random number $r(e)$ to each edge e belonging to Y in advance, and define a hash value $Z(X)$ for a subset X of Y as the exclusive OR of the random numbers corresponding to the edges belonging to X. Precisely, $Z(X)$ is defined by the following expression:

$$Z(X) = \bigoplus_{e \in X} r(e).$$

The hash value of a trail W obtained by extending the initial trail $I(15)$ is computed as $Z(E(W) - E(I(15)))$.

A method is proposed below to determine whether or not an Eulerian circuit satisfying the condition P_{12} can be obtained by extending a trail belonging to the segmented set

$W(P, 60)$ for a given segmentation parameter P. In this method, both the segmentation parameters P and $\rho(P)$ are processed in pairs.

Perform a depth-first search for the segmentation parameters $\rho(P)$ and P consecutively. First, register the hash value $Z(Y - \varphi(E(W)))$ of each trail $W \in W(\rho(P), 60)$ obtained in the first depth-first search in the hash table H. Then, search from H the hash value $Z = Z(E(W) - Y)$ of each trail $W \in W(P, 59)$ obtained in the second depth-first search. If the hash value $z = Z(E(W) - Y)$ of the trail $W \in W(P, 59)$ is contained in H, we say that W is a false positive for the existence of an Eulerian circuit satisfying condition P_{12}. Hereafter, the statement "W is a false positive for the existence of an Eulerian circuit satisfying condition P_{12}." is simply written as "W is a false positive." Once W is found as evidence for the existence of an Eulerian circuit satisfying condition P_{12}, a regular depth-first search is performed using W as the initial trail to determine whether it is possible to construct an Eulerian circuit that satisfies condition P_{12}. Since finding a false positive trail $W \in W(P, 59)$ is expected to be very rare for any segmentation parameter P, we expect that processing by the proposed method will not cause a significant increase in overall computation time.

During the depth-first search, it is desirable to detect as soon as possible that the current trail W cannot be extended to a trail belonging to the segmented set $W(P, 60)$. If such a situation is detected, the depth-first search is immediately forced to backtrack to make the search more efficient. For this purpose, the depth-first search immediately backtracks as soon as it is determined that the edge specified as having to be chosen by the segmentation parameter P can no longer be chosen. In detail, do the following. One occurrence of a pivot vertex on a trail results in the appearance of two edges incident with the pivot vertex. Let n_v be the number of times v can appear on the remaining trails at time point t when the next vertex is reached after passing the pivot vertex v during the depth-first search. Furthermore, let n_e be the number of edges incident with v, which has not yet appeared at time point t but whose appearance is specified by P. If $2n_v < n_e$ holds, then it is impossible to extend the current trail to a trail belonging to $W(P, 60)$. The depth-first search therefore immediately backtracks at time point t.

When the segmented set to be searched is specified by the singular parameter P, the following modifications are made to the depth-first search behavior. In the following, assume that there exist vertices a and b satisfying $a \in p(P, b)$ and $b \notin p(P, a)$ without loss of generality. In this case, there must appear vertex a at position 45 and vertex b at position 46 on any trail $W \in W(P, 60)$. Register edge ab as passed before the search, set the maximum depth of depth-first search to 58, i.e., to position 44, and then perform depth-first search. When the search reaches its maximum depth, if a trail of length 60, obtained by further adding vertex a at position 45 and vertex b at position 46, belongs to $W(P, 60)$, it has successfully generated a trail belonging to $W(P, 60)$. Otherwise, the depth-first search is forced to backtrack because it fails to generate a trail belonging to $W(P, 60)$.

4 Concluding remarks

Let $e(n)$ denote the Eulerian recurrence length of K_n. It is known as a previous result that $n - 4 \leqq e(n) \leqq n - 3$ for all odd integers $n \geqq 15$. The author and others conjecture that $e(n) = n - 4$ for all odd integers $n \geqq 15$. Especially, $e(15) = 11$ should be proved by brute force search. However, the search space seems to be too huge to determine $e(15)$ by single DFS on an ordinary PC.

In this paper, a method to divide the search problem into a large number of subproblems has been proposed. The set of all the subproblems might be solved by large-scale distributed processing. A parameter P to divide the problem is a set of pairs of a pivot vertex v and a set S of vertices adjacent to the pivot vertex. All the edges that join v and a vertex in S must appear on the first half of an Eulerian circuit.

Let $W(P, 60)$ denote the set of all trails subject to parameter P obtained by extending the initial trail $I(15) = 13 \to 0 \to 1 \to 2 \to \cdots \to 11 \to 0 \to 12$. Mapping $\varphi : V(K_{15}) \to V(K_{15})$, is defined as $\varphi(13) = 12$, $\varphi(12) = 13$, $\varphi(0) = 0$, $\varphi(14) = 14$, and $\varphi(i) = 12 - i$ for $i \in \{1, 2, \ldots, 11\}$. By searching $W(P, 60)$ and $W(\varphi(P), 60)$, we can determine whether or not there is an Eulerian circuit C of K_{15} that satisfies condition P_{12}.

The computational complexity to solve all the divided subproblems is expected to be considerably larger than the one to solve the original problem with a single depth-first search. Preliminary experiments should provide an accurate estimate of the computational complexity required to solve all the divided subproblems and find an effective dividing method.

Acknowledgment

This work was supported by the Research Institute for Mathematical Sciences, an International Joint Usage/Research Center located in Kyoto University.

References

[1] Jimbo, Shuji, and Akira Maruoka. "The Upper Bound on the Eulerian Recurrent Lengths of Complete Graphs Obtained by an IP Solver." International Workshop on Algorithms and Computation. Springer, Cham, 2019.

[2] Jimbo, Shuji. "A Study of Constraints on Eulerian Circuits." Logic, Algebraic system, Language and Related Areas in Computer Science. RIMS Kôkyûroku 2229, pp.81-87, 2022.

[3] Wilson, Robin J.. "Introduction to Graph Theory." 5th Ed., ISBN 9780273728894, Longman, 2010.