Vol.2010-AL-131 No.7
2010/9/22

オイラー路の列挙

菊 地 洋 右[†1]

単純グラフ G のすべての辺を含む路をオイラー路、オイラー路が閉路となっているものをオイラー回路という。オイラー路をもつグラフを semi-eulerian とよび、オイラー回路をもつグラフをオイラーグラフとよぶ。オイラーグラフの特徴付けはグラフ理論の教科書に必ず載っていると言っていいほどよく知られている。オイラーグラフが与えられたときに、オイラー回路の数え上げは#P-完全であることが知られている。本研究は、単純グラフ G がオイラー路をもつとき、重複も抜けもなく、そのオイラー路を列挙するアルゴリズムを提案する。

本研究のアルゴリズムではまず Fleury's Algorithm を用いて単純グラフ G のオイラー路を求める。 このオイラー路から順次、オイラー路を求めていくことで列挙を行う。提案するアルゴリズムは、Fleury's Algorithm を適用した後に、すべてのグラフ的列を 1 つあたり $O(m)$ 時間で列挙する。

Enumerating All Eulerian Trails

Yosuke Kikuchi [†1]

For simple graph G, eulerian trail is a trail that has all edges in G. If eulerian trail is close circuit, it is called eulerian circuit. If G has a eulerian trail, G is called semi-eulerian and if G has a eulerian circuit, then G is eulerian graph. A characterization of eulerian graph is well-known and may appear in any graph theory. Given eulerian graph, counting eulerian circuits is $\#P-$complete. This paper will propose an algorithm to generate all eulerian trail for simple graph G, if such trail exists. At first, we obtain the minimum eulerian trail of G, applying Fleury's algorithm. Next, we generate all eulerian trails. Our algorithm generates all eulerian trails in $O(m^2)$ for each, after applying Fleury's algorithm, where m is the number of edge in G.

†1 津山工業高等専門学校情報工学科

1. はじめに

本研究では、与えられた連結グラフ G に対してそのオイラー路をすべて列挙するアルゴリズムについて考察する。単純グラフ G のすべての辺を含む路をオイラー路、オイラー路が閉路となっているものをオイラー回路という。連結グラフ G が与えられたときに、G がオイラー路をもつかどうかの判定は奇頂点が 2 個以下かどうかを調べればよい。連結グラフ G について G がオイラー路をもつための必要十分条件は G の奇頂点の数が高々2 であることであり、G がオイラー閉路をもつための必要十分条件は G の奇頂点の数が 0 であることである。また、Fleury's Algorithm を用いれば G のオイラー路が得られる。オイラー路の数え上げに関する研究では、Brightwell と Winkler[1] がオイラー閉路の数え上げが$\#P-$完全であることを示している。また、McKay と Robinson,[2] は完全グラフのオイラー路の数について漸近的な値を求めている。本研究では単純グラフのオイラー路を列挙することを考える。次の章で定義と用語を述べる。3 章では Fleury's Algorithm を改良して最小オイラー路を構成するアルゴリズムについて述べる。最小オイラー路については 2 章で定義する。3 章で得られた最小オイラー路から順にオイラー路を構成する方法について 4 章で述べ、5 章はまとめである。

2. 定　　義

グラフ G において頂点 u から v へのウォークは頂点 u から始まり v で終わる頂点と辺の交互列で、各辺の前後にある頂点はその辺と接続しているものをいう。これを u-v ウォークとよぶ。u-v ウォークにおいて各辺がちょうど 1 度ずつ現れているものを u-v 路という。単純グラフあっては各頂点間の辺数は高々1 なのでウォークを頂点だけの列で表せる。グラフ G のすべての辺を含む u-v 路をオイラー路とよぶ。特に $u = v$ であるオイラー路をオイラー回路とよぶ。グラフ G の頂点 v に対して, その接続している辺数を v の次数とよぶ。次数が奇数である頂点を奇頂点、次数が偶数である頂点を偶頂点とよぶ。また、次数が 0 である頂点を孤立点とよぶ。 図 3 において v_3 と v_4 はともに次数が 3 であるので奇頂点であり、その他の頂点は偶頂点である。よってこのグラフではオイラー路は存在するが、オイラー回路は存在しない。実際、頂点列 $(v_3, v_2, v_1, v_5, v_4, v_2, v_5, v_3, v_4)$ はこのグラフのオイラー路である。頂点列 $(v_3, v_2, v_1, v_5, v_4, v_3, v_5, v_2, v_4)$ や $(v_4, v_3, v_5, v_2, v_4, v_5, v_1, v_2, v_3)$ なども図 3 のグラフのオイラー路である。しかし、$(v_3, v_2, v_1, v_5, v_4, v_2, v_5, v_3, v_4)$ と $(v_4, v_3, v_5, v_2, v_4, v_5, v_1, v_2, v_3)$ は頂点の並びが逆になっているだけである。ここでは、このような場合は同一のオイラー路と

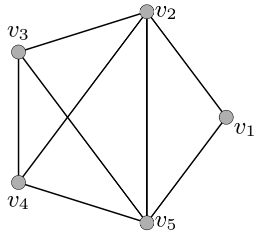

図 1 オイラー路をもつグラフの例

して扱うことにする。図 2 においては全ての頂点の次数は偶数である偶頂点である。よってこのグラフではオイラー回路が存在する。実際、頂点列 $(v_3, v_2, v_1, v_5, v_4, v_2, v_5, v_3)$ はこのグラフのオイラー路である。頂点列 $(v_3, v_2, v_4, v_5, v_1, v_2, v_5, v_3)$、$(v_3, v_5, v_2, v_4, v_5, v_1, v_2, v_3)$、$(v_4, v_2, v_5, v_3, v_2, v_1, v_5, v_4)$ なども図 2 のグラフのオイラー路である。しかし、これらの回路において $(v_3, v_2, v_1, v_5, v_4, v_2, v_5, v_3)$ と $(v_3, v_5, v_2, v_4, v_5, v_1, v_2, v_3)$ は頂点の並びが逆になっているだけであり、$(v_3, v_2, v_1, v_5, v_4, v_2, v_5, v_3)$ と $(v_4, v_2, v_5, v_3, v_2, v_1, v_5, v_4)$ は頂点列の始まりが違うだけである。これらも同一のオイラー回路として扱う。

図 3 のグラフにおいて添え字だけを取り出して、その頂点のラベルとしよう。

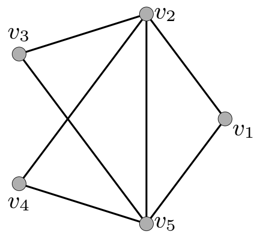

図 2 オイラー回路をもつグラフの例

すると、オイラー路 $(v_3, v_2, v_1, v_5, v_4, v_2, v_5, v_3, v_4)$ は $(3, 2, 1, 5, 4, 2, 5, 3, 4)$ となり、$(v_3, v_2, v_1, v_5, v_4, v_3, v_5, v_2, v_4)$ は $(3, 2, 1, 5, 4, 3, 5, 2, 4)$ となる。今後、頂点のラベルは整数として考えることにする。まず、このラベルに基づいてオイラー路の表記について次のように定める。与えられたグラフ G のオイラー路の表記は同一のオイラー路を表す表記の中で最小のものとする、これをオイラー路の代表表記とよぶ。つまり、$(a_1, a_2, \ldots, a_m)$ がオイラー路の代表表記とは任意の同一のオイラー路を表す $(a'_1, a'_2, \ldots, a'_m)$ に対して、ある整数 k が存在して、$i < k$ なる i に対して $a_i = a'_i$ かつ $a_k \leq a'_k$ であるときをいう。先に述べたように図 3 のグラフに対して、オイラー路 $(3, 2, 1, 5, 4, 2, 5, 3, 4)$ と $(4, 3, 5, 2, 4, 5, 1, 2, 3)$ を同一のものとして扱うとした。このオイラー路の代表表記は $(3, 2, 1, 5, 4, 2, 5, 3, 4)$ となる。次に、オイラー路の代表表記について順序を定める。与えられたグラフ G の異なるオイラー路の代表表記 $(a_1, a_2, \ldots, a_m)$ と $(b_1, b_2, \ldots, b_m)$ に対して、$(a_1, a_2, \ldots, a_m) \leq (b_1, b_2, \ldots, b_m)$ であるとはある整数 k が存在して、$i < k$ なる i に対して $a_i = b_i$ かつ $a_k \leq b_k$ であるときをいう。特にある整数 k が存在して、$i < k$ なる i に対して $a_i = b_i$ かつ $a_k < b_k$ であるときは $(a_1, a_2, \ldots, a_m) < (b_1, b_2, \ldots, b_m)$ と表す。図 3 のグラフのオイラー路 $(3, 2, 1, 5, 4, 2, 5, 3, 4)$ と $(3, 2, 1, 5, 4, 3, 5, 2, 4)$ では $(3, 2, 1, 5, 4, 2, 5, 3, 4) \leq (3, 2, 1, 5, 4, 3, 5, 2, 4)$ である。この順序によって、グラフが与えられたときに、そのオイラー路の集合に対して全順序が定義された。

補題 1 定義した順序によって、与えられたグラフのオイラー路の集合は全順序集合となる。 ■

補題 1 から与えられたグラフのオイラー路の集合には最小元が存在する。それを最小オイラー路とよぶ。図 3 のグラフにおいては最小オイラー路は $(3, 2, 1, 5, 4, 2, 5, 3, 4)$ である。

数列 $A = (a_1, a_2, \ldots, a_m)$ に対する操作として flipping を定義する。flipping $\mathrm{fl}(A, (a_i, a_j))$ とは $A = (a_1, a_2, \ldots, a_i, \ldots, a_j \ldots, a_m)$ に対して、数列 $(a_1, a_2, \ldots, a_{i-1}, a_j, a_{j-1}, \ldots, a_{i+1}, a_i, a_{j+1}, \ldots, a_m)$ を構成する操作である。つまり数列 A に対して、その部分数列 $(a_i, \ldots, a_j)$ を反転させた数列を新たに作る操作である。

数列 $A = (a_1, a_2, \ldots, a_m)$ に対して、$a_i = a_j$ かつ任意の $h > i, k > i$ 対して $a_h \neq a_k$ であるような (a_i, a_j) を最後尾同一要素対 $\mathrm{lpe}(A)$ という。また、2-$\mathrm{lpe}(A)$ とは A から a_i を除いて得られる最後尾同一要素対とする。$A = (3, 2, 1, 5, 4, 2, 5, 3, 4)$ においては先頭から 5 番目と 9 番目の要素 4 が最後尾同一要素対であり、2-$\mathrm{lpe}(A)$ は 4 番目と 7 番目の要素 5 である。

数列 $A = (a_1, a_2, \ldots, a_m)$ に対して、$a_i = a_j$,$a_{i+1} < a_{j-1}$ かつ任意の $h > i, k > i$ 対して $a_h \neq a_k$ であるような (a_i, a_j) を最後尾昇順同一要素対 $\mathrm{lpe}_<(A)$ という。また、2-$\mathrm{lpe}_<(A)$ とは A から a_i を除いて得られる最後尾順同一要素対とする。$A = (3, 2, 1, 5, 4, 2, 5, 3, 4)$ においては先頭から 4 番目と 7 番目の要素 5 が最後尾昇順同一要素対である。

Vol.2010-AL-131 No.7
2010/9/22

3. 最小オイラー路を構成するアルゴリズム

Fleury's Algorithm はオイラー路を構成するアルゴリズムとして知られている。このアルゴリズムをもとにした最小オイラー路を構成するアルゴリズムは次の通りである。

Min_euler

$T = \emptyset$, $Num = 1$ とする。

1. 奇頂点 v を 1 つ選ぶ (奇頂点がない場合は偶頂点を 1 つ選ぶ)。
   v を cv とおく。v に 1 を割り当てる。v を T に加える。Num の値を 1 増やす。
2. cv に接続する辺の中で橋でないものの集合を E_{cv} とする。
3. $E_{cv} = \emptyset$ のときは橋 $e = (cv, x)$ を選ぶ。それ以外のときは 6 へ。
4. x が番号をもっているならば、cv を x に更新し、x を T に加える。
5. x が番号をもっていないならば、cv を x に更新し、x を T に加える。x に Num の値を割り当てる。Num の値を 1 増やす。10 へ。
6. E_{cv} のどの辺に対して、cv とは異なる端点が番号をもたない場合は、E_{cv} から任意の辺 $e = (cv, w)$ を選ぶ。それ以外のときは 8 へ。
7. cv を w に更新する。w を T に加える。w に Num の値を割り当てる。Num の値を 1 増やす。10 へ。
8. E_{cv} で cv とは異なる端点が最小の番号もつ辺 $e = (cv, u)$ を選ぶ。
9. cv を u に更新する。u を T に加える。10 へ。
10. e を削除し、孤立点を削除する。
11. 2 から 9 の操作ができなくなるまで繰り返す。

上のアルゴリズムで与えられたグラフ G がアルゴリズム終了後に空になれば、G はオイラー路 T をもつ。空にならないときは G はオイラーグラフではない。上記のアルゴリズムで頂点には番号付けがされている。この番号をグラフの頂点のラベルとする。するとオイラー路 T はこのラベルでの最小オイラー路となっている。

図は図のグラフにアルゴリズム **Min_euler** を適用した例である。v_3 を最初の cv として選び、次に e として (v_3, v_5) を選んでいる。すると cv が v_5 に更新される。次に $e = (v_5, v_2)$ としている。cv が v_2 に更新されると、**Min_euler** の 8 によって、cv は v_3 になる。以下、アルゴリズムに従うと、頂点 $(v_1, v_2, v_3, v_4.v_5)$ はそれぞれ番号 $(5, 3, 1, 4, 2)$ をもち、得られるオイラー路を、この番号で表すと $(1, 2, 3, 1, 4, 2, 5, 3, 4)$ となる。

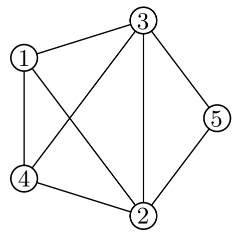

図 3 図 1 のグラフに **Min_euler** を実行したときの例

定理 1 アルゴリズム **Min_euler** は、アルゴリズムで与えた頂点の番号付けによって最小オイラー路を $O(m^2)$ で構成する。

出力される最小オイラー路については次が成り立つ。

補題 2 アルゴリズム **Min_euler** によって出力される最小オイラー路の先頭は $(1, 2, 3)$ である。

4. オイラー路の列挙アルゴリズム

補題 1 からオイラー路は全順序集合になっており、定理 1 からアルゴリズム **Min_euler** によって得られたオイラー路は最小オイラー路である。よって最小オイラー路から順序どおりにオイラー路を構成すればよい。今単純グラフ G に対し、オイラー路 $B = (b_1, b_2, \ldots, b_m)$ を出力し、B の次のオイラー路 C を構成することを考える。$A = (a_1, a_2, \ldots, a_m)$ を B の直前に出力したオイラー路とする。A が存在しない場合は B は最小オイラー路である。このときは $C = \mathrm{fl}(B, \mathrm{lpe}(B))$ である。

B が最小オイラー路でないときを考える。A と B について $a_1 = b = 1, a_2 = b_2, \ldots, a_{i-1} = b_{i-1}$ かつ $a_i \neq b_i$ としよう。このとき、$b_1, \ldots, b_i$ を固定したオイラー路の中で B は最小オイラー路になっている。

$\mathrm{lpe}(B)$ の 2 要素がともに $(b_{i+1}, \ldots, b_m)$ に含まれる場合

$B < \mathrm{fl}(B, \mathrm{lpe}(B))$ である。もし、$B = \mathrm{fl}(B, \mathrm{lpe}(B))$ ならば G が単純グラフであることに反する。$B < D < \mathrm{fl}(B, \mathrm{lpe}(B))$ となる D は存在しないので $C = \mathrm{fl}(B, \mathrm{lpe}(B))$ である。

$\mathrm{lpe}(B)$ の 1 要素が b_i であり、別の 1 要素 b_j が $(b_{i+1}, \ldots, b_m)$ に含まれる場合

このとき、b_i と同じ要素は $(b_{i+1}, \ldots, b_m)$ の中に b_j しか存在しない。もし b_j 以外に b_i と同

じ要素が存在するならば、$\mathrm{lpe}(B)$ の 2 要素が $(b_{i+1},\ldots,b_m)$ に含まれることになるからである。このとき、$B < \mathrm{fl}(B,\mathrm{lpe}(B))$ である。もし、$\mathrm{fl}(B,\mathrm{lpe}(B)) < B$ ならば $b_{i+1} > b_{j-1}$ ということになり、$b_1,\ldots,b_i$ を固定したオイラー路の中で B が最小オイラー路であったことに反する。また、$B < D < \mathrm{fl}(B,\mathrm{lpe}(B))$ となる D は存在しないので $C = \mathrm{fl}(B,\mathrm{lpe}(B))$ である。

$(b_i,\ldots,b_m)$ に $\mathrm{lpe}(B)$ が存在しない場合

$\mathrm{lpe}(B) = (b_h, b_j)$ は $h < i$ である。このとき、$b_{h+1} < b_{j-1}$ ならば $B < \mathrm{fl}(B,\mathrm{lpe}(B))$ であり、$B < D < \mathrm{fl}(B,\mathrm{lpe}(B))$ となる D は存在しないので $C = \mathrm{fl}(B,\mathrm{lpe}(B))$ である。$b_{h+1} > b_{j-1}$ ならば $\mathrm{fl}(B,\mathrm{lpe}(B)) < B$ である。そこで 2-$\mathrm{lpe}_<(\mathrm{fl}(B,\mathrm{lpe}(B))) = (b_p, b_q)$ をみつける。このとき、$\mathrm{lpe}_<(\mathrm{fl}(B,\mathrm{lpe}(B))) = (b_j, b_h)$ であり、$p < j$ である。このとき B での q と j, h の関係によって場合に分けて考える。

$q < h$ のとき $C = \mathrm{fl}(\mathrm{fl}(B,\mathrm{lpe}(B)), 2\text{-}\mathrm{lpe}_<(\mathrm{fl}(B,\mathrm{lpe}(B))))$ である。

$q = h$ のとき $b_{h-1}, b_{h+1}, b_{j-1}$ の中で b_{p+1} より大きくかつもっとも b_{p+1} に近いもの b' とする。$b' = b_{h-1}$ の場合は $C = \mathrm{fl}(\mathrm{fl}(B,\mathrm{lpe}(B)), (b_p, b_j)) = (b_1,\ldots,b_{p-1}, b_j, b_{h-1},\ldots, b_{p+1}, b_p, b_{j-1},\ldots, b_{h+1}, b_h, b_{j+1},\ldots, b_m)$ である。

$b' = b_{h+1}$ の場合は $b_{h-1} < b_{p+1}$ と $b_{h-1} > b_{p+1}$ の 2 通りについて考える。

$b_{h-1} < b_{p+1}$ のとき、$C = \mathrm{fl}(\mathrm{fl}(B,\mathrm{lpe}(B)), (b_h, b_p)) = (b_1,\ldots,b_{p-1}, b_h, b_{h+1},\ldots, b_{j-1}, b_j, b_{h-1},\ldots, b_{p+1}, b_p, b_{j+1},\ldots, b_m)$ である。

$b_{h-1} > b_{p+1}$ のとき、$C = \mathrm{fl}(\mathrm{fl}(\mathrm{fl}(B,\mathrm{lpe}(B)), (b_h, b_p)), (b_j, b_p)) = (b_1,\ldots,b_{p-1}, b_h, b_{h+1},\ldots, b_{j-1}, b_p, b_{p+1},\ldots, b_{h-1}, b_j, b_{j+1},\ldots, b_m)$ である。

$b' = b_{j-1}$ の場合も $b_{h-1} < b_{p+1}$ と $b_{h-1} > b_{p+1}$ の 2 通りについて考える。

$b_{h-1} < b_{p+1}$ のとき、$C = \mathrm{fl}(B, (b_p, b_j)) = (b_1,\ldots,b_{p-1}, b_j, b_{j-1},\ldots, b_{h+1}, b_h, b_{h-1},\ldots, b_{p+1}, b_p, b_{j+1},\ldots, b_m)$ である。

$b_{h-1} > b_{p+1}$ のとき、$C = \mathrm{fl}(\mathrm{fl}(B, (b_p, b_j)), (b_h, b_p)) = (b_1,\ldots,b_{p-1}, b_j, b_{j-1},\ldots, b_{h+1}, b_p, b_{p+1},\ldots, b_{h-1}, b_h, b_{j+1},\ldots, b_m)$ である。

$h < q < j$ のとき b_{q-1}, b_{q+1} の中で b_{p+1} より大きくかつもっとも b_{p+1} に近いもの b' とする。

$b' = b_{q-1}$ の場合は $b_{h-1} < b_{j-1}$ と $b_{h-1} > b_{j-1}$ の 2 通りについて考える。

$b_{h-1} < b_{j-1}$ のとき、$C = \mathrm{fl}(B, (b_p, b_q)) = (b_1,\ldots,b_{p-1}, b_q, b_{q-1},\ldots, b_{h+1}, b_h, b_{h-1},\ldots, b_{p+1}, b_p, b_{q+1},\ldots, b_m)$ である。

$b_{h-1} > b_{j-1}$ のとき、$C = \mathrm{fl}(\mathrm{fl}(B, (b_p, b_q)), (b_h, b_j)) = (b_1,\ldots,b_{p-1}, b_q, b_{q-1},\ldots, b_{h+1}, b_j, b_{j-1},\ldots, b_{q+1}, b_p, b_{p+1},\ldots, b_{h-1}, b_h, b_{j+1},\ldots, b_m)$ である。

$b' = b_{q+1}$ の場合は $b_{h-1} < b_{h+1}$ と $b_{h-1} > b_{h+1}$ の 2 通りについて考える。

$b_{h-1} < b_{h+1}$ のとき、$C = \mathrm{fl}(\mathrm{fl}(B,\mathrm{lpe}(B)), (b_p, b_q)) = (b_1,\ldots,b_{p-1}, b_q, b_{q+1},\ldots, b_{j-1}, b_j, b_{h-1},\ldots, b_{p+1}, b_p, b_{q-1},\ldots, b_{h+1}, b_h, b_{j+1},\ldots, b_m)$ である。

$b_{h-1} > b_{h+1}$ のとき、$C = \mathrm{fl}(\mathrm{fl}(\mathrm{fl}(B,\mathrm{lpe}(B)), (b_p, b_q)), \mathrm{lpe}(b)) = (b_1,\ldots,b_{p-1}, b_q, b_{q+1},\ldots, b_{j+1}, b_h, b_{h+1},\ldots, b_{q-1}, b_p, b_{p+1},\ldots, b_{h-1}, b_j, b_{j+1},\ldots, b_m)$ である。

$j < q$ のとき $C = \mathrm{fl}(B, (b_p, b_q))$ とする。

$\mathrm{lpe}(B)$ が存在しないときは G はパスである。

グラフ G がオイラー路をもち、奇頂点をもつ場合は補題 2 より、オイラー路の先頭を 1 に固定すれば、オイラー路の代表表記のみを列挙することになる。なぜならば、この場合の同一のオイラー路は頂点列の反転しかなく、オイラー路の終点は 1 より大きい整数が割り当てられているからである。また、グラフ G がオイラー回路をもつ場合は補題 2 より、オイラー路の先頭を $1, 2$ に固定すれば代表表記のみを列挙することになる。

上記の議論から B の次のオイラー路を得るために高々3 回のフリップを行えばよい。また、フリップする要素を見つけるためにオイラー路を走査する必要があるが、これも定数回でよい。よって、次が成り立つ。

定理 2 与えられた単純グラフに対して、$O(m^2)$ の前処理の後、そのオイラー路を 1 つあたり $O(m)$ で列挙できる。

5. まとめ

本研究ではオイラー路の列挙について考察した。提案するアルゴリズムは $O(m^2)$ の前処理の後にオイラー路 1 つあたり $O(m)$ で列挙する。前処理では Fleury's Algorithm を用いて最小オイラー路を構成している。また、オイラー路の列挙にあたってはオイラー路の集合に全順序を定義して、その順序を利用している。

参考文献

1) G. Brightwell and P. Winkler, *Counting eulerian circuits is #P-complete*, Proc. ALENEX /ANALCO 2005, 259–262, 2005.
2) B.D.McKay and R.W.Robinson, *Asymptotic enumeration of Eulerian circuits in the complete graph*, Combinatorics, Probability and Computing, 7, 437–449,1998.