6K-04

グラフに含まれる大きな内周を持つ部分グラフの効率良い列挙

栗田 和宏[1,a)]　Alessio Conte[3]　和佐 州洋[2]　宇野 毅明[2]　有村 博紀[1]

受付日 2015年3月4日, 採録日 2015年8月1日

概要：内周はグラフに含まれる最小の閉路長である．本論文では，内周が k 以上の部分グラフを解 1 つあたり $O(n)$ 時間より効率よく列挙するアルゴリズムを与える．このアルゴリズムの単純な拡張により，k 以上の内周を持つ重み付きグラフの部分グラフや非連結な部分グラフの列挙も可能である．

キーワード：部分グラフ列挙，ならし多項式時間，内周，分割法

Efficient Enumeration of Subgraphs with Large Girth

Kazuhiro Kurita[1,a)]　Alessio Conte[3]　Kunihiro Wasa[2]　Takeaki Uno[2]　Hiroki Arimura[1]

Received: March 4, 2015, Accepted: August 1, 2015

Abstract: The girth is the length of a shortest cycle in a graph. In this paper, we propose an algorithm for the enumeration of subgraphs with girth at least k. The algorithm runs in $O(n)$ time per solution, and can be easily adapted to the weighted case and the non-connected case.

Keywords: Subgraph enumeration, Amortized polynomial time, Girth, Binary partition

1. はじめに

本論文では内周が k 以上の部分グラフを列挙する部分グラフ列挙問題を扱う．ここで，内周とはグラフに含まれる最小閉路長である．**部分グラフ列挙問題**とは与えられたグラフ G に対して，ある制約 $\mathcal{R}$ が与えられたとき，$\mathcal{R}$ を満たす G の部分グラフをすべて出力する問題である．

一般に，アルゴリズムの時間や空間の効率は入力サイズ N に対して測られるが，本論文では列挙アルゴリズムの効率を N と出力サイズ M の両方について測る．ここで，M とは入力グラフ G に対する解の総数である．列挙アルゴリズムが $O(M \cdot poly(N))$ 時間で抑えられる場合，そのアルゴリズムを**ならし多項式時間アルゴリズム**と呼ばれる．

主結果として，我々はグラフ中の内周 k 以上の部分グラフを列挙するアルゴリズム ELG-S を提案する．ELG-S は上記の問題を $O\left(\sum_{S\in\mathcal{S}}|V(G[S])|\right)$ 時間と $O\left(\max_{S\in\mathcal{S}}\left\{|V(G[S])|^3\right\}\right)$ 空間で解く．ここで，$\mathcal{S}$ はすべての解からなる集合とする．$|V(G[S])|$ がたかだか G の頂点数 n であることから，この計算量はならし $O(n)$ 時間，$O\left(n^3\right)$ 空間よりも効率的な結果である．最後に本論文で扱う問題を定義と図 1 に入力グラフ G と解の例を示す．

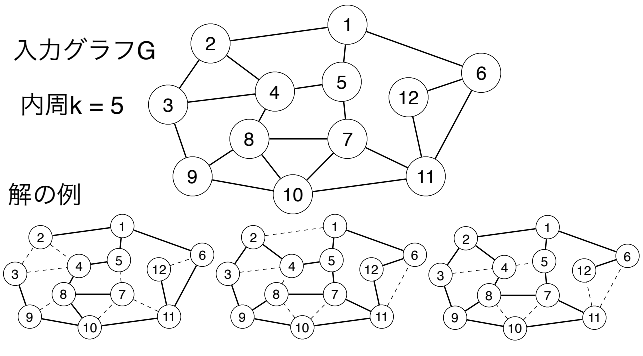

図 1　入力グラフ G と $k = 5$ の場合の解の例

問題 1 (k-内周部分グラフ列挙問題)．与えられたグラフ G に含まれる内周 k 以上の部分グラフをもれなく重複なく出力せよ．

2. 提案アルゴリズムの概要

ELG-S の概要を Algoritrhm 1 に示す．ELG-S は分割法を用いて構築されている．ここで，ELG-S 中の 1 つの再帰手続き RecELG を**反復**という．分割法とは解集

[1] 北海道大学 大学院情報科学研究科 情報理工学専攻, 北海道
[2] 国立情報学研究所, 東京
[3] Universit di Pisa, Pisa, Italy
[a)] k-kurita@ist.hokudai.ac.jp

Algorithm 1: 内周 k 以上の部分グラフを列挙するアルゴリズム ELG-S .

1 **Procedure** *ELG-S* (G, k) // G: 入力グラフ, k: 正整数
2 　RecELG$(\emptyset, G)$;
3 **Procedure** *RecELG* (S, G) // S: 現在の解
4 　Output S;
5 　DONE $\leftarrow \emptyset$;
6 　**for** $e \in C(S)$ **do**
7 　　RecELG$(S \cup \{e\}, G \setminus$ DONE$)$;
8 　　DONE $\leftarrow$ DONE $\cup \{e\}$;
9 　**return**;

合 $\mathcal{S}$ を 2 つの互いに素な集合 $\mathcal{S}_0$ と $\mathcal{S}_1$ に再帰的に分割し，$|\mathcal{S}_0|$ や $|\mathcal{S}_1|$ が 1 になるまで繰り返し分割することで解を列挙する手法である．ここで，$\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_0$ を満たす．ELG-S では，$\mathcal{S}$ をある辺 e_1 を持つ解集合 $\mathcal{S}_1$，e_2 を持つ解集合... e_ℓ を持つ解集合 $\mathcal{S}_\ell$ に分割を行う．このとき，辺集合 $C(S)$ を次のように定義し，$C(S)$ 中のそれぞれの辺を持つかどうかで解集合の分割を行う．$C(S) = \{e \in E(G) \setminus S \mid G[S \cup \{e\}]$ の内周が k 以上 $\}$．ここで，$C(S)$ を**候補集合**という次に ELG-S が解を列挙することを示す．以下の補題で，解のサイズとは部分グラフの辺数を表す．

補題 1. *ELG-S* の出力は全て解である．

証明． ELG-S の出力のサイズに対する帰納法を用いて証明する．サイズ k 以下の出力がすべて解と仮定する．ELG-S の構成より，出力 $G[S]$ に加えられる辺は $C(S)$ 中の辺である．$C(S)$ の定義より，任意の $e \in C(S)$ に対して，$G[S \cup \{e\}]$ の内周は k 以上である． □

補題 2. G 中の解は全て出力される．

証明． 解のサイズに対する帰納法を用いて証明する．サイズ k 以下の解が列挙されていると仮定し，サイズ $k+1$ の解 S を考える．このとき，(1)$S_X \subset S$ かつ DONE $\cap S = \emptyset$ を満たす反復 X が存在する．X が ELG-S に呼び出せれた反復の場合に明らかに条件 (1) を満たすので，このような X は少なくとも 1 つは存在する．内周の定義と DONE$\cap S = \emptyset$ より，$S \setminus S_X \subseteq C(S_X)$ を満たす．ここで，$e \in S \setminus S_X$ を S_X に加えると，条件 (1) を満たす $|S_X|$ より大きな解を持つ反復 Y が存在する．これを繰り返すことにより，S を出力する反復ができる． □

補題 1 と補題 2 より次の定理が成り立つ．

定理 1. *ELG-S* は G 中の解を列挙する．

3. 提案アルゴリズムの解析

ELG-S では $C(S')$ の計算時間が支配的である．ここで $S' = S \cup \{e_i\} (1 \leq i \leq \ell)$ とする．$C(S')$ を計算するために，すべての $f \in C(S) \setminus \{e\}$ に対して $G[S' \cup \{f\}]$ の内周を計算することで，$C(S')$ が計算できる．これは Itai らのアルゴリズム [2] を用いて $O(nm^2)$ で計算可能である．ELG-S は任意の 2 点間の距離を持つ距離行列 M を用いて $C(S')$ を効率良く計算する．M を持つことで，$G[S']$ の内周が既知であるならば，$G[S' \cup \{f\}]$ の内周は定数時間で計算できる．したがって，M を用いると $C(S')$ は $O(m)$ 時間で計算できる．さらに，M は Floyed-Warshall アルゴリズム [1] と似た動的計画法を用いて，$O(|V(G[S'])|)$ 時間で更新できる．したがって，ELG-S の 7 行目の再帰呼び出しは $O(m + |V(G[S'])|)$ 時間で計算可能である．

Algorithm 2: ELG-S における距離行列を用いた候補集合の計算.

Procedure *NextC*$(C(S), M(S), S, k, G)$
　if $C_{in}(S) \neq \emptyset$ **then** $e \leftarrow C_{in}(S)$; **else** $e \leftarrow C_{out}(S)$;
　$S' \leftarrow S \cup \{e\}$;
　$C(S') \leftarrow$ UpdateCand(e, S);
　$M(S') \leftarrow$ UpdateDistanceMatrix$(e, C(S'))$;
Function *UpdateCand*$(e = \{u, v\}, S)$
　if $e \in C_{in}(S)$ **then**
　　for $f \in C_{in}(S) \setminus \{e\}$ **do**
　　　if $G[S \cup \{e, f\}]$ の内周が k 以上 **then**
　　　　$C_{in}(S) \leftarrow C_{in}(S) \cup \{f\}$;
　else // $u \in G[S]$ と $v \notin G[S]$ を仮定する.
　　for $w \in N(v)$ **do** // $f = \{v, w\}$ とする.
　　　if $G[S \cup \{e, f\}])$ の内周が k 未満 **then**
　　　　$C_{out}(S) \leftarrow C_{out}(S) \setminus f$
　　　else if $w \in G[S]$ **then**
　　　　$(C_{in}(S), C_{out}(S)) \leftarrow$
　　　　$(C_{in}(S) \cup f, C_{out}(S) \setminus f)$;
　　　else
　　　　$C_{out}(S) \leftarrow C_{out}(S) \cup f$
　return $C_{in}(S) \cup C_{out}(S)$;

より効率的に $C(S')$ を計算するために ELG-S の 6 行目における辺の選択順を考える．ここで，C_{in} と C_{out} を次のように定義する．$C_{in}(S) = \{\{u, v\} \in C(S) \mid u, v \in V(G[S])\}$，$C_{out}(S) = C(S) \setminus C_{in}(S)$．6 行目において，$C_{in} \neq \emptyset$ ならば $e \in C_{in}$ を選択する．このとき，次の補題が成り立つ．

補題 3. $e \in C_{in}(S)$ とする．このとき，$C_{out}(S) = C_{out}(S \cup \{e\})$ である．

したがって，$e \in C_{in}(S)$ ならば，$C(S)$ の更新の際，$C_{in}(S)$ のみ考慮すれば良い．さらに，この選択順で辺を選択することにより，$e \in C_{out}(S)$ ならば，$C_{in} = \emptyset$ になる．1 つの $C_{out}(S)$ 中の辺 e を加えることで $C_{in}(S)$ 中の辺は $O(\Delta)$ 本しか増えないため，次の補題が成り立つ．

補題 4. 任意の反復 X と X が持つ解 S_X とする．このとき，$|C_{in}(S_X)| = O(\Delta)$ である．

したがって，7 行目の再帰呼び出しの計算時間は $O(\Delta + |V(G[S])|)$ になり，以下の定理が成り立つ．

定理 2. *ELG-S* はグラフ G 中の内周 k 以上の部分グラフを $O\left(\sum_{S \in \mathcal{S}} |V(G[S])|\right)$ 時間と $O\left(\max_{S \in \mathcal{S}} \left\{|V(G[S])|^3\right\}\right)$ 空間で列挙する．

参考文献

[1] T. H. Cormen. *Introduction to algorithms.* MIT press, 2009.

[2] A. Itai and M. Rodeh. Finding a minimum circuit in a graph. *SIAM J. Comput.*, 7(4):413–423, 1978.