4G－7

拡張 2-b-SPG グラフにおける Legal sequence number の計算について *

吉岡尚美　葛 崎偉 †
山口大学 教育学部 ‡

1．はじめに

有向閉路のないグラフ DAG(directed acyclic graph) $G=(V,E)$ におけるトポロジカルソーティング問題は、枝 $(u,v)\in E$ のときノード u はノード v より前に順序づけられるというように、すべてのノードの順序づけを求める問題である。順序づけられたノード系列を *legal sequence* と呼び、その総数 LSN(legal sequence number) を求める問題が提起されている [1]。

本論文では、DAG のサブクラスである拡張 2-b-SPG の LSN を求めることを目的としている。拡張 2-b-SPG の LSN 計算が可能であることは知られているが [1]、その具体的な手順は定められていない。そこで、拡張 2-b-SPG の柱と呼ばれる２本のパスを探索し、更にその LSN を求めることを考察する。

2．準備

DAG $G=(V,E)$ において、G が唯一のソースノード s とシンクノード t を持つとき、st-DAG(two terminal DAG) という。SPG (series-parallel graph) は DAG のサブクラスで、b-SPG (basic series-parallel graph) と呼ばれる s と t 以外のノードが入出力枝をそれぞれ１本ずつ持つグラフを直列や並列に組み合わせたグラフである。s から t までのパスが λ 本ある b-SPG を λ-b-SPG といい、そのパスのことを柱と呼ぶ [図 1]。また、[図 1] のように 2-b-SPG の２本の柱の間に、複数の枝を加えたグラフを拡張 2-b-SPG と呼び、その枝を *bridge* と呼ぶ。

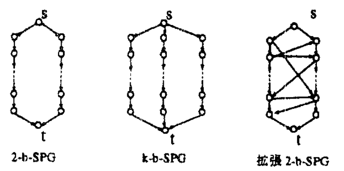

図１．b-SPG と 拡張2-b-SPGの例

現在、SPG と拡張 2-b-SPG に関してのみ LSN の計算が可能であることが分かっており、以下の結果が得られている [1]。

DAG $G=(V,E)$ において、ある枝 $(u,v)\in E$ に対してノード u から v に枝 (u,v) を含まない有向パスが存在するとき枝 (u,v) を余分枝といい、余分枝を含まないグラフを極小グラフという。

[結果１] (1) 有向グラフ G が有向閉路を持つならば、$LSN(G)=0$ が成り立つ。(2)G_m が DAG G の極小グラフならば、$LSN(G)=LSN(G_m)$ が成り立つ。

[結果２] st-DAG $G=(V,E)$ の任意の２つのノード μ_1,μ_2 に対して枝を付加した２つのグラフ $G_1=(V,E\cup\{(\mu_1,\mu_2)\})$、$G_2=(V,E\cup\{(\mu_2,\mu_1)\})$ に関して、$LSN(G)=LSN(G_1)+LSN(G_2)$ が成り立つ。

[結果３] $G_1=(V_1,E_1)$,$G_2=(V_2,E_2)$ を２つの b-SPG とする。G_s と G_p がそれぞれ G_1,G_2 を直列と並列に構成した SPG ならば、$LSN(G_s)=LSN(G_1)\times LSN(G_2)$ と $LSN(G_p)=(LSN(G_1)\times LSN(G_2))\times C(|V_1|-2,|V_2|-2)$ が成り立つ。

[結果４] [結果１]〜[結果３] より、k 本の *bridge* が並行で交互の向きに加わった 2-b-SPG[図 2] に対して、次の計算式が成り立つ。

$$
\begin{aligned}
LSN(G)=&\sum_{i_1=0}^{p_1} C(p_0+1+i_1,q_0)\\
&\sum_{i_2=0}^{q_2} C(q_1+1+i_2,p_1-i_1)\\
&\sum_{i_3=0}^{p_3} C(p_2+1+i_3,q_2-i_2)\\
&\vdots\\
&\sum_{i_k=0}^{x_k}(C(x_{k-1}+1+i_k,y_{k-1}-i_{k-1})\\
&\times C(x_k-i_k,y_k))
\end{aligned}
$$

このとき (u_i,v_i) は *bridge*、p_i,q_i はそれぞれ u_i,v_i,s,t を除いた柱上のノード数である [図 2]。また、*bridge* の数が奇数のとき $x_j=p_j,y_j=q_j$ となり、偶数のとき $x_j=q_j,y_j=p_j$ となる。

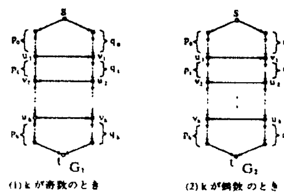

図２．k 本のbridge を持つ拡張2-b-SPG

3．拡張 2-b-SPG の LSN を求める手順

拡張 2-b-SPG の LSN 計算が可能であるが、*bridge* が並行でかつ交互の向きのグラフ以外の場合は、式が適応できる形にグラフを変換しなければならない。そこで本章では任意の拡張 2-b-SPG が与えられたと仮定して、以下の手順に従ってグラフを処理することにより LSN 計算が可能であることについて考察していく。

《LSN 計算の手順》

*On Computation of Legal Sequence Number for Extended 2-b-SPG Graphs
†Naomi Yoshioka and Qi-Wei Ge
‡Faculty of Education, Yamaguchi University, Japan

step1. 有向閉路の有無の判定
step2. 余分枝の削除
step3. 2-b-SPG の判別
step4. 2-b-SPG の柱の探索
step5. 交差枝の探索と削除
step6. LSNの計算

まず **step1** で、与えられた拡張 2-b-SPG G=(V,E) が DAG であるかどうかを判定する。もし有向閉路が存在するならば、[結果 1] より $LSN(G)$=0 であり LSN 計算は終了する。有向閉路探索の手法は $Tarjan$ のアルゴリズムを利用する [2]。次に **step2** で、余分枝を削除し極小 DAG にする。余分枝削除は [2] の《階層化アルゴリズム》で階層構造型有向グラフを作成し、《余分情報削除アルゴリズム》を適用すればよい。余分枝を削除したグラフについて次の性質がいえる。

[性質 1] 拡張 2-b-SPG から余分枝を削除したグラフを G とする。(1)G のすべてのノードに関して、入力枝および出力枝は高々 2 本である。(2)G は複数の拡張 2-b-SPG が直列につながったグラフである。

[性質 1] より、**step3** で拡張 2-b-SPG 成分を探索する。これは余分枝が削除された階層構造型有向グラフにおいて、ノードが 1 つしか含まれない階層に着目すれば容易に探索できる。各々の拡張 2-b-SPG 成分に対して **step4,5** の処理を行ない、計算式が適用可能となったグラフに対して、**step6** で LSNの計算を行い、《LSN計算の手順》が終了する。

従って残り **step4,5** を解決が必要であり、本論文では次章で **step4** の柱を探索する手順を述べ、**step5** に関しては簡単に手法のみを以下に述べる。

step5 では、拡張 2-b-SPG に $bridge$ がどのように加わっているかを調べ、交差している $bridge$ を探索する。次に $bridge$ が並行で交互の向きになるように、[結果 2] を利用して枝の付加や余分枝削除を繰り返し、交差を取り除く。**step5** では多数のグラフが生成されるため、手法の改善が必要である。

4. 柱を探索するアルゴリズム

この章では、拡張 2-b-SPG の柱の探索の手順を詳しく述べる。柱のノードを探索するために、ソースノードから逐次的にノードを削除する方法をとり、新たにソースとなったノードがどちらの柱に属するべきかの判定を行ない、判定できたノードを削除する。最後にグラフが空となり、アルゴリズムが終了する。

《柱の探索法》

step0.

2 本の柱をそれぞれ $X=(x_1,x_2,\cdots,x_n)$, $Y=(y_1,y_2,\cdots,y_m)$ と表記し、$i \leftarrow 1$, $j \leftarrow 1$, $G' \leftarrow G$, $flag\_x \leftarrow 0$, $flag\_y \leftarrow 0$ とする。

step1.

s と隣接する 2 つのノードをそれぞれ x,y とし、$x_i \leftarrow x$, $y_j \leftarrow y$, $G' \leftarrow G'$-$\{s\}$, $i \leftarrow i+1, j \leftarrow j+1$ とする。

step2. (柱 X の探索)

(1)$x=t$ のとき

(i)$y=t$ ならば、$G' \leftarrow G'$-$\{t\}$ とし、終了。

(ii)$y \neq t$ ならば、step3 へ。

(2)x と隣接するノードが 1 つのとき (u とする)

$x_i \leftarrow u$, $x \leftarrow u$, $G' \leftarrow G'$-$\{x\}$, $i \leftarrow i+1$ とし、step2 を繰り返す。

(3)x と隣接するノードが 2 つのとき (u_1,u_2 とする)

(i) 一方のノードの入力枝が 1 本のとき

$x_i \leftarrow u_{1or2}$, $x \leftarrow u_{1or2}$, $G' \leftarrow G'$-$\{x\}$, $i \leftarrow i+1$ とし、step2 を繰り返す。

(ii) 両方のノードの入力枝が 2 本のとき

(α)$flag\_y = 0$ ならば、$flag\_x \leftarrow 1$ とし、step3 へ。

(β)$flag\_y = 1$ ならば、$x_i \leftarrow u_1$, $y_j \leftarrow u_2$, $x \leftarrow u_1$, $y \leftarrow u_2$, $G' \leftarrow G'$-$\{x,y\}$, $i \leftarrow i+1$, $j \leftarrow j+1$, $flag\_x \leftarrow 0$, $flag\_y \leftarrow 0$ とし、step2 を繰り返す。

step3. (柱 Y の探索)

step2.(1)(2)(3) において、x と y、u と v、x_i と y_j、$u_{1,2}$ と $v_{1,2}$、i と j、$flag\_x$ と $flag\_y$、step2 と step3 を入れ換えて実行する。

上記のアルゴリズムは、$O(|V|+|E|)$ で実行可能である。

5. まとめ

本論文では、与えられた拡張 2-b-SPG に対する LSN を求めることを目的とし、実際にグラフが与えられた際のグラフに対する処理手順を考察した。また、拡張 2-b-SPG の柱を探索するアルゴリズムを与えた。今後の課題として、**step5**:交差枝の探索と削除をアルゴリズム化し、更に LSN問題が NP-$hard$ 問題または#P 問題であるかについての究明をし、一般の DAG に対する LSN計算手法の拡張または近似法の開発を行なうことがあげられる。

参考文献

[1] Q.W. Ge, N. Yoshioka: "Method of Finding Legal Sequence Number for a Class of Extended Series-Parallel Digraphs," IEICE Trans., vol.E80-A, no.4, pp.635-642, Apr.1997.

[2] Q.W. Ge, R. Wu, and H. Yanagida: "On Module Arrangement in Generating Asemble Programs via Directed Graph," Bulletin of Faculty of Education, Yamaguchi Univ., vol.45, pt.2, pp.53-64, Oct. 1995.