構造解析のためのデータ・プレプロセシングに関する一考察

岡山大学工学部　正員　○谷口健男
川田工業　岡　清志
巴組鉄工所　沼田　克

1. まえがき

今日のマトリックス構造解析においては扱われる系の巨大化、および より精度の高い解の必要性により、多量の入力データを扱わねばならず、従って、そのための有効なデータ処理法の開発が望まれるに到っており、その一つが 多元連立一次方程式の各種数値計算法のデータ前処理である Renumbering および Reordering 問題である。本研究においては、それらのうち、特に入力データ量を最小にしうるスパースマトリックス法のための Reordering 問題に対して考察を加える。この数値計算法のための入力データ量は対象とする行列内の非零要素数と、消去演算過程において非零化する要素（フィル・インと呼ばれ、Fi で示す）数で決定されることより、このための入力データ量最小化問題は、フィル・イン最小化問題に置き換えられる。この問題に対する研究は D. J. Rose[1]、A. George[2]、谷口・白石[3]らによって行なわれ、すでにいくつかの手法も提案されているが、真の最小値を与え、かつ汎用性のある方法はいまだ見い出されていないのが現状である。しかしながら、文献3)によれば 系の示すグラフを分割し、それにより得られた部分グラフに対し最小化を行えば 最小フィル・インを達成しうることが証明され、また、Nested Dissection 法[2] は特殊なグラフに対しては非常に良い結果を与えることができる。本研究においては、これら両者の研究を基にして、さらに一般的なグラフに対して適用可能なフィル・イン最小化法のための基礎的研究を行う。

2. 最小フィル・イン問題

対称かつ疎な行列に対する消去過程は、その行列の示すグラフを対象とすれば vertex elimination[1] として認識され、従って Minimum Fill-in Problem は elimination process において新たに発生する枝（線）の数の最小化問題と考えることができる。文献3）によれば 任意の行列の示す1つのグラフ：G を対象に、Min. Fi を目的として消去を行えば G に含まれる全ての点は、次に示すいずれかのタイプに属することがすでに証明されている。

Type 1 ： $x \notin$ FVG
Type 2 ： $x \in$ FVG, adj. $x \cap$ FVG $\neq \phi$
Type 3 ： $x \in$ FVG, adj. $x \cap$ FVG $= \phi$
Type 4 ： $x \in$ FVG's, adj. $x \cap$ FVG's $= \phi$

ここで x は任意の点、FVG は G においてすでにいくつかの点が消去されている時、消去点に隣接する完全部分グラフを示し、adj. x は点 x に隣接する部分グラフを示す。これら4つの Type の消去は、それぞれ FVG の発生（Type 1）、発達（Type 2）、合併（Type 4）、縮小（Type 3）という物理的意味付けがなされ、さらに、elimination process はこの順序で行なわれることがこれら4Typesの前後関係を探ることにより論理的に得られる。さらに重要なことは、Type 4の2つのFVG'sの合併の直前において それらFVG'sの発達が停止することであり、これを停止FVG（$\overline{\text{FVG}}$ で示す）と呼ぶと、G を $\overline{\text{FVG}}$'s の個所において分割し、その結果得られる部分グラフに対し独立に Min. Fi を行いうるということである。これより、Min. Fill-in問題において最も重要なことは $\overline{\text{FVG}}$ を探ることであると結論づけられる。

文献3）において optimal elimination process は Type 1 ⟶ Type 2 ⟶ Type 4 ⟶ Type 3の繰り返しであると考えられてきたが、この elimination process を用いれば グラフの内部点（G が連続体の離散化モデルであれば 境界より隔たった所に位置する点）は必然的に Type 2, 3, 4 のいずれかで消去されることになり、従って、もし G の全体の形が矩形であり regular mesh（内部点の次数が一定）であるとすれば、

そこに発生するFUGは必然的にGの幅により支配されることになる。Fill-inは|FUG|（ここで||は節点数を示す）の2乗に比例することより[3]，もしグラフの幅が大きくなれば上記elimination process modelにおいて多くのfill-inが発生することになる。一方Nested Dissection法によれば，Gの内部点においてもType 1の消去が発生し，また文献1)に示されるMinimum Degree，およびMinimum Deficiency Algorithmsを適用してもこの現象がみられる。文献3)のFill-inの評価式をみれば，Fill-inの値に影響を及ぼす要因として，上述したGの幅以外に，各節点の次数（点のもつ線の数）が挙げられる。従って，以下において，次数のFUGに対する影響を考察してみる。

Regular mesh graphを対象とした時，もしGの幅がある値以下の場合，そして各点の次数がある値以上であれば，文献3)のelimination process modelで十分であろう（これは，節点数の少いグラフに対する数値実験結果より得られる）。しかしながら，同じ節点数のグラフであっても各点の次数が小さい場合，例えば5点差分型regular mesh graphにおいては数値実験においてoptimal eliminationはGの諸点をまず1つとばしにType-1 eliminationされることが認められる。更に，このeliminationの後得られるグラフは9点差分型regular mesh graphとなることより，上述したelimination process modelが成り立ち，最小フィル・インを与えることになる。このことは，もし，そのグラフの幅が十分大きければ，例えそのgraphが9点差分型であったとしても，5点差分型のeliminationが発生する可能性を示すものと考えられる。

数多くの数値実験と，文献3)に示されるelimination process modelに対する上記考察を基にして，以下に示すような新たなelimination process modelを提案する。なお，この新たなmodelは上述したType 1～Type 4の消去タイプのうち，Type 1と2のみを用いるものであって，文献3)の修正モデルと考えられるものである。いま与えられたグラフをG(0)で示す。

1. G(0)を|FUG(0)|の境界長さを有する凸な部分グラフの集合に分割する。
2. 分割された部分グラフ内諸点をType 1とType 2により全て消去する。
3. 未消去点はFUG(0)のみであり，これを新たなグラフ：G(1)と考える。G(1)を再び|FUG(1)|の境界長さを有する凸な部分グラフの集合に分割し，以下，2，3，1の各プロセスを繰り返す。

この新たなmodelにおいて消去の対象となるグラフは$G(0) \to G(1) \to G(2) \to \cdots$と順次変化し，各段階における点の次数は順次大きくなり，|FUG(i)|も同様に大きくなる。ここで重要なのは，|FUG(i)|の大きさであるが，現在得られているのは|FUG(0)|のみであり，その値は11である。

3. 数値実験

右表は，N×Nに分割された9点差分型regular mesh graphに対する今回の方法，文献3)の方法，Min.Deg，Min.Def，およびNested Dissection法により得られたFill-inの比較表である。いずれも今回の方法が最小値を与えている。

	節点数	今回の方法	P.M.1	MIN・DEF	MIN・DEG	N・D
5 × 5	25	28	28	28		
6 × 6	36	64	64	64		
7 × 7	49	104	104	110		
8 × 8	64	191	191	207	218	
9 × 9	81	288	288	316	349	300
10 × 10	100	420	420	439	473	
11 × 11	121	573	585	610	677	
12 × 12	144	748	795	799	898	
13 × 13	169	968	1041	1018	1280	
14 × 14	196	1235	1336	1320	1367	
15 × 15	225	1519	1691	1681	1861	
16 × 16	256	1860	2079	2002	2316	
17 × 17	289	2268	2528	2406	2898	2280
18 × 18	324	2716	3056	2820	3080	
19 × 19	361	3173	3727	3302	3857	

4. あとがき

本研究において，まず最小フィル・イン問題に関して考察を加えることにより，新たな消去選択モデルの提案を行い，ついで他のいくつかの手法との比較により，現在の所optimalであるとの結果を得た。本手法は，Nested Dissection，文献3)の方法に似かよっているが，より汎用性があるのみならず，更に良い結果を与えることができる。

参考文献　1). D.J.Rose, Graph Theory & Computing, 1972, 183-217, 2). A.George, SIAM J. Numer. Anal., 10, 2, 1973, 345-363, 3). 谷口・白石，第13回マトリクス解析法論文集，S.54，103～108