数 値 解 析 31－4
（1989．12．8）

大型疎線形計画問題に対するＲｅｉｄ法の効率化について

井村浩也　大柳俊夫　大内　東
北海道大学工学部

大型の線形計画問題の基底行列は一般にスパース（疎）な構造となっており、計算機のメモリーや処理能力の制約のもとで大規模な問題を高速に解くためには、基底の更新の際にその疎な性質をいかに保持するかが問題となる。この問題に対し現在まで多くの研究がなされており、その中でJ.K.Reidによる方法は、計算速度と数値的安定性の両方の面で優れた方法として知られている。Reid法は、基底行列の行と列の置換により基底行列内のバンプと呼ばれる部分行列をできる限り縮小し、Sub-Diagonal要素の消去によるフィル・インの発生を抑制するものである。

本論文では、バンプがバンプ内の最初と最後の列を置換することにより大幅に縮小される場合があることを示し、Reid法にその場合を判定する手順を加えた改良Reid法を提案する。

On Improvement of Reid Method for Large Sparse Linear Programming Problems

Hiroya Imura　Toshio Ohyanagi　Azuma Ohuchi
Faculty of Engineering,
Hokkaido University

North 13 West 8, North ward, Sapporo 060, Japan

A structure of a basis matrix of a large sparse linear programming problem is usually sparse. To solve large problems under space and time constraints on computers, the sparsity of the basis matrix must be maintained when it is updated. Many researches studies this problem and proposed several algorithms. J.K.Reid also proposed the algorithm which improved an efficiency of Bartels-Golub algorithm. In Reid algorithm column or row permutations in a bump which is a submatrix of the basis matrix are used to restrain occurences of fill-in.

This paper intends to improve Reid method. For this purpose a new column permutation strategy in the bump is proposed. Then some computational experiments are made to clarify an effectiveness of the proposed method.

1．はじめに

大型の線形計画問題の基底行列は一般にスパース（疎）な構造となっている。そこで、計算機のメモリーや処理能力の制約のもとでできる限り大規模な問題を高速に解くためには、基底の更新の際にその疎な性質をいかに保持するかが問題となり、この問題を解決するために現在まで多くの研究がなされている。[3]その中でJ.K.Reidによる方法は、計算速度と数値的安定性の両方の面で優れた方法として知られている。その方法は、基底行列を三角分解して保存しておくR.H.BartelsとG.H.Golubによる方法[1]を改良したもので、基底行列の行と列の置換により基底行列内のバンプと呼ばれる部分行列をできる限り縮小し、Sub-Diagonal要素の消去によるフィル・インの発生を抑制するものである。[2]

本論文では、バンプがバンプ内の最初と最後の列を置換することにより大幅に縮小される場合があることを示し、Reidの提案による方法（以下Reid法と呼ぶ）にその場合を判定する手順を加えた改良Reid法を提案する。そして、数値実験により改良Reid法の有効性を示す。

以下、第２章でReid法の概略を述べ、第３章で今回提案する改良Reid法を説明する。第４章では改良Reid法の有効性を示すために行なった数値実験結果について述べる。

2．Reid法

2．1　疎Bartels-Golub法[2]

改訂シンプレックス法の各反復において、２組のｍ元連立一次方程式

$$y\,B = c_b \qquad (1)$$

$$B\,d = a \qquad (2)$$

を必ず解く必要がある。ここで、Ｂは基底行列（ｍ×ｍ）、ｙはシンプレックス乗数ベクトル（ｍ）、c_bは基底変数に対応する費用係数ベクトル（ｍ）であり、ｄは新しく基底に入る列ベクトル（ｍ）、ａはそれに対応する係数ベクトル（ｍ）である。

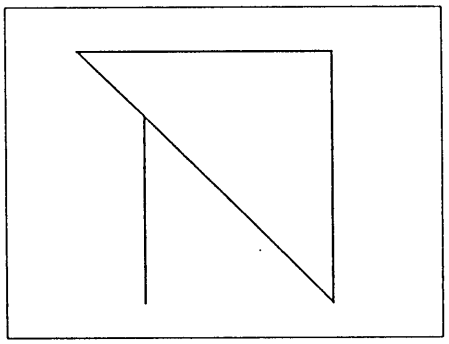

図１　Spiked matrix.

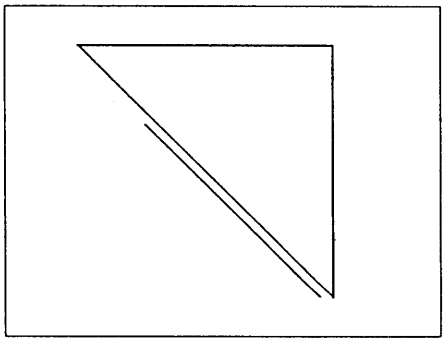

図２　Upper Hessenberg matrix.

疎Bartels-Golub法では、(1)式と(2)式の連立方程式を解くための方法として基底行列の三角分解を用いる。この三角分解は行及び列の置換を含み、次式のように表わすことができる。

$$M_r M_{r-1} \cdots M_1 B = P\,U\,Q \qquad (3)$$

但し、M_i は単位行列と一つの非対角要素のみ異なる行列（ｍ×ｍ）、ＰとＱは置換行列（ｍ×ｍ）、Ｕは上三角行列（ｍ×ｍ）である。この(3)式を用いることにより、(1)式、(2)式は容易に解くことができる。

次の反復において、基底から出る列が新しく基底に入る列ｄといれ代わり、Ｂが$\bar{B}$へと更新されると(3)は次式のようになる。

$$M_r M_{r-1} \cdots M_1 \bar{B} = P\,S\,Q \qquad (4)$$

ここで、行列Ｓ（ｍ×ｍ）はＢと$\bar{B}$が異なる列に対応する列のみＵと異なる。この列をスパイク列と呼

び、その対角要素以下をスパイクと呼ぶ(図１)。

この方法では非零要素の消去を行なうときに、フィル・インの発生をできる限り避けるためにスパイク列を最後の列に移動する。この操作により、Upper-Hessenberg行列が得られる(図２)。この行列のSub-Diagonal要素は、スパイク移動前は対角要素であり、Ｂは正則であると仮定しているので全て非零である。

Sub-Diagonal要素に対して消去を行なうと、分解は更新されて、

$M_r \cdot M_{r-1} \cdots M_r \cdots M_1 \overline{B} = P U Q$　　(5)

となる。

(5)式は(3)式と同じ形式であるので、この反復においても(1)式、(2)式は容易に解くことができる。

このように疎Bartels-Golub法は、各反復の基底行列の更新に伴いその三角分解を更新し、連立一次方程式を容易に解くための道具とする方法である。

２．２　Reid法[2]

大型線形計画問題の基底行列は非常に疎であるので、スパイクが最後の行まで達している可能性は少ない。Reid法はこの性質を利用し、基底の更新時の(5)式においてフィル・インの原因となる消去をできるだけ避けるように、疎Bartels-Golub法の改良を行ったものである。

疎Bartels-Golub法において、Upper-Hessenberg行列上のSub-Diagonal要素はスパイクの長さだけ生じ、そのSub-Diagonal要素を消去することによりフィル・インが生じる可能性がある。したがってスパイクの長さを短くするような行置換Ｐ及び列置換Ｑを選択すれば、結果としてフィル・インの発生を抑制することができる。

基底行列において、スパイクを一辺とする最小の正方部分行列をバンプと呼ぶ(図３の四角内部)。バンプの次数はスパイクの長さで決まるので、スパイクの長さが短くなれば、その分バンプの次数も減少する。

ここで、Reid法の手順及びｍ＝９の場合の簡単な例を以下に示す。例においては非零要素を×で表わし、図3における行番号、列番号を図4～図9においてもラベル付けする。なお、以下の説明において、バンプ内に非零要素が一つしかない行及び列のその非零要素を、列シングルトン及び行シングルトンと呼ぶ。

【手順１】バンプの縮小

バンプを列と行の置換により、以下のように縮小する。

①バンプ内のスパイクを除く列をバンプの左の列から探索し、列シングルトンが存在すればそれをバンプ左上隅に移動するような対称置換を行なう。

列シングルトンをバンプ左上隅に移動することは、列シングルトンが存在する行及び列をバンプの最初の行及び列に移動し、その間の行及び列を下及び右に１ずつシフトすることに対応する。

この操作を列シングルトンがなくなるまで繰り返す。

②バンプ内の最初と最後の行を除く行をバンプの下の行から探索し、行シングルトンが存在すればそれをバンプ右下隅に移動するような対称置換を行なう。

行シングルトンをバンプ右下隅に移動することは、行シングルトンが存在する行及び列をバンプの最後の行及び列に移動し、その間の行及び列を上及び左に１ずつシフトすることに対応する。

この操作を行シングルトンがなくなるまで繰り返す。

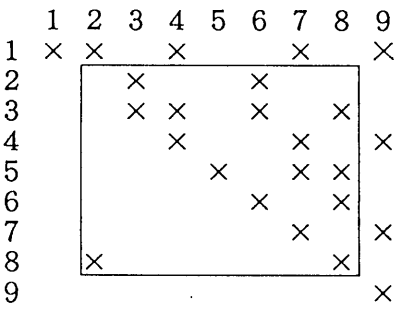

図3 Original matrix.

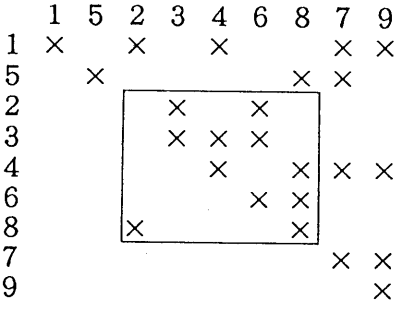

図5 After first row singleton moved to end of bump.

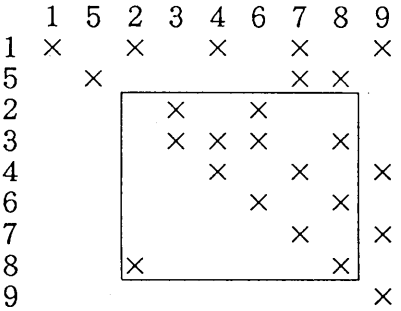

図4 After first and second column singleton moved to front of bump.

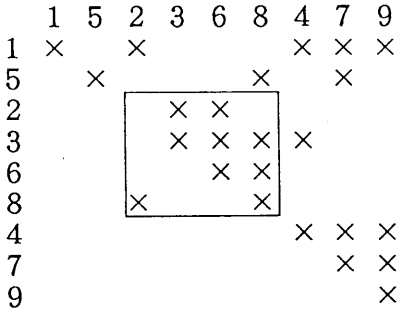

図6 After second row singleton moved to end of bump.

①及び②においてシングルトンを１つ移動することにより、スパイク長は１だけ短くなり、それに伴いバンプの次数は１だけ縮小される。しかし図１で示されるような形式は保持される。

【手順２】Upper-Hessenberg行列の生成

手順1で得られた行列に対して、スパイクをバンプの最後の列に移動し、その間の列を１左にシフトすることによりUpper-Hessenberg行列を生成する。

【手順３】バンプ最後の列シングルトン移動

バンプ内の最後の列に列シングルトンが存在すれば、それをバンプ左上隅に移動する。

列シングルトンが存在しなくなくなるまでこの操作を繰り返す。

【手順４】Sub-Diagonal要素の消去

最後に残ったSub-Diagonal要素の消去を行なう。

図3の例を見ると、列５に列シングルトンが存在していることがわかる。手順１の①を適用すると図4となる。図4には、これ以上列シングルトンは存在しないので手順１の②へ移る。

図4において、行７に行シングルトンが存在するので手順１の②を適用すると図5となる。図4において行４には行シングルトンは存在しなかったが、行７を移動することによりバンプが１縮小され、図5では行４に行シングルトンが存在することになる。同様に手順１の②を適用し、図6を得る。これ以上行シングルトンは存在しないので次の手順に進む。

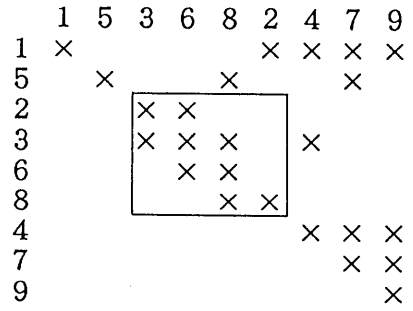

図7 After column permutation
that makes bump upper Hessenberg.

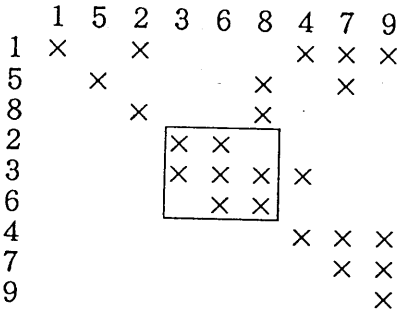

図8 After first column singleton of upper Hessenberg bump moved to front of bump.

手順２に従って図6のスパイク列をバンプの最後の列に移動する。この操作により図7のUpper-Hessenberg行列が生成される。

図7においてバンプの最後の列に列シングルトンが存在するので、手順３に従って列シングルトンを移動する（図8）。バンプ内にこれ以上列シングルトンが存在しないので手順４に従いSub-Diagonal要素を消去すると、三角分解が完了する（図9）。

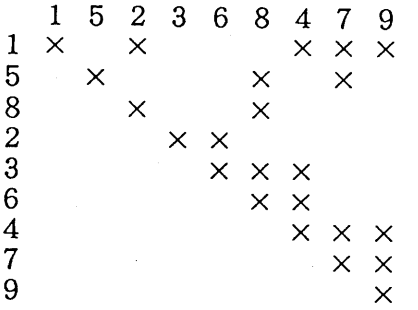

図9 After two sub-diagonal elements eliminated.

３．Reid法の改良

前節で説明したReid法について、バンプの最後の列に列シングルトンが存在する場合及びスパイクに列シングルトンが存在する場合を考え、以下のような改良を加える。以下この方法を改良Reid法を呼ぶ。

Ⅰ．バンプの最後の列に列シングルトンが存在する場合

Reid法に従うと、列シングルトンはバンプの左から探索するので、バンプの最後の列は最後の探索されることになる。しかしバンプの最後の列に列シングルトンが存在する場合、このシングルトンを第一番目に移動することにより、バンプの大幅な縮小が可能となる。また最良の場合このシングルトンの移動により三角分解が完了する場合もある。

いまスパイクがｓ列にありｔ行まで達しているとする。またスパイク上の最後の非零要素の一つ前の非零要素がｋ行（ｓ≦ｋ＜ｔ）

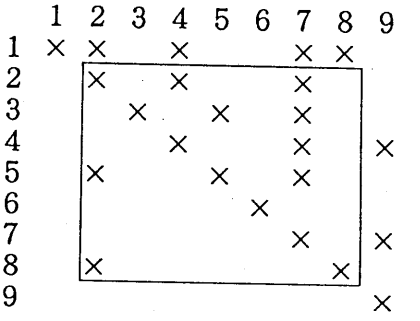

図10 Original matrix.

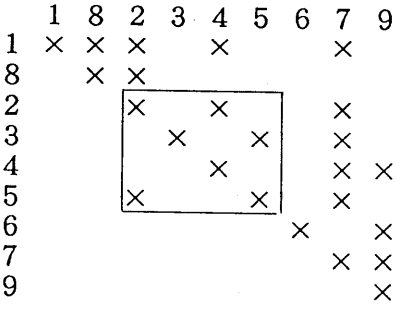

図11 After last colun singleton
moved to front of bump.

にあるとする（図10）。バンプの最後の列シングルトンを移動することによりｔ－ｋだけバンプは縮小される。特にｋ＝ｓの場合、すなわちｋ行がバンプの最初の行である場合には、一度の列シングルトン移動により三角分解は完了する。よって最大ｔ－ｓ回の置換を省略できることになる。このようにバンプの大幅な縮小により、その後のシングルトン探索数がReid法より少なくなり、また全体的な置換数も減少する。

そこで、Reid法の手順１の①における列シングルトンの探索を、まず第一にバンプの最後の列を調べ、その後でスパイクを除いたバンプについて左から順に探索するように改良する。

例えば図10において、最初にバンプの最後の列の列シングルトンを移動すると、図11が示すようにバンプの次数は３だけ縮小される。行６列６と行７列７はバンプ外となり、その後のシングルトン探索の対象外となる。また、図12ではバンプの最後の列の列シングルトンを移動することにより、図13のように三角分解が完了し、他のシングルトンの探索が不必要となる。

	1	2	3	4	5	6	7	8	9
1	×	×		×			×	×	
2		×		×			×		
3			×		×		×		
4				×			×		×
5					×		×		
6						×			
7							×		×
8		×						×	
9									×

図12 Original matrix.

	1	8	2	3	4	5	6	7	9
1	×	×	×		×			×	
8		×	×						
2			×		×			×	
3				×		×		×	
4					×			×	×
5						×		×	
6							×		
7								×	×
9									×

図13 After the last column singleton moved to front of bump.

Ⅱ．スパイクに列シングルトンが存在する場合

Reid法の手順１の①において、スパイク列は列シングルトンの探索範囲から除かれる。なぜなら、このシングルトンをバンプ右上隅に移動した場合には、図１で示される形式が保持されなくなるからである。しかし、このスパイク列とバンプの最後の列を入れ替えることにより、Ⅰの場合の行列が得られ、図１の形式を保持しながらも上で説明したような効率化を図ることが可能となる。そこで、スパイクに列シングルトンが存在する場合には、この列とバンプの最後の列を入れ替えてからこの列シングルトンをバンプ右上隅に移動するように改良する。

４．数値実験

改良Reid法の有効性を調べるために、全体的なシングルトン移動数に注目した実験を行なった。テスト問題としては、現実の大規模問題に近い構造とするために係数行列を疎な帯行列とし、次のようなランダム問題を考えた[4]。

maxmize　$c^t x$

subject to $Ax \leqq b, x \geqq 0$

ここで、ｃとｂは１と２の間の一様乱数を要素とするベクトルである。Ａは帯幅21の帯行列でその係数は、帯の内側では０または１と２の間の一様乱数値を持ち、外側では０の値をとる。Ａの対角要素は非有界性を避けるために１とした。また列平均非零要素数を３とし、最大問題サイズ500とした。

Reid法と改良Reid法のコードを作成し、上の形式の問題を解いた結果の代表的な例を表１に示す。表

表１．数値実験結果

問題のサイズ	反復回数	Reid法			改訂Reid法				
		列シングルトンの移動回数	行シングルトンの移動回数	全移動回数	列シングルトンの移動回数	行シングルトンの移動回数	バンプの最後の列の列シングルトンの移動回数	スパイク列の列シングルトンの移動回数	全移動回数
250	236	3618	137	3755	3170	130	89	8	3397
300	271	5028	192	5220	4009	167	107	22	4305
350	315	5370	242	5612	4730	219	109	26	5084
400	339	4578	150	4728	4330	109	140	27	4606
450	407	6103	147	6250	5363	157	149	22	5691
500	441	9619	339	9958	8534	323	169	27	9053

１には、最適解を得るまでの全列シングルトン移動数及び全行シングルトン移動数が両方法について示され、さらに改良Reid法についてはバンプの最後の列シングルトンの全移動数及びスパイク列とバンプの最後の列の全入れ代え数も示している。

表１より各サイズの問題について、改良Reid法で加えた判定及び置換が実際に生じており、Reid法に比べ改良Reid法が全シングルトン移動数について約１０％～２０％減少していることがわかる。また、全シングルトン移動数の減少という結果からシングルトン探索を行なった行及び列の数は、Reid法に比べ改良Reid法では減少したと考えられる。

６．おわりに

大型疎線形計画問題の優れた解法とされているReid法のより効率的なバンプ縮小方法として、スパイク列とバンプの最後の列の置換を積極的に行なう改良Reid法を提案した。そして、提案方法の有効性を調べるための実験を行なった結果、全シングルトン移動数が約１０％～２０％減少していることがわかり、改良Reid法の有効性が確認された。また、この方法により不必要なシングルトン移動が削減されたので、シングルトン探索を行なわなければならない行及び列の数はReid法に比べ減少したと考えられる。従って、シングルトンの探索も効率的になったと思われる。

今後、今回実験を行なった構造以外の大型疎線形計画問題について検討する予定である。

参考文献

[1]R.H.Bertels and G.H.Goub:"The simplex method of linear programing using LU decomposition",communication of the ACM.

[2]J.K.Reid:"A sprsity-exploiting variant of the Bartels-Golub decomposition for linearprogramming bases",Mathematical Programming Vol 24 No.1.

[3]V.Chvatal:"Linear Programming",W.H.Freeman and Company.

[4]大堀、大内:"スパース処理法を用いたカーマーカー法による大規模線形計画問題の解法",T.IEE Japan Vol 109-C No 11,'89