Vol.2017-HPC-159 No.3
2017/4/17

色数を抑えた改良Reverse Cuthill-McKee法による線形ソルバの並列化について

俵谷 健太郎[1]　横川 三津夫[1,a)]

概要：偏微分方程式を有限差分法などで離散化する場合，大規模な連立一次方程式を解く問題に帰着される．連立一次方程式を解く時間は，全体計算時間の大部分を占めることが多く，計算時間短縮のため，前処理付き反復法がよく用いられる．前処理では，並列計算が可能となるようにループ内のデータ依存関係をなくす様々なマルチカラー法による変数のオーダリングが提案されてきた．
本稿では，rereverse Cuthill-McKee 法（RCM 法）に焦点をあて，並列化のオーバーヘッドを低減させる改良案として，一番最初の色付けを複数のノードから開始する複数初期点 RCM 法（MIP-RCM 法）を提案した．対称ガウス・ザイデル前処理付共役残差法に対して，MIP-RCM 法と従来の RCM 法による計算時間の比較を行った結果，8 つの問題において MIP-RCM 法の計算時間が短かった．特に，thermal2 の問題の thread 並列計算における反復計算時間の比較では，RCM 法の最も短い 2threads の計算時間に対し，MIP-RCM 法は 12threads で約 2.93 倍の時間短縮が達成され，提案手法の有効性が確認できた．

キーワード：並列解法，マルチカラー法，reverse Cuthill-McKee 法，複数初期点

An improved reverse Cuthill-McKee method for parallel linear solvers

Kentaro Hyotani[1]　Mitsuo Yokokawa[1,a)]

Abstract: A large-scale linear system of equations often appears in solving partial differential equations by discretizing with finite difference method, etc. Since computation time for solvers of the system occupies the most of the whole calculation time, preconditioned iterative linear solvers are often used to reduce the calculation time. As for preconditioning methods, various multi-color ordering methods have been proposed to vanish data dependencies in the preconditioning procedure.
In this paper, we proposed a multiple initial-points reverse Cuthill-McKee (MIP-RCM) method as an improvement of the RCM method to reduce the overhead of parallelization. Numerical experiments showed that the computation times of the MIP-RCM method was shorter than those of the RCM method for eight linear systems with different matrices. In particular, the computation time of the solver with MIP-RCM method was shorter than that with RCM method. We confirmed that MIP-RCM method is efficient compared to RCM method.

Keywords: Parallel computation, Multicolor method, Reverse Cuthill-McKee ordering, Multiple initial points for coloring.

1. はじめに

計算機シミュレーションによる解析においては，現象等を表現する偏微分方程式を有限差分法などで離散化し，大規模な連立一次方程式を解く問題に帰着させることが多い．連立一次方程式を解く時間は，全体計算時間の大部分を占めることが多く，高速な解法や並列計算による計算時間の短縮が求められている．

大規模疎行列を係数行列に持つ連立一次方程式の解法には，前処理付き共役勾配法や共役残差法が用いられること

[1] 神戸大学大学院システム情報学研究科
[a)] yokokawa@port.kobe-u.ac.jp

が多い．前処理手法として，対称ガウス・ザイデル法（SGS 法や不完全コレスキー分解法（IC 分解法）が有名だが，通常の変数のオーダリングでは，前処理アルゴリズムの計算で現れる変数に正の依存関係があるため，逐次処理をしなければならず，そのままの形では並列計算が出来ない．このため，変数のオーダリングを変えることによりこの依存関係を排除し，並列計算が出来るような多くのオーダリング手法が提案されてきた．例えば，red-black ordering，hyperplane ordering，reverse Cuthill-McKee（RCM）法，cyclic multicoloring-RCM 法，代数ブロック化多色順序付け法などがある [1], [2], [3], [4]．これらの中で，RCM 法は収束への影響は少ないが，並列化のオーバーヘッドが大きいと言われている [1], [2]．

本稿では，RCM 法の並列化のオーバーヘッドを低減させる改良案として，一番最初の色付けを複数のノードから開始する改良 RCM 法（複数初期点 RCM 法，MIP-RCM(p) 法）を提案する [5]．また，MPI-RCM 法について，既存の RCM 法との比較を行い，その性能を評価する．

2. 改良 Reverse CutHill-McKee 法

2.1 Reverse Cuthill-Mckee 法

Cuthill-McKee （CM）法は，本来，疎行列を係数行列に持つ連立一次方程式に対し，行列の帯幅を減らすためのオーダリングアルゴリズムとして考えられた [6]．このアルゴリズムを **Algorithm 1** に示す．CM 法では，複数の色で変数の組を作った後，その色の組でオーダリングする方法である．一つの色の組内での順番は元の変数の順番で良い．**Algorithm 1** において，次数とは，ある節点に接続して節点の数である．

Algorithm 1: Cuthill-Mckee 法

```
k = 1
次数が最小の格子点を k 色目で彩色する．
while 未彩色の格子点が存在する do
    k = k + 1
    k − 1 色目で彩色した格子点に接続する未彩色の格子点を k 色目で彩色する．
end
各色の組に属する格子点を順番に再番号付けをする．
```

Reverse Cuthill-McKee ordering（RCM）法は，CM 法でオーダリングした後，そのオーダリングを逆順に並び替えるアルゴリズムである．コレスキー分解による解法などでは，CM 法によるオーダリングの時よりも RCM 法の方が，fill-in が少なくなることが知られている [6]．

2.2 修正 Reverse Cuthill-McKee 法

CM 法の色分けにおいて，同じ色の要素 $x_i^{(k)}$ $(i = 1, \ldots, n)$ を互いに依存しないような集合に色分けできれば，違う色の変数の間に依存関係がなくなり，同じ色の要素の更新処理は並列に実行できる．この方針に基づく並列化向けに修正を加えた CM 法のアルゴリズムを **Algorithm 2** に示す．

Algorithm 2: 修正 Cuthill-McKee 法

```
k = 1
次数が最小の格子点を k 色目で彩色する．
while 未彩色の格子点が存在する do
    k = k + 1
    k − 1 色目で彩色した格子点に接続する未彩色の格子点のうち，互いに接続しない　格子点のみを k 色目で彩色する．
end
各色の組に属する格子点を順番に再番号付けをする．
```

本報告では，修正 CM 法に対し，さらにそのオーダリングを逆順にした修正 reverse Cuthill-McKee 法（RCM 法と呼ぶ）を比較対象とする．

並列計算において，計算時間に影響を与える要素として，incompatible node（ICN）や色数がある．ICN とは，反復法の更新処理の過程で他の点から影響を受けない点のことである．修正 CM 法の場合は，1 色目の組となる最初の 1 点のみが ICN である．また，色数とは彩色による組分けにおいて最終的に必要となった色数（組）である．色の違う組に属する格子点間にはデータの依存関係があるので，色数が多いと，同時に更新できる変数が少なくなり，高い並列化効率を得ることは困難である．したがって，高い並列性を得るためには，出来るだけ色数は少ない方がよい．

修正 CM 法は 1 点のみを 1 色目で彩色し，それに接続する点から順に塗り広げていくため，色数の多さは離散格子の形状に大きく依存する．2 次元直交格子では，2 つの方向の格子数が同じ場合，格子 1 辺のサイズを N とすると色数は $2N - 1$ となる．修正 CM 法のように，色数が多く，ICN の数が少ない multicolor 法などのオーダリングでは並列性は低いが，red-black ordering 法のような色数の少ない multicolor 法よりも優れた収束性を持つ傾向にあると言われている [1]．

2.3 複数初期点 Reverse Cuthill-McKee 法

RCM 法で課題となる色数（すなわち並列度）について，可能な限り ICN の増加を抑えつつ，色数が少なくなるような方法を考えた．

CM 法は，次数の最も少ない格子点から彩色が始まり，その格子点の周りの格子点へと彩色するため，最初の格子点からの距離が遠い格子点が存在すると，その格子点を彩色するまでに多くの色数が必要となる．したがって，最初の格子点からの距離が遠い格子点も同じ色で彩色し，それらの点から同時に彩色することで，最終的に必要となる色数を減らすことができる．

最初の色で彩色する格子点の数を p とし，この方法で彩色した上にさらに変数の順を逆順にしたものを複数初期点 RCM(p) 法（multiple initial-points RCM 法, MIP-RCM(p) 法）と呼ぶことにした．MIP-RCM(p) 法のアルゴリズム **Algorithm 3** に示す．

このアルゴリズムにおいて，1 色目で彩色する格子点（初期格子点）数を $p = 1$ とした場合は，RCM 法のアルゴリズムと同等となる．

格子点数 $n = 5^2$ の直交格子を MIP-RCM(1) 法，すなわち RCM 法で彩色した様子を図 1 に，MIP-RCM(5) 法で彩色した様子を図 2 に示す．これらの場合，それぞれ色数は 9 色，3 色となることが分かる．

特に，2 つの方向の格子数がどちらも $N = 2^{k-1} + 1\,(k = 1, 2, \ldots)$ であるような 2 次元直交格子の場合，選ばれる初期格子点の配置やその後の彩色の順序は規則的になることが分かる．初期点格子は 1 個から $\frac{N^2+1}{2}$ 個まで取ることができ，さらにその初期格子点数 p に応じて色数 c は，

$$c = \frac{N-1}{2^{l-2}} + 1 \tag{1}$$

となる．ここで，

Algorithm 3: MIP-RCM(p) 法

【定義】
n：格子点数（node の数），n 個の node は予め順序付けられているとする．
$\mathbb{N} = \{i \mid i = 1, \ldots, n\}$：初期格子点の候補の集合とする．
$a_{i,j} = \begin{cases} 1 & i \text{ と } j \text{ が接続している時} \\ 0 & \text{otherwise} \end{cases}$
$\mathsf{deg}(i), (i \in \mathbb{N})$：各格子点の次数をとする．
$\mathsf{dis}(i,j), (i,j \in \mathbb{N})$：2 点 i, j 間の最短経路における辺数とする．
$\lambda(i) = n\,(i \in \mathbb{N})$ ：$\lambda(i)$ は初期点から格子点 i までの距離の最小値に相当

初期格子点数 p をパラメータとして与える．
$k = 1$
while $(t > 0) \wedge (\mathbb{N} \neq \emptyset)$ **do**
  $t = t - 1$
  $\max_{i \in \mathbb{N}}\{n\lambda(i) - \mathsf{deg}(i)\}$ となるような格子点 i を k 色目で彩色．
  その i について，$\mathbb{N} = \mathbb{N} \setminus \{i, j \mid a_{i,j} \neq 0, j \in \mathbb{N}\}$,
  $\lambda(j) = \min\{\lambda(j), \mathsf{dis}(i,j)\} \quad (j \in \mathbb{N})$.
end
while 未彩色の格子点が存在する．**do**
  $k = k + 1$
  $k - 1$ 色目で彩色した点に接続する未彩色の格子点のうち，互いに接続しない格子点のみを k 色目で彩色する．
end
各色の組に属する格子点を順番に再番号付けをした後，すべての順番を逆順に並べる．

$$l = \begin{cases} 1 & (p = 1) \\ \left\lfloor \log_2\left(\sqrt{2p-1} - 1\right) \right\rfloor + 2 & (p = 2, \ldots, \frac{N^2+1}{2}) \end{cases} \tag{2}$$

である．

初期格子点を最大の $p = \frac{N^2+1}{2}$ 個に取った場合，色数は 2 色となる．これは red-black ordering 法で彩色したものと同等である．図 2 の場合は，2 つの方向の格子数が $N = 5$, すなわち $2^{k-1} + 1, k = 3$) に相当するので，初期点を $p = 5$ の場合の色数 c は

$$l = \left\lfloor \log_2\left(\sqrt{2 \cdot 5 - 1} - 1\right) \right\rfloor + 2 = 3 \tag{3}$$

$$c = \frac{N-1}{2^{3-2}} + 1 = 3 \tag{4}$$

と求められる．

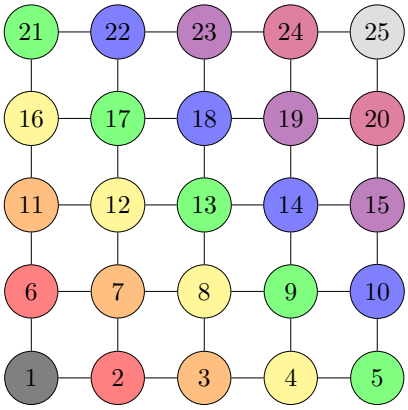

図 1　Orthogonal grid colored by MIP-RCM(1)

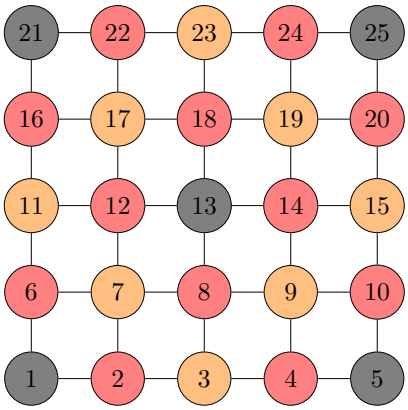

図 2　Orthogonal grid colored by MIP-RCM(5)

3. 数値実験

3.1 実行環境

プログラムは Fortran で実装し，神戸大学の π-VizStudio (SGI UV 300) で OpenMP による thread 並列性能を評価した．π-VizStudio の計算サーバ vizcore の性能を表 1 に

表 1 Specification of π-VizStudio (SGI UV300)

CPU	Intel Xeon E7-8857 v2 * 32
Cores	12
Clock	3.0 GHz
Memory	16 TiB (DDR3 32 GiB * 512)
Compiler	Intel Parallel Studio XE Cluster Edition (ifort version 15.0.3)
Numerical Library	Intel Math Kernel Library(2015.3.187)

表 2 Properties of matrices

Name	Size	Nonzeros	Bandwidth
poisson	1,050,625	5,249,025	1,025
seismic	89,014	3,026,354	88,785
s3rmt3m3	5,357	207,123	5,302
s3dkt3m2	90,449	3,686,223	614
bmw7st_1	141,347	7,318,399	121,856
hood	220,542	9,895,422	219,759
cfd2	123,440	3,085,406	4,332
thermal2	1,228,045	8,580,313	1,226,000

示す．使用したコンパイラは ifort，最適化，並列化，及びライブラリをリンクするためのオプションは -xAVX -O3 -ipo -no-prec-div -qopenmp -parallel -lmkl_intel_lp64 -lmkl_intel_thread -lmkl_core -liomp5 -free -Tf とした．

3.2 実験に用いた行列

提案した方法の評価には，8 種類の行列を用いた．それらの行列のサイズ，非零成分の数，帯幅を表 2 に示す．

行列 poisson は，式 5 のポアソン方程式を，有限差分法により直交格子上で離散化した時に得られる行列である．周囲境界を除く離散格子点数 $n = 1025^2$ である．

$$\begin{aligned} -u_{xx} - u_{yy} &= 2(1-6x^2)y^2(1-y^2) \\ &\quad +2(1-6y^2)x^2(1-x^2) \quad \text{in } \Omega, \qquad (5) \\ u &= 0 \quad \text{on } \partial\Omega. \end{aligned}$$

行列 seismic は，大規模構造物の地震応答解析において，約 2 万節点の有限要素モデルから得られる行列である．これら以外の行列は，University of Florida Sparse Matrix Collection から入手した [7], [8]．いずれも正定値対称な疎行列である．

行列 poisson, seismic，thermal2 以外の行列に対しては，連立一次方程式としての右辺ベクトル $\boldsymbol{b}$ がデータとして与えられていないため，解ベクトル $\boldsymbol{x}$ が $(1, \ldots, 1)^{\mathrm{T}}$ として，右辺ベクトルを設定した．

3.3 反復解法

表 2 の行列を係数行列とする連立一次方程式を，前処理付き反復法で解いた．プログラム内では，行列データを compressed row storage（CRS）方式で格納している．前処理として，対角項スケーリング法 (S)，対称ガウス・ザイデル法（SGS），不完全コレスキー分解法（IC）を用い，反復法には共役勾配法（CG 法），共役残差法（CR 法）を用いた．また，RCM 法と MIP-RCM(p) 法のオーダリングによる thread 並列計算を行った．反復法の初期解を $\boldsymbol{x}^{(0)} = \boldsymbol{0}$，反復法の収束判定を相対残差 L_2 ノルム $\frac{||b-Ax^{(k)}||_2}{||b||_2} \leq 10^{-8}$ とした．反復回数の上限を 20,001 回とし，それまでに解が得られない場合は，“収束しなかった” とした．

表 3 CG and CR

	Iterations		Solve time(sec.)	
	SGSCG	SGSCR	SGSCG	SGSCR
poisson	1143	1074	63.509	64.941
seismic	4967	3977	58.423	47.915
s3rmt3m3	3593	3393	1.625	1.552
s3dkt3m2	14655	8492	202.492	120.737
bmw7st_1	1111	1394	32.798	42.083
hood	1512	1445	61.824	60.408
cfd2	760	440	9.381	5.680
thermal2	2022	1845	175.638	170.396

4. 数値実験結果

4.1 CG 法と CR 法の比較

まず，前処理に SGS を適用した CG 法と CR 法を比較した．8 つの線形方程式に対する反復回数と計算時間を表 3 に示す．いずれも逐次処理による結果である．

反復回数は，bmw7st_1 以外の 7 ケースで SGSCR 法の方が少なく，計算時間は，poisson と bmw7st_1 以外の 6 ケースで SGSCR 法の方が短い．行列はすべて正定値対称であり，1 反復あたりの計算量は前処理付き CR 法が大きいが，今回実験に用いた行列に対しては，SGSCR 法の計算時間が短かかった．これは，非零要素に対する間接参照などの要因によるものと思われる．以下の数値実験では，CR 法を用いた．

4.2 前処理法の比較

CR 法に対し，S，SGS，IC の前処理を適用した時の計算時間を計測した．SGS 前処理と IC 前処理に対しては，RCM 法によるオーダリングを変えた場合の計算時間も計測した．一例として，行列 thermal2 に対する反復回数と計算時間を表 4 に示す．いずれも逐次処理による結果である．表において，S，IC 前処理，及び RCM 法では，対角行列の準備，行列の分解，彩色とその後の並び替えなどの処理が必要なため，それらの時間を Setup 時間として，反復処理の計算時間（Solve）とは別に計測を行った．

表 4 に示すように，すべてのケースで収束しており，最も速いのは ICCR(RCM) であった．8 ケースに対して，IC 前処理を適用した方法は，収束する場合には最も計算時間が短かったが，3 ケースしか収束しなかった．これは IC の近似度が不十分なためと考えられる．全体的には，SGS 前

表 4　Effect of preconditioning (thermal2)

Method	Iterations	Setup (sec.)	Solve (sec.)	Total (sec.)
SCR	3958	0.011	206.815	206.826
SGSCR	1845	0.000	170.396	170.396
SGSCR(RCM)	1811	2.672	147.121	149.793
ICCR	1675	0.373	150.241	150.614
ICCR(RCM)	1599	2.949	130.099	133.048

処理付き CR 法が安定に解が求められており，RCM 法と MIP-RCM(p) 法の比較にはこの方法を用いた．

4.3　MIP-RCM(p) 法の初期点数の影響

MIP-RCM(p) 法において，初期点の個数 p を変えたときの状況を調べるために，SGS 前処理を適用した CR 法に，初期点数を 1 から 1001 まで変えた複数初期点 RCM 法によるオーダリングの変更を施したときの実行時間を計測した．初期点数を，行列 poisson については 1 刻み，それ以外の行列いついては 5 刻みで変化させ，12 スレッドの並列計算を行った．

一例として，図 3，図 4，図 5 に，行列 thermal2 に対する MIP-RCM(p) 法の色数，オーダリングの処理時間，反復に要する時間を示す．

アルゴリズムの通り，初期点数 p を増加させると，すべてのケースで色数が減少した．ただし，彩色のための時間

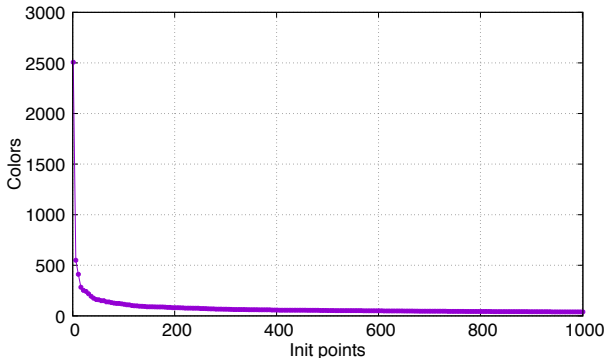

図 3　Number of colors for each number of initial points (thermal2)

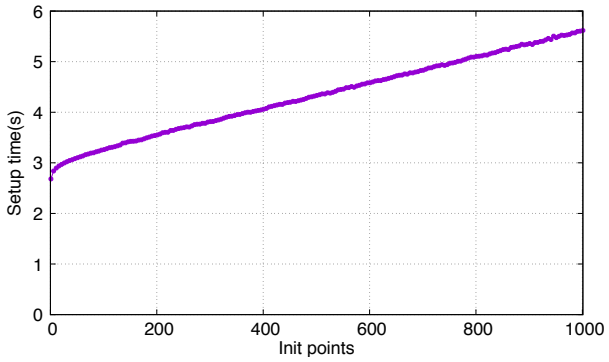

図 4　Setup time for each number of initial points (thermal2)

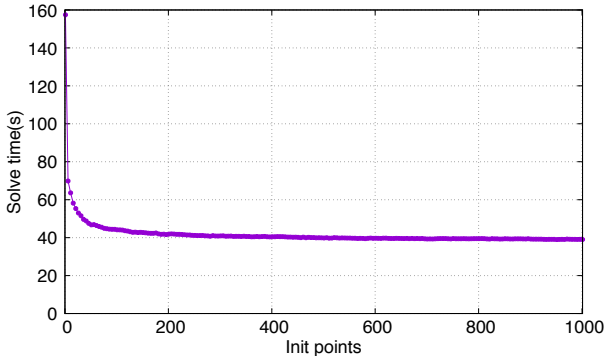

図 5　Solving time for each number of initial points (thermal2)

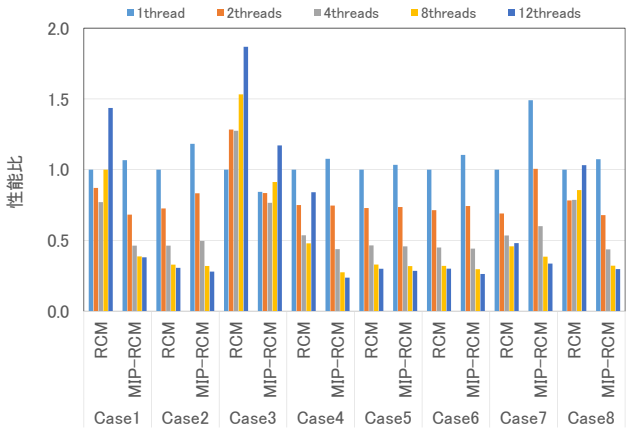

図 6　Calculation time of RCM and MIP-RCM

は増加する．収束までの反復回数と彩色数の間には明確な関係は見られなかった．図 5 に示すように，行列 thermal2 に対しては，彩色数 50 程度まで計算時間が急激に減少している．

今回のケースでは，6 ケースについて計算時間の短縮が認められた．彩色数と反復回数が計算時間に影響していると考えられるが，その関係にについては今後，さらに検討する必要がある．

4.4　MIP-RCM（p) 法と RCM 法の比較

MIP-RCM（p) 法と RCM 法の thread 並列性能を比較するために，表 2 に示したすべての行列に対し，SGS 前処理を適用した CR 法を用いて，thread 数を 1，2，4，8，12 と変えた時の計算時間を測定した．MIP-RCM（p) 法に対する測定では，反復計算時間が最も短かった初期点数 p を用いた．

それぞれの行列に対する Total 時間は，行列サイズに依存して長さが異なるので，RCM 法を用いた時の 1thread の Total 時間（=setup time+solve time）を基準にして，実行時間を比較した．図 6 に，すべての行列に対する RCM 法と MIP-RCM（p) 法の実行時間比を示す．図の横軸の Case 番号は，それぞれの行列に順に対応している．Case3 を除くすべてのケースについて，MIP-RCM（p) 法の 12threads

表 5　Calculation time of RCM and MIP-RCM for thermal2

Method	Number of Threads	Setup (sec.)	Solve (sec.)	Total (sec.)
RCM	1	2.672	147.121	149.793
	2	2.676	114.548	117.224
	4	2.675	115.104	117.779
	8	2.674	125.497	128.171
	12	2.679	151.862	154.541
MIP-RCM	1	5.496	155.323	160.819
($p = 951$)	2	5.497	96.170	101.667
	4	5.464	59.945	65.409
	8	5.489	42.670	48.159
	12	5.480	39.086	44.566

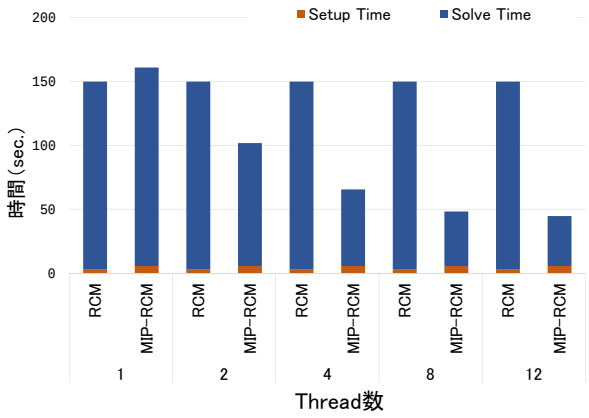

図 7　Calculation time of RCM and MIP-RCM for thermal2

の計算時間が最も短いことが分かった．

表 5，図 7 に，行列 thermal2 のケースに対する setup time, solve time, total time を示す．MIP-RCM 法と RCM 法の反復回数は，それぞれ 1783，1811 であった．

両方法の setup に要する時間は，thread 数を変えてもほぼ同じである．RCM 法の反復法の計算時間は，2 threads 並列のときが最も計算時間が短く，thread 数をそれより大きくすると計算時間が増加した．一方，MIP-RCM(p) 法では，12 threads 並列まで計算時間が短縮され，12threads 並列時の反復時間は 39.086 sec. と，RCM 法の最も短い反復計算時間と比較して約 2.93 倍の速くなった．

5. まとめ

本稿では，連立一次方程式の並列解法に用いるオーダリングの一つである RCM 法を取り上げ，並列化のオーバーヘッドを低減させる改良案として，MIP-RCM(p) 法を提案し，その性能評価を行った．

数値実験により，今回対象とした 8 ケースの行列に対し，MIP-RCM(p) 法は RCM 法よりも計算時間の短縮が見られ，かつ並列化向上率もよいことが明らかとなった．特に，行列 thermal2 に対しては，MIP-RCM(p) 法の実行時間が 12 threads 並列まで短縮され，RCM 法との最も短い反復実行時間と比較して，約 2.93 倍の高速化が達成された．

MIP-RCM 法の今後の課題として，最適な初期点数の選択方法，初期点の順序付けなどがあげられる．RCM 法の彩色数は，離散格子のサイズ（格子点数）と直径（2 点 i, j 間の最短経路における辺数を距離 $\mathsf{dis}(i, j)$ としてときの最大値）など行列の隣接グラフとしての性質が大きく関係している．MIP-RCM(p) 法では，直径が大きい場合に複数の格子点を最初の色とすることで彩色数を抑えることが出来ており，行列の隣接グラフとの関係の検討が必要である．

参考文献

[1] 岩下武史，高橋康人，中島　浩：代数ブロック化多色順序付け法による並列化 ICCG ソルバの性能評価，情報処理学会研究報告，Vol. 2009-HPC-121, No. 11, pp. 1–8 (2009).
[2] 中島研吾：前処理付きマルチスレッド並列疎行列ソルバー，情報処理学会研究報告，Vol. 2013-HPC-139, No. 6, pp. 1–6 (2013).
[3] 中島研吾：拡張型 ELL 行列格納手法に基づくメニィコア向け疎行列ソルバー，情報処理学会研究報告，Vol. 2014-HPC-147, No. 3, pp. 1–8 (2014).
[4] 大島聡史, 松本正晴, 片桐孝洋, 塙　敏博, 中島研吾：様々な計算機環境における OpenMP/OpenACC を用いた ICCG 法の性能評価，情報処理学会研究報告，Vol. 2014-HPC-145, No. 21, pp. 1–10 (2014).
[5] 俵谷健太郎：連立一次方程式の並列解法に用いる Reverse Cuthill-McKeehou の改良と性能評価（神戸大学 2016 年度修士論文）．
[6] Golub, G. H. and Loan, C. F. V.: *Matrix Computations (4th edition)*, pp. 602–604, JHU Press (2013).
[7] Davis, T. A. and Hu, Y.: The university of Florida sparse matrix collection, *ACM Transactions on Mathematical Software*, Vol. 38, No. 1, pp. 1–25 (2011).
[8] Davis, T. A. and Hu, Y.: The SuiteSparse Matrix Collection, Texas A&M University (online), available from ⟨http://faculty.cse.tamu.edu/davis/matrices.html⟩ (accessed 2017-3-15).