

目次

1	はじめに	1
2	正定値行列補完を用いた準ニュートン法 (MCQN 法)	1
2.1	準ニュートン法とその問題点	2
2.2	MCQN 法	2
3	一般の問題に対する MCQN 法の実装方法	5
3.1	コーダル拡張の求め方	5
3.2	RIP を満たす極大クリーク族の求め方	7
4	数値実験	9
4.1	実験で用いる問題	9
4.2	実験結果	9
5	まとめと今後の課題	11

1 はじめに

本報告書では次の制約なし最小化問題を考える．

$$\begin{aligned} & \text{minimize} && f(x) \\ & \text{subject to} && x \in R^n \end{aligned} \tag{1}$$

ここで、 f は R^n から R の 2 回連続微分可能な関数とする．本報告書では n が大きく、ヘッセ行列 $\nabla^2 f(x)$ が疎な行列、つまりほとんどの成分が 0 となる場合を考える．

制約なし最小化問題の解法としては、準ニュートン法、信頼領域法、共役勾配法がある [2, 3, 6]．その中でも、準ニュートン法は、実装が容易であり収束の性質も良いので、よく用いられている．準ニュートン法はヘッセ行列の逆行列を近似した行列 H_k を用いて次の点 $x^{k+1} = x^k + \alpha^k d^k$ を生成する反復法である．近似行列 $\{H_k\}$ を保持するためには一般には $O(n^2)$ の計算容量が必要である．そのために、準ニュートン法を大規模な問題に対して実装することは困難である．

大規模な最小化問題では、一般にヘッセ行列が疎になる．その疎性を利用した近似行列 H_k の更新規則があれば、計算容量を大幅に減らすことが期待できる．最近、山下 [10] はヘッセ行列の疎性と最新の行列補完の結果を利用した新しい H_k の更新規則 (Matrix Completion Quasi Newton method, MCQN) を提案した．この手法では、ヘッセ行列が疎になる場合、1 回の反復に必要とする計算容量、計算時間は BFGS 法に比べて劇的に少なくなる．山下はいくつかの特別な構造をもった最小化問題に対して数値実験を行い、提案した手法が非常に有効であることを報告している．しかし、一般の最小化問題に対して、この手法の性質は解明されていない．そこで、本報告書では、より一般的な最小化問題に対して MCQN の実装をし、数値実験を行うことによって MCQN の性質を調べる．

本報告書の構成は以下の通りである．まず、2 節において正定値補完を用いた準ニュートン法 (MCQN) を紹介する．次に、MCQN の実装方法を 3 節で提案する．4 節では、数値実験結果を報告し、MCQN の性質を考察する．最後に 5 節において、まとめと今後の課題を述べる．

なお、本報告書では以下の記号を用いる．正方行列 H に対して、 $H \succeq 0$ は H が半正定値行列であることを表す． $V = \{1, 2, \dots, n\}$ とする． V の部分集合 S と T と行列 $H \in R^{n \times n}$ に対して、 $n \times n$ 行列 H から $i \notin S$ であるすべての行と $j \notin T$ であるすべての列を取り除いた部分行列を H_{ST} で表す．

2 正定値行列補完を用いた準ニュートン法 (MCQN 法)

この節ではまず、準ニュートン法、特に BFGS 法の欠点について論じる．次に準ニュートン法の欠点を補うために提案された正定値行列補完を用いた準ニュートン法 (MCQN 法) [10] を紹介する．

2.1 準ニュートン法とその問題点

準ニュートン法の探索方向の計算には，ヘッセ行列の逆行列を近似した H_k が用いられる．よく用いられる準ニュートン法，例えばBFGS法やDFP法では，近似行列 $\{H_k\}$ はセカント条件： $H_{k+1}y^k = s^k$ を満たすように更新される [3]．ここで， $s^k = x^{k+1} - x^k$ であり， $y^k = \nabla f(x^{k+1}) - \nabla f(x^k)$ である．BFGS法では次の問題の解を H_{k+1} とする [6]．

$$\begin{aligned} \min_H \quad & \|H - H_k\|_{W,F} \\ \text{subject to} \quad & Hy^k = s^k, H = H^T, \end{aligned} \quad (2)$$

ここで W は $Ws^k = y^k$ を満たす任意の対称行列であり， $\|\cdot\|_{W,F}$ は W の重みつきフロベニウスノルム¹ である．問題 (2) の解は次のBFGS公式で陽に計算することができる．

BFGS 公式：

$$H_{k+1} = H_k - \frac{H_k y^k (s^k)^T + s^k (H_k y^k)^T}{(s^k)^T y^k} + \left(1 + \frac{(y^k)^T H_k y^k}{(s^k)^T y^k}\right) \frac{s^k (s^k)^T}{(s^k)^T y^k}$$

H_k と s^k, y^k が与えられたとき， H_{k+1} は $O(n^2)$ の計算で求めることができるので，ニュートン法と比べて，1回の反復で必要となる計算量が小さい．しかし， $s^k (s^k)^T$ などの影響で H_{k+1} は密な行列になるので，行列 H_{k+1} の保持には $O(n^2)$ のメモリーが必ず必要となる．そのため，超大規模な問題に対してはBFGS法を使うことができない．

BFGS法の欠点を克服するため，記憶制限つきBFGS法が提案されている [5]．しかし，この手法は理論的に1次収束しか保証されていない．また，数値実験においても，ヘッセ行列の条件数が悪い問題では，収束が非常におそくなることが報告されている．

2.2 MCQN 法

次元 n が大きい問題では，目的関数のヘッセ行列が疎になることが多い．しかし，一般の準ニュートン法ではヘッセ行列の疎性は無視されていた．例えば，BFGS法では H_{k+1} を与える問題 (2) において，セカント条件と行列の対称性に関する制約はあるが，疎性に関する制約はない．もし，ヘッセ行列の疎性の情報を利用できたら，BFGS法の計算容量に対する欠点は解消され，計算時間も短縮できるはずである．そのことに着目し，山下 [10] はMCQN法を提案した．

[10] では，まず，問題 (2) に対して疎性を条件に加えた次の問題を考えた．

$$\begin{aligned} \min_H \quad & \|H - H_k\|_{W,F} \\ \text{subject to} \quad & Hy^k = s^k, H = H^T \\ & (H^{-1})_{ij} = 0, (i, j) \notin F \\ & H \succeq 0 \end{aligned} \quad (3)$$

¹ $\|A\|_{W,F} = \|W^{1/2} A W^{1/2}\|_F, \|C\|_F = \sqrt{\sum_i \sum_j C_{ij}^2}$

ここで, $F \supseteq E = \{(i, j) \mid \text{ある } x \in R^n \text{ において } \nabla^2 f(x)_{i,j} \neq 0\}$ である. 本報告書では, E をヘッセ行列の疎構造と呼び, F を E の拡張と呼ぶことにする (なるべく $F = E$ となるように F を選ぶことが望ましい). 残念ながら, 問題 (3) の解は問題 (2) に対する BFGS 公式のように陽に与えることができない. そこで, 問題 (3) の近似解を H_{k+1} とすることを考える.

[10] では以下の手順で問題 (3) の近似解 H_{k+1} を求める準ニュートン法を提案した.

MCQN 法

第 1 段階: 制約条件 $(H^{-1})_{ij} = 0, (i, j) \notin F$ を無視した問題 (5) の解を \bar{H} とする (このとき問題 (3) は問題 (2) となるため, \bar{H} は BFGS 法によって求まる.)

第 2 段階: $\bar{H}_{ij}, (i, j) \in F$ をパラメータとした次の問題の解を H_{k+1} とする.

$$\begin{aligned} \min_H \quad & \|H - H_k\|_{W,F} \\ \text{subject to} \quad & H = H^T, H_{ij} = \bar{H}_{ij}, (i, j) \in F \\ & (H^{-1})_{ij} = 0, (i, j) \notin F \\ & H \succeq 0 \end{aligned} \quad (4)$$

問題 (4) において \bar{H} で使われるのは $(i, j) \in F$ の成分だけである. そのため, 第 1 段階においては $\bar{H}(i, j), (i, j) \in F$ だけを計算すれば良い.

問題 (4) は一般には簡単に解くことができない. しかし, E の拡張 F が次に紹介するコーダル性をもつとき, その解を陽に表すことができる [10].

行列の疎性を利用した効率のよい計算には, その疎構造に対応したグラフを利用することが多い [1]. そこで, 集合 F に対して $\bar{F} = F \setminus \{(i, i) \mid i = 1, \dots, n\}$ とし, 頂点の集合を V , 辺の集合を \bar{F} としたグラフ $G(V, \bar{F})$ と考える.

まず, 本報告書で用いるグラフに関する用語を以下に定義する.

定義 1 • 任意の 2 点が接続しているグラフを完全グラフという.

- あるグラフの部分グラフで完全なものをそのグラフのクリークという.
- 他のクリークの真の部分グラフにならないクリークを極大であるという. さらに, あるグラフの極大クリークの集合をそのグラフの極大クリーク族という.
- その点と隣接している頂点がクリークを形成する点を単純という.
- サイクル中でつながっていない 2 点を結ぶ辺をコード (弦) という.
- グラフ中の長さが 4 位上のすべてのサイクルがコードを持てば, このグラフをコーダルグラフという.
- $G(V, \bar{F})$ がコーダルグラフで $\bar{F} \supseteq \bar{E}$ であるとき, $G(V, \bar{F})$ は $G(V, \bar{E})$ のコーダル拡張という.

グラフ $G(V, \bar{F})$ のコーダル性と $\bar{H}_{ij}, (i, j) \in F$ の正定値補完には以下の関係が成り立つ .

定理 1 H_k は正定値行列で $(y^k)^T s^k > 0$ であるとする . さらに , $G(V, \bar{F})$ がコーダルグラフであるとする . このとき , 手順 1 で求めた $\bar{H}_{ij}, (i, j) \in F$ は正定値行列補完できる . さらに , 行列式を最大にする正定値補完行列 H は唯一存在し , $H_{ij}^{-1}, \forall (i, j) \notin F$ となる .

この定理により , $G(V, \bar{F})$ が $G(V, \bar{E})$ のコーダル拡張であれば , 定理 1 より問題 (4) の解が存在することがわかる . 一般のグラフに対して , 要素数が最小となるコーダル拡張を求める問題は NP 完全である . そのため , これまでにいろいろな近似解法やヒューリスティクスが提案されている . 次節では , そのうちの 1 つである最小次数法によるコーダル拡張の求め方を紹介する .

次に $G(V, \bar{F})$ がコーダル拡張であるときに , 問題 (4) の解が陽に計算できる疎な行列の積で表すことができることを示す . $G(V, \bar{F})$ がコーダルグラフであれば , 次の性質を満たす極大クリーク族 $\{C_r | r = 1, \dots, l\}$ が存在する [1] .

- $F = \cup_{r=1}^l C_r \times C_r$.
- Running Intersection Property (以降は RIP と表す): $r = 1, 2, \dots, l - 1$ に対して

$$\exists s \geq r + 1: C_r \cap (C_{r+1} \cup C_{r+2} \cup \dots \cup C_l) \subset C_s$$

を満たす .

この極大クリーク族を用いて , 集合族 $\{S_r | r = 1, \dots, l\}$ と $\{U_r | r = 1, \dots, l\}$ を以下のよ
うに定義する .

$$\begin{aligned} S_r &= C_r \setminus (C_{r+1} \cup C_{r+2} \cup \dots \cup C_l), r = 1, \dots, l \\ U_r &= C_r \cap (C_{r+1} \cup C_{r+2} \cup \dots \cup C_l), r = 1, \dots, l \end{aligned}$$

さらに , S_1, S_2, \dots, S_l の要素を順番にならべたものを $\pi = \{v_1, v_2, \dots, v_n\}$ とする . 例えば , $S_1 = \{4, 5\}, S_2 = \{1, 2\}, S_3 = \{3\}$ であれば , $\pi = \{4, 5, 1, 2, 3\}$ である .

このとき , V から π への順番を入れ替える置換行列を P とすると , 問題 (4) の解 H_{k+1} は以下の式で与えられる [10] .

$$H_{k+1} = P^T L_1^T L_2^T \dots L_{l-1}^T D L_{l-1} \dots L_2 L_1 P \quad (5)$$

ここで , L_r および D は ,

$$[L_r]_{ij} = \begin{cases} 1 & i = j \\ (\bar{H}_{U_r U_r}^{-1} \bar{H}_{U_r S_r})_{ij} & (i, j) \in U_r \times S_r \\ 0 & \text{それ以外} \end{cases}$$

であり ,

$$D = \begin{pmatrix} D_{S_1 S_1} & & & \\ & D_{S_2 S_2} & & \\ & & \dots & \\ & & & D_{S_l S_l} \end{pmatrix}$$

である． ここで

$$D_{S_r, S_r} = \begin{cases} \bar{H}_{S_r, S_r} - \bar{H}_{S_r, U_r} \bar{H}_{U_r, U_r}^{-1} \bar{H}_{U_r, S_r} & r \leq l-1 \\ \bar{H}_{S_r, S_r} & r = l \end{cases}$$

である．

一般にヘッセ行列が疎な場合は C_r, U_r, S_r は要素が少ない集合になる．そのとき，行列 L_r と D は疎な行列となる．そのため，少ない計算容量で MCQN 法は実装することができる．

[10] では，コーダル拡張が簡単に求まる特殊な構造をもった問題に対する数値実験結果が報告されていた．そこでは，条件数が悪い問題であっても，MCQN 法は BFGS 法とほとんど変わらない反復回数で解を得ることができていた．このことは，計算容量が少なくすむ手法としては画期的なことである．しかしながら，コーダル拡張を求めることが難しい一般の問題に対しては数値実験の結果の報告がなく，MCQN 法がどのような振る舞いをするかは明らかにされていない．

3 一般の問題に対する MCQN 法の実装方法

本節では，コーダル拡張が簡単に求めることができない一般の問題に対する MCQN の実装方法を提案する．

MCQN 法の実装には以下の手順が必要である．

MCQN 法の実装手順

- 手順 1 : ヘッセ行列の疎構造 E を求める．
- 手順 2 : $G(V, \bar{E})$ のコーダル拡張 $G(V, \bar{F})$ を求める．
- 手順 3 : RIP を満たす極大クリーク族を計算する．

手順 1 では，目的関数 $f(x), x \in R^n$ が与えられたとき，自動微分を用いれば，ヘッセ行列の疎構造を計算できる．手順 2,3 の詳細については次の副節で述べる．極大クリーク族が求まれば (5) 式によって，MCQN 法を実行することができる．

3.1 コーダル拡張の求め方

コーダル拡張は疎な対称正定値行列のガウスの消去法（コレスキー分解）と関連がある．対称正定値行列 $A \in R^{n \times n}$ が与えられたとき，行列 A の疎構造を W とし， W に対応したグラフを $G(V, \bar{W}), \bar{W} = W \setminus \{(i, i) | i \in V\}$ とする． LL^T を A のコレスキー分解とする．このとき， L は下三角行列とである． L の疎構造を $Q = \{(i, j) : L_{ij} \neq 0 \text{ or } L_{ji} \neq 0, i \neq j\}$ と定義し， Q に対応したグラフを $G(V, \bar{F})$ とする．コレスキー分解の計算の過程で数値的な打消しが起こらないと仮定すると， L の疎構造 Q は行列 A の疎構造 W によって決まる．

特に, $Q \supseteq W$ となり, グラフ $G(V, \bar{Q})$ は $G(V, \bar{W})$ のコーダル拡張となることが知られている. コレスキー分解において, 新たに加えられた辺 ($\bar{Q} \setminus \bar{W}$ に属する辺) を充填 (fill-in) と呼ぶ. 充填の数は, コレスキー分解の計算過程における行 (または列) の順序に依存する. 最小のコーダル拡張を求めることは, 充填の数を最小とする順序を求めることと等価である. しかし, 充填が最小となる順序を求める問題は NP-完全であることが知られている. そのため, ヒューリスティックスを用いて充填をなるべく少なくする手法が提案されており, 中でも Minimum degree ordering algorithm (最小次数法) と Nested dissection ordering algorithm は有名である. 本報告書では, $G(V, \bar{E})$ のコーダル拡張を求める手法として, 実装の容易な最小次数法を用いることを提案する.

最小次数法では次数の小さい頂点をグラフから除去し, その除去された部分グラフを作る. 頂点がすべて除去されるまで同様の操作を繰り返すアルゴリズムである. この除去される順番が, コレスキー分解の計算過程における行の順番に対応する.

最小次数法の詳細は以下の通りである.

最小次数法 [9, 4]

ステップ 1 : $V_0 = V$, $E_0 = \bar{E}$, $i = 1$ とする.

ステップ 2 : $G_{i-1} = G(V_{i-1}, E_{i-1})$ とし, グラフ G_{i-1} において次数が最小の頂点を v_i とする. $S(i) = v_i$ とする.

ステップ 3 : $G_{i-1} = G(V_{i-1}, E_{i-1})$ から v_i を削除し, 新しいグラフ $G_i = (V_i, E_i)$ を作る.

ステップ 4 : $i = i + 1$ とし, $i > |V|$ ならば終了. そうでなければ, ステップ 2 へ.

このアルゴリズムを適用すると頂点の順番は S に保存される. S の第 i 要素が第 i 番目の頂点を表している. 図 1 に最小次数法の実行例を示す. この図では, 最小次数法による頂点の順番は (c, e, a, b, d, f) となる.

最小次数法は充填が少なくなるようにグラフの頂点の順序付けをする手法であり, 充填を求める手法ではない. 最小次数法による頂点の順序に基づいてグラフの充填を求める, つまりコーダル拡張を求めるには, 次の Parter が提案した手法 [7] を用いればよい. ここで, 最小次数法による頂点の順番を $S = (v_1, v_2, \dots, v_n)$ とする.

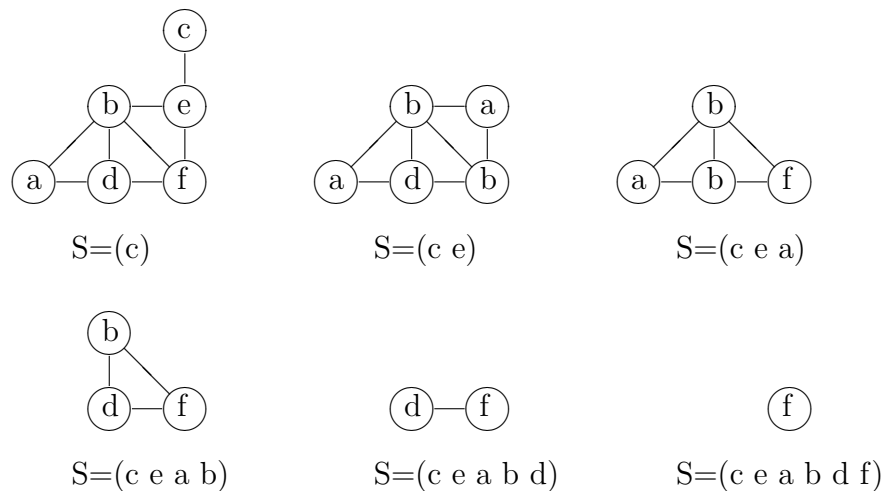


図 1: 最小次数法によりラベル付け (c e a b d f というオーダリングになる)

順番 S からコーダルグラフを求めるアルゴリズム

ステップ 1 : $i = 1$ とする .

ステップ 2 : $i > n$ ならば終了 , そうでなければノード v_i とそれについている枝を $G(V, \bar{E})$ から削除する .

ステップ 3 : $\text{Adj}(v_i)$ が完全グラフとなるように枝を追加する . ただし , $\text{Adj}(v_i)$ は v_i に隣接するノードの集合 .

ステップ 4 : $i = i + 1$ とし , ステップ 2 へ .

このアルゴリズムにより追加された枝が充填に対応する . そして , この枝を加えたグラフは求めたいコーダル拡張に他ならない .

3.2 RIP を満たす極大クリーク族の求め方

RIP を満たした極大クリーク族を求めるために , まず , コーダルグラフ $G(V, \bar{F})$ から極大クリーク族を求める . 次にクリーク木を用いて RIP を満たすように極大クリーク族の並べ替えをする .

コーダルグラフから極大クリーク族とコーダル木を求めるのには , 以下で定義する完全消去順序を求める必要がある . コーダルグラフは , 必ず単純な頂点を持ち , コーダルグラフから単純な頂点を消去した部分グラフもコーダルグラフとなる . この単純な頂点の消去によって $G(V, \bar{F})$ の頂点に順序付けをすることができる . この順序を完全消去順序と呼ぶ .

完全消去順序の計算は, Tarjan と Yannakakis が提案した Maximum Cardinality Search アルゴリズム (以下は MCS 法) [8] を用いれば, $O(|V| + |\bar{F}|)$ で求めることができる.

MCS 法

ステップ 1 : $\text{mark}(v_l) = \text{label}(v_l) = 0, l = 1, 2, \dots, n . i = n$ とする .

ステップ 2 : $i = 0$ ならば終了, さもなければ, $\text{mark}(v_j) = 0$ となる頂点から $\text{label}(v_j)$ が最大となる v_j を選び, $\delta(j) = v_j, \text{mark}(v_j) = 1$ とする .

ステップ 3 : $\text{Adj}(v_j)$ の $\text{mark}(v_k) = 0$ となるすべての頂点 v_k に対して, $\text{label}(v_k) = \text{label}(v_j) + 1$ とする . ただし, $\text{Adj}(v_j)$ は v_j に隣接する頂点の集合である .

ステップ 4 : $i = i - 1$ とし, ステップ 2 へ .

この MCS 法によって計算される完全消去順序は δ である .

極大クリーク族の構造は各ノードが極大クリークになっている木を使って表現することができる . 以下にクリーク木の定義を与える .

定義 2 コーダルグラフ $G(V, \bar{F})$ の極大クリーク族を κ とする . κ の要素を次の性質を満たすようにつなげた木をクリーク木と呼ぶ .

Clique-Intersection Property : τ は木を表すものとする . すべての相異なるクリーク C_i, C_j 属する κ において, それらの共通部分 $C_i \cap C_j$ は, τ において C_i と C_j を結ぶパス上にあるすべてのクリークに含まれる .

RIP は, クリーク木の頂点の 1 つを根とし, 葉から順に番号をつけることで表現できる [1] .

コーダルグラフとその完全消去順序から, 極大クリーク族とクリーク木は以下のアルゴリズムによって求めることができる .

完全消去順序 δ から極大クリーク族とクリーク木を求めるアルゴリズム

ステップ 1 : $r = 1, C_1 = \{v_n\}, K_n = \{C_1\}, i = n - 1$ とする .

ステップ 2 : $i = 0$ ならば終了, さもなければ $A_i = \text{Adj}(v_i) \cap \{v_{i+1}, \dots, v_n\}$ とする . $C_q \supseteq A_i$ となるような $C_q \in K_{i+1}$ を求める .

ステップ 3 : $C_q = A_i$ ならば, $C_q = C_q \cup \{v_i\}$ とする . さもなければ, $r = r + 1,$ $C_r = A_i \cup \{v_i\}, \text{parent}(C_r) \equiv C_q, K_i = K_{i+1} \cup \{C_r\}$ とする .

ステップ 4 : $i = i - 1$ とし, ステップ 2 へ .

このアルゴリズムは極大クリーク族 $\{C_r\}$ を求めると同時にクリーク木を求めている . クリーク木は各ノード (クリーク) の親ノード (クリーク) を表す関数 parent によって表現されている . RIP 条件を満たす極大クリーク族の順番は, このクリーク木の葉から順に番号を付けることによって求めることができる .

4 数値実験

この節では、疎構造 E をランダムに生成した問題に対して、前節で提案した実装方法に基づいた MCQN 法の数値実験結果を報告する。数値実験では MCQN と BFGS 法、記憶制限つき BFGS 法の比較をした。アルゴリズムは Matlab6.5 を用いて実装した。

4.1 実験で用いる問題

数値実験では以下の問題を解いた。

$$\text{問題 1 : } f(x) = x^T A x + b^T x$$

$$\text{問題 2 : } f(x) = x^T A x + b^T x + \sum_{i=1}^n \sin(x_i)$$

$$\text{問題 3 : } f(x) = x^T A x + b^T x + \sum_{i=1}^n \exp(x_i)$$

ここで、行列 A はランダムに生成された正定値行列である。行列 A の生成は Matlab の関数 `sprandsym` を用いて、以下の式によって求めた。

$$A = \text{sprandsym}(n, \text{ratio}, rc, 2) + rc \begin{pmatrix} 2 & 1 & \cdots & 0 \\ 1 & 2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 2 \end{pmatrix}$$

関数 `sprandsym` は、条件数が $1/rc$ となり、非ゼロ要素率 (=非ゼロ要素の数/ n^2) が ratio となるようにランダムに $n \times n$ の対称正定値行列を生成する関数である。目的関数 f が分解構造を持つとき、つまり、ヘッセ行列がブロック対角行列であらわすことができるときに、MCQN 法は著しくよい性質を持つ。そのため、本実験では、ランダムに生成した行列がブロック対角にならないように上式の第 2 項を用いた。行列 b は各成分を $[0, 1]$ からランダムに選んだ n 次元ベクトルである。なお、問題 1 は凸 2 次計画問題であり、問題 2 と 3 は非線形計画問題である。

4.2 実験結果

本実験では、 A と b を 10 回生成して各問題を作成し、それぞれの問題の解を求めるのに必要となった反復回数の平均値を求めた。終了条件には、

$$\|\nabla f(x^k)\| < 10^{-5}$$

を用いた。

まず、 $rc = 0.001$, $\text{ratio} = 3/n$ (各行 (列) の非ゼロ要素が平均で 3 個ある) と固定して、次元数を 10, 100, 1000 とした実験結果を表 1 にまとめる。なお、L-BFGS は記憶制限つき BFGS 法のことである。この表より、ほとんどの場合において、MCQN 法は BFGS 法と L-BFGS 法より反復回数が少ないことがわかる。しかし、L-BFGS 法と比較して圧倒的に

問題	n	MC-BFGS 法	BFGS 法	L-BFGS 法
問題 1	10	18.8	50.0	50.0
	100	91.9	167.0	278.7
	1000	167.6	166.8	380.5
問題 2	10	17.2	17.9	21.2
	100	33.5	54.8	50.7
	1000	70.8	91.1	121.4
問題 3	10	22.9	46.0	50.0
	100	67.7	182.0	354.4
	1000	202.6	191.9	409.3

表 1: 各問題に対する実験結果

rc	MCQN 法	BFGS 法	L-BFGS 法
0.01	50.9	53.9	90.3
0.001	94.4	163.5	279.9
0.0001	100.3	327.9	875.9

表 2: 条件数に対する実験結果

優れているわけではない。L-BFGS 法は、実装が容易であり、各反復にかかる計算量、計算容量は MCQN 法よりも少ない。そのため、この実験から MCQN が L-BFGS 法と比較することはできない。

そこで、L-BFGS 法が不得意である条件数が悪い場合の実験を行った。表 2 に、 $rc = 0.01, 0.001, 0.0001$ の 3 つの場合の結果を載せる。なお、解いた問題は問題 1 ($n = 100$) であり、 $ratio$ は 0.01 とした。表 2 より、条件数が極めて悪いとき、L-BFGS 法と比較して MCQN 法が非常に良いことがわかる。

最後に $n = 100$ と $rc = 0.001$ と固定して非ゼロ要素率を変化させて場合の実験結果を表 3 にまとめる。表 3 より、BFGS 法の収束性はヘッセ行列の疎性との関係がほとんど見られないのに対して、MCQN 法では疎な問題であるほど、収束が速くなることがわかる。

非ゼロ要素率	MCQN 法	BFGS 法	L-BFGS 法
$3/n$	98.0	166.6	299.4
$10/n$	183.1	144.7	326.8

表 3: 非ゼロ要素率に対する実験結果

5 まとめと今後の課題

本報告書では、最近提案された MCQN 法の実装方法を考えた。その実装方法に基づいて、一般の問題に対して数値実験を行った。数値実験結果から、MCQN はヘッセ行列の条件数が極めて悪いとき、BFGS 法と記憶制限つき BFGS 法に比べて、とても有効な手法であることを確認することができた。

しかし、MCQN 法には数多くの研究課題が残されている。以下にその課題をまとめる。

- 計算時間の比較

本数値実験では、反復回数について、MCQN 法と BFGS 法、記憶制限つき BFGS 法の比較を行った。しかし、今後、計算時間についての比較を行う必要もある。そのために、Matlab ではなく、例えば、C 言語でプログラムを実装する必要がある。

- 複雑な問題での数値実験

本研究では、凸 2 次計画問題に対して実験を行った。もっと複雑な問題に対して、MCQN の振る舞いを調べる必要がある。

- 疎構造 E の計算の自動化

自動微分を用いて、自動的に疎構造を計算するアルゴリズムの実装は今後の課題である。

- 収束率の解明

数値実験では MCQN で生成された点列は超 1 次収束のような振る舞いをすることが報告されている [10]。しかしながら、理論的にはまだ証明されていない。

謝辞

本報告書を作成するにあたり，ご指導してくださいました福島雅夫教授，山下信雄助手に感謝いたします．また，大変お世話になった福島研究室の皆様に厚く御礼申し上げます．

参考文献

- [1] J.R.S. Blair and B. Peyton, *An introduction to chordal graphs and clique trees*, in: A. George, J.R. Gilbert and J. W. H. Liu, eds., *Graph Theory and Sparse Matrix Completion*, pp. 1–29, Springer-Verlag, New York, 1993.
- [2] J.E Dennis JR and Robert B.Schnabel, *Numerical methods for unconstrained optimization and nonlinear equations*, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1983
- [3] 茨木俊秀・福島雅夫, 最適化の手法, 共立出版株式会社, 1993.
- [4] A.George and J.W.Liu, *Computer solution of large sparse positive definite systems*, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1981.
- [5] J. Nocedal, *Updating quasi-Newton matrices with limited storage*, *Mathematical Computing*, 35, pp. 773-782, 1980.
- [6] J. Nocedal and S.J. Wright, *Numerical Optimization*, Springer-Verlag New York, Inc., New York, 35, 1999.
- [7] S.V.Parter, *The use of linear graphs in Gauss elimination*, *SIAM Review*, 3, pp. 119–130, 1961.
- [8] R.E.Tarjan and M.Yannakakis, *Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs*, *SIAM J. Computing*, 13, pp. 566–579, 1984.
- [9] W.F.Tinney, *Comments on using sparsity techniques for power system problems*, *Sparse Matrix Proceedings*, IBM Research Report. RAI 3-12-69, 1969.
- [10] 山下信雄. 大規模な制約なし最小化問題に対する準ニュートン法, 第16回RAMPシンポジウム予稿集, 2004.