

特別研究報告書

大規模な制約なし最小化問題に対する
コーダル部分グラフを用いた
スパース準ニュートン法

指導教員 福嶋雅夫 教授
山下信雄 助教授

京都大学工学部情報学科
数理工学コース
平成15年4月入学
平成19年3月卒業

黒川 典俊

平成19年1月31日提出

大規模な制約なし最小化問題に対する コーダル部分グラフを用いたスパース準ニュートン法

黒川 典俊

摘要

本報告書では、大規模な制約なし最小化問題に対する準ニュートン法を取り扱う。準ニュートン法が中小規模の問題に対して有効な解法であることはよく知られているが、各反復で更新される近似ヘッセ行列は密になるため、大規模な問題に対して準ニュートン法を適用するには何らかの工夫が必要である。大規模な問題では、目的関数のヘッセ行列は一般に疎な行列となる。その疎性を利用した準ニュートン更新の手法として、Matrix Completion Quasi-Newton method (行列補完を用いた準ニュートン法, MCQN 法) が提案されている。

ヘッセ行列の疎構造に対応するグラフ(疎構造グラフ)がコーダルグラフとなるとき、MCQN 法で更新される近似行列は、疎な三角行列の積として表すことができる。疎構造グラフがコーダルグラフとならないとき、従来の MCQN 法では、疎構造グラフのコーダル拡張を用いて行列の更新を行っている。そのため、問題によっては、ヘッセ行列の非ゼロ要素数と比べて近似行列の非ゼロ要素数が大幅に増え、必要とする記憶量と計算時間が増大してしまう。

上記の問題を解決するため、本報告書では疎構造グラフのコーダルな部分グラフを用いて準ニュートン更新を行う手法を提案する。この手法を用いると、近似行列の非ゼロ要素数はヘッセ行列の非ゼロ要素数以下に抑えることができるので、実装に必要な記憶量と各反復の計算時間を減らすことができる。提案手法が速く収束するためには、元の疎構造グラフの枝をより多く含むようなコーダル部分グラフを用いることが望ましい。本報告書では、そのような部分グラフを求める工夫についても考察する。また、いくつかの数値実験結果から提案手法の有効性を議論する。

目次

1	序論	1
2	行列の疎性とコーダルグラフ	2
2.1	コーダルグラフの基本的性質	2
2.2	行列の疎性とコーダルグラフ	4
2.3	コーダル拡張を求めるアルゴリズム	5
2.4	コーダル縮小を求めるアルゴリズム	6
3	コーダル部分グラフを用いた準ニュートン法	8
3.1	準ニュートン法とその問題点	8
3.2	コーダル拡張を用いた MCQN 法とその計算量	9
3.2.1	Ext-MCQN 法の概要	9
3.2.2	Ext-MCQN 法の計算量	11
3.3	コーダル縮小を用いた MCQN 法とその計算量	12
3.3.1	Del-MCQN 法の概要	12
3.3.2	Del-MCQN 法の計算量	14
4	数値実験	16
5	まとめと今後の課題	19
A	コーダルグラフに関するアルゴリズム	22
A.1	MCS アルゴリズム	22
A.2	RIP を満たす極大クリーク族とクリーク木を求めるアルゴリズム	22
B	半正定値行列補完	22
C	4 節【実験 2】の結果	24

1 序論

次の制約なし最小化問題を考える．

$$\begin{aligned} & \text{minimize} && f(x) \\ & \text{subject to} && x \in \mathbb{R}^n \end{aligned} \tag{1.1}$$

ここで、 $f: \mathbb{R}^n \rightarrow \mathbb{R}$ は 2 回連続的微分可能な関数とする．本報告書では、特に n が大きく、 f のヘッセ行列が疎、つまりヘッセ行列のほとんどの成分が 0 となる場合を取り扱う．

制約なし最小化問題の解法としては、最急降下法、ニュートン法、準ニュートン法、共役勾配法などがある．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、広く用いられている [5, 9, 14]．準ニュートン法は、目的関数のヘッセ行列（もしくはその逆行列）の近似行列を用いて、点列を生成する反復法である．近似行列の更新公式としては BFGS 公式や DFP 公式が代表的である．準ニュートン法によって生成される点列はある条件下で超一次収束することが知られている．よって、準ニュートン法を用いれば、比較的少ない反復回数で問題 (1.1) の解を得ることができる．しかし、ヘッセ行列が疎であっても、BFGS 公式や DFP 公式で生成された近似ヘッセ行列は一般に密な行列となるため、 n の 2 乗に比例した記憶容量が必要となる．そのため、これまで大規模な問題には準ニュートン法を適用することができなかった．

大規模な最小化問題では、一般にヘッセ行列が疎になる．そのため、ヘッセ行列の疎性を利用して近似行列を更新すれば、記憶容量を大幅に減らすことができる．ヘッセ行列の疎性を利用した近似行列の更新規則として、行列補完を用いた準ニュートン法 (Matrix Completion Quasi-Newton method, 以下 MCQN 法) が提案されている [15]．MCQN 法は、コーダルグラフと半正定値行列補完に関する理論を巧みに用いて、ヘッセ行列の疎性を利用する方法である．MCQN 法では、まず従来の準ニュートン法における近似行列の更新公式 (BFGS 公式、DFP 公式など) を用いて疎構造に対応した部分行列の更新を行う．次に、その部分行列の (行列式を最大にする) 正定値行列補完を求め、その行列を近似行列とする．疎構造がコーダルグラフに関連した性質を持つとき、部分行列の (行列式を最大にする) 正定値行列補完は疎な三角行列の積として表せることが知られている．

一般にヘッセ行列の疎構造に対応するグラフ (以下、疎構造グラフ) は必ずしもコーダルグラフとは限らない．そこで、[15] では、疎構造グラフのなるべく枝数の少ない「コーダル拡張」¹ を用いて行列の更新を行うことを提案している．この方法を、本報告書ではコーダル拡張を用いた MCQN 法 (Ext-MCQN 法) と呼ぶことにする．Ext-MCQN 法で求めた近似ヘッセ行列は疎な三角行列の積で表すことができるので、Ext-MCQN 法は従来の準ニュートン法よりも大幅に少ない記憶容量で実装できる．したがって、Ext-MCQN 法は大規模な問題には非常に有効である．理論的な面からも Ext-MCQN 法は優れた性質を持っており、従来の DFP 公式と同様の仮定のもとで、Ext-MCQN 法によって生成される点列は超 1 次収束することが示されている．しかし、Ext-MCQN 法では、枝数のできるだけ少ないコーダル拡張を用いたとしても、問題によってはヘッセ行列の非ゼロ要素数と比べて近似行列の非ゼロ要素数が大幅に増えてしまうことがある．

そこで、本報告書では、疎構造グラフのコーダル部分グラフを用いる手法 (Del-MCQN 法) を提案する．この場合、近似行列の非ゼロ要素数をヘッセ行列の非ゼロ要素数以下に抑えることができるので、Del-MCQN 法は Ext-MCQN 法よりもさらに少ない記憶容量で実装することができる．

一方、Del-MCQN 法ではヘッセ行列の情報を省いてしまうため、MCQN 法がもつ高速な収束性が失われてしまう可能性がある．そのため、コーダル部分グラフを求める際、疎構造グラフから枝をできるだけ削らないようにすることが望ましい．そのような問題は、疎構造グラフの枝数最大のコーダル部分グラフを求める問題 (以後、最大コーダル部分グラフ問題と呼ぶ) として定式化できる．最大コーダル部分グラフ問題は NP 完全であるため [8]、それを近似的に解く実用的なヒューリスティックアルゴリズムがいくつか提案されている [2, 13]．本報告書では、Xue のアルゴリズム [13] によって得られるコーダル部分グラフを用いて Del-MCQN 法を構成する．さらに、数値実験を行い、既存の方法との比較を行った結果を報告する．

本報告書の構成は、以下のとおりである．まず、2 節において行列の疎性と関連の深いコーダルグラフの性質について簡単にまとめる．Del-MCQN 法を実装する際に必要となる、最大コーダル部分グラフ問題に

¹コーダル拡張については、2 章で説明する．

対するアルゴリズムについてもこの節で紹介する．次に，3.1,3.2 節で準ニュートン法，ならびに Ext-MCQN 法を紹介し，その問題点についてまとめる．さらに，3.3 節で，コーダル部分グラフを用いた近似行列の新たな更新規則 (Del-MCQN 法) を提案する．4 節では，3.3 節で提案した Del-MCQN 法を凸 2 次計画問題に適用した数値実験の結果を報告し，提案手法の有効性を議論する．最後に，5 節で，まとめと今後の課題について述べる．また，付録として，実験に用いたコーダルグラフに関するアルゴリズム，ならびに半正定値行列補完に関する理論のまとめを与える．

2 行列の疎性とコーダルグラフ

2.1 コーダルグラフの基本的性質

本節では，コーダルグラフに関する基本的な事柄をまとめる．本節で述べるコーダルグラフの性質は，主に [1], [4] の内容をまとめたものであるので，詳しくはそちらを参照していただきたい．特に，[1] は，コーダルグラフとクリーク木について丁寧に解説している論文である．一方，[4] は，コーダルグラフについての論文ではないが，2.1 節にコーダルグラフについての簡単なまとめが載っている．

本報告書を通して， V を頂点の集合， $E \subseteq V \times V$ を枝の集合とし， $G = (V, E)$ で無向グラフを表すものとする．グラフにはループがない，すなわち，すべての $v \in V$ に対して， $(v, v) \notin E$ と仮定する．

本報告書で用いるグラフに関する用語を，以下で定義する．

定義 2.1 (基本的な用語)

- 2 つの頂点 $u, v \in V$ は $(u, v) \in E$ のとき隣接しているという．
- $v \in V$ に隣接している頂点の集合を $\text{Adj}_G(v) = \{u \in V \mid (u, v) \in E\}$ で表す．紛れのないときは， G を省略して，単に $\text{Adj}(v)$ とかく．
- $v \in V$ に接続している枝の数を v の次数といい， $\text{deg}(v)$ で表す²．
- 相異なる 2 つの頂点がすべて隣接しているとき，グラフは完全であるという．
- 2 つのグラフ $G = (V, E)$ と $G' = (V', E')$ に対して， $V' \subseteq V$ かつ $E' \subseteq E$ が成り立つとき， G' を G の部分グラフという．
- V' をグラフ $G = (V, E)$ の頂点集合 V の部分集合とする．このとき，頂点集合を V' ，枝集合を $E' := E \cap (V' \times V')$ とするグラフ $G' = (V', E')$ を (V' による) G の誘導部分グラフという³．
- C をグラフ $G = (V, E)$ の頂点集合 V の部分集合とする．このとき， C による $G = (V, E)$ の誘導部分グラフが完全であるならば⁴，その部分グラフを G のクリークという．クリーク中の任意の 2 頂点間には枝があることから，本報告書ではそれに属する頂点集合のみを明示して，クリーク C と表す．
- ある頂点 $v \in V$ に隣接する頂点の集合 $\text{Adj}(v)$ が，グラフ $G = (V, E)$ 上でクリークを形成しているとする．このとき，その頂点 v を単体的頂点という．
- 他のクリークの真の部分グラフにならないクリークを，極大クリークという．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．
- サイクル中の連続していない 2 つの頂点を結ぶ枝を弦 (コード) という．

²すなわち， $\text{deg}(v) = |\text{Adj}_G(v)|$ である．

³ G' は G の部分グラフになっていることに注意

⁴すなわち，異なる $i, j \in C$ のすべてのペアに対して $(i, j) \in E$

定義 2.2 (コーダルグラフ) グラフ $G = (V, E)$ に含まれる長さ 4 以上のすべてのサイクルが弦をもつとき, G はコーダルグラフである, または単にコーダルであるという.

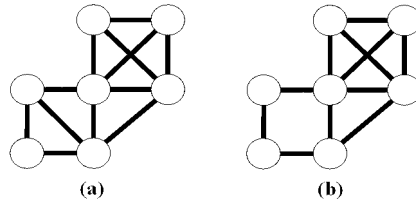


図 1: コーダルグラフの例 (a) とコーダルグラフでない例 (b)

図 1 に示したグラフのうち, 左のグラフ (a) はコーダルグラフであるが, 右のグラフ (b) は左下の長さ 4 のサイクルが弦を持たないので, コーダルグラフでない.

コーダルグラフの最も基本的な性質は, 次の 2 つである.

性質 1 コーダルグラフは, 単体的頂点をもつ.

性質 2 $G = (V, E)$ をコーダルグラフとし, $v_1 \in V$ をその単体的頂点とする. このとき, $V \setminus \{v_1\}$ により誘導される部分グラフもまたコーダルである.

性質 1, 2 より, $V \setminus \{v_1\}$ により誘導される部分グラフも単体的頂点をもつ. それを v_2 とする. これを繰り返すことで, $\text{Adj}(v_i) \cap \{v_{i+1}, v_{i+2}, \dots, v_n\}$ がすべての $i = 1, 2, \dots, n-1$ に対してクリークになるように, G の頂点に順序付け (v_1, v_2, \dots, v_n) (ただし, $n = |V|$) を行うことができる. この順序を完全消去順序(perfect elimination ordering, 以後 PEO と表す) と呼ぶ. PEO の存在は, 次のようにコーダル性を特徴づけている.

性質 3 グラフ $G = (V, E)$ がコーダルであるための必要十分条件は, G が PEO をもつことである.

注意 1 PEO は唯一ではない. 例えば, 図 2 のコーダルグラフにおいて, $(1, 2, 3, 4, 5, 6, 7)$ は PEO であるが, $(3, 2, 4, 6, 7, 1, 5)$ もまた PEO である.

コーダルグラフの極大クリークは, PEO を用いて次のように簡単に列挙することができる. $G = (V, E)$ をコーダルグラフとし, (v_1, v_2, \dots, v_n) を G の PEO とする. v_1 は単体的頂点なので, v_1 を含む極大クリークは唯一であり, $\{v_1\} \cup \text{Adj}(v_1)$ で与えられる. そして, v_1 を含まない極大クリークは, $\{v_2, v_3, \dots, v_n\}$ から導かれた部分グラフの極大クリークである. したがって, コーダルグラフ $G = (V, E)$ の極大クリーク族は, $\{v_i\} \cup (\text{Adj}(v_i) \cap \{v_{i+1}, v_{i+2}, \dots, v_n\}), i = 1, 2, \dots, n$ の中で極大なものの集合として与えられる. よって, l を G の極大クリークの総数とすると, 次の性質が成り立つ.

性質 4 $G = (V, E)$ をコーダルグラフとし, (v_1, \dots, v_n) を G の PEO とする. このとき, G の極大クリーク族 $\{C_r \mid r = 1, 2, \dots, l\}$ は, 次のように構成することができる:

$$C_r = \{v_i\} \cup (\text{Adj}(v_i) \cap \{v_{i+1}, v_{i+2}, \dots, v_n\}), \quad i = \min_{v_j \in C_r} j$$

性質 4 から, 極大クリークの数 l の上界は n であることがわかる. 図 2 に, コーダルグラフとその極大クリークの例を示す. $(1, 2, 3, 4, 5, 6, 7)$ がこのグラフの PEO である. この例では, 4 つの極大クリークが存在し, $C_1 = \{1, 5, 7\}, C_2 = \{2, 3, 4, 6\}, C_3 = \{4, 6, 7\}, C_4 = \{5, 6, 7\}$ である.

さらに, コーダルグラフの極大クリーク族は, 次の条件を満たすように添え字をつけることができることが知られている.

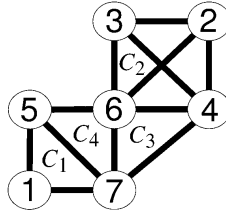


図 2: コーダルグラフとその極大クリークの例

性質 5 $G = (V, E)$ がコーダルグラフであるとき, $r = 1, 2, \dots, l-1$ に対して,

$$\exists s \geq r+1 : C_r \cap (C_{r+1} \cup C_{r+2} \cup \dots \cup C_l) \subseteq C_s \quad (2.1)$$

となる極大クリーク族 $\{C_r | r = 1, 2, \dots, l\}$ が存在する.

性質 5 は **Running Intersection Property** (以降 **RIP** と表す) と呼ばれる. 図 2 に示した極大クリーク族 $\{C_1, C_2, C_3, C_4\}$ は, RIP を満たしている.

2.2 行列の疎性とコーダルグラフ

本節では, 行列の疎性とコーダルグラフの関係について述べる.

まず, $n \times n$ 対称行列 A に対し, 集合 E を $E := \{(i, j) | A_{ij} \neq 0\}$ と定義し, これを本報告書では行列 A の疎構造と呼ぶことにする. 次に, 集合 V, \bar{E} を, $V := \{1, 2, \dots, n\}, \bar{E} := E \setminus \{(i, i) | i = 1, \dots, n\}$ で定め, 頂点集合を V , 枝集合を \bar{E} とする無向グラフ $G = (V, \bar{E})$ を考える⁵. 本報告書では $G = (V, \bar{E})$ を行列 A の疎構造グラフと呼ぶことにする⁶.

例 1 行列 A の非ゼロ要素が図 3(a) のように与えられているとする. ただし, 図中の * は $A_{ij} \neq 0$, 他は $A_{ij} = 0$ を表す. 行列 A の疎構造グラフは図 3(b) となる.

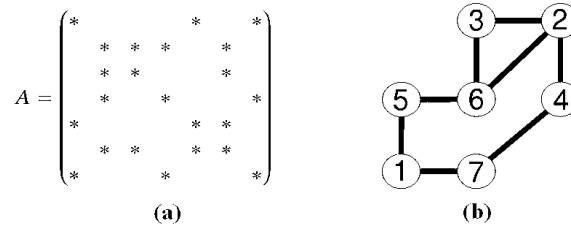


図 3: 行列 A とその疎構造グラフの例

図 3(b) のように, グラフ $G = (V, \bar{E})$ は必ずしもコーダルでないことに注意する. そこで, 疎構造を表すグラフにコーダル性をもたせることを考える.

定義 2.3 一般の (必ずしもコーダルでない) グラフ $G = (V, E)$ が与えられたとする.

- グラフ $G' = (V, F)$ が G のコーダル拡張 (chordal extension) であるとは, G' がコーダルグラフであり, かつ $F \supseteq E$ が成り立つときをいう.

⁵したがって, $G = (V, \bar{E})$ はループを持たない.

⁶行列 A の非対角成分において非ゼロな箇所は, グラフ $G = (V, \bar{E})$ の隣接行列の非対角成分において非ゼロな箇所と一致することに注意しておく.

- グラフ $G' = (V, F)$ が G のコーダル縮小 (chordal deletion) であるとは, G' がコーダルグラフであり, かつ $F \subseteq E$ が成り立つときをいう.

すなわち, 一般のグラフ $G = (V, E)$ にいくつか枝を付け加えてコーダルグラフにしたものを (G の) コーダル拡張, 逆にグラフ G からいくつか枝を削ってコーダルグラフにしたものを (G の) コーダル縮小と呼ぶ. 今後, 文脈から紛れのないときは, 「 G の」を省略し, 単に「コーダル拡張」, 「コーダル縮小」と呼ぶ. なお, グラフ G の頂点集合と, そのコーダル拡張, コーダル縮小の頂点集合は同じであることに注意する. また, コーダル拡張, コーダル縮小は一意ではない.

例 2 図 3(b) で与えられるグラフ G のコーダル拡張, コーダル縮小の例をそれぞれ図 4(a),(b) に示す. (a) は G に (3, 4), (5, 7), (6, 4), (6, 7) 間の枝を付け加えたグラフ, (b) は G から (1, 7) 間の枝を削除したグラフである.

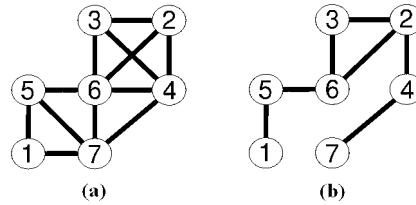


図 4: 図 3(b) で与えられるグラフ $G = (V, E)$ のコーダル拡張 (a), コーダル縮小 (b) の例

コーダル拡張は, 疎行列のコレスキー分解と密接に関連することが知られている [11]. 行列 A を疎な正定値対称行列とし, $A = LL^T$ (ただし, L は下三角行列) をそのコレスキー分解とする. このとき, $A_{ij} = 0$ であっても, コレスキー分解を行うと $L_{ij} \neq 0$ となり, 非ゼロ要素が増加してしまうことがある. このようにコレスキー分解前の行列で 0 であったところにコレスキー分解後 0 でない値が入ることを充填 (fill-in) という. 行列 L の疎構造を $G' = (V, \bar{F})$ (ただし, $V := \{1, 2, \dots, n\}$, $F := \{(i, j) \mid L_{ij} \neq 0 \text{ or } L_{ji} \neq 0\}$, $\bar{F} := F \setminus \{(i, i) \mid i = 1, 2, \dots, n\}$) で表現すると (コレスキー分解の計算過程で数値的な打ち消しが起こらないと仮定すれば) グラフ G' は, 行列 A の疎構造グラフ $G = (V, E)$ のコーダル拡張になることが知られている [4].

充填が起こると, 行列 L の非ゼロ要素が増加する. 非ゼロ要素の増加はそのまま計算量の増加につながり, 行列 A が疎であってもその性質を有効に利用することができない. そこで, 行列 A に前処理を行って, なるべく充填の数を少なくすることが求められる⁷. 上で述べたことから, 行列 A のコレスキー分解における充填の最小化問題は, 行列 A の疎構造を表現するグラフの枝数最小のコーダル拡張を求める問題と等価である.

一方, コーダル縮小は, 一般のグラフに含まれる最大のクリークを見つける問題 (最大クリーク問題) や疎な線形方程式を解くアルゴリズムへの応用に関連して研究されている [2, 13]. 一般のグラフに対する最大クリーク問題は NP 困難であるが, 与えられたグラフの枝数最大のコーダル縮小を用いることで, 最大クリーク問題に対する比較的よい解を見つけられることが知られている.

しかし, 与えられたグラフの枝数最小のコーダル拡張を求める問題は NP 完全であることが知られている [8]. 同様に, 与えられたグラフの枝数最大のコーダル縮小を求める問題も NP 完全である [8]. そのため, これらの問題に対する発見的手法がいくつか提案されている. その方法を次節以降で説明する.

2.3 コーダル拡張を求めるアルゴリズム

枝数最小のコーダル拡張を求めるヒューリスティクスとしては, 最小次数法 (最小次数順序法, Minimum degree ordering algorithm) を用いた手法が有名である. 最小次数法は, コレスキー分解における充填の発生

⁷ 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) の順序に依存することが知られている. したがって, 充填の数が少なくなるようにあらかじめ行列 A の行 (列) を並べ替えておく.

を観察することにより考案されたグラフの頂点のオーダリング手法である。その基本的な考え方は、「隣接頂点の少ない順に頂点に番号を振る」ということである。具体的には、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。そして、頂点がすべて除去されるまで同様の操作を繰り返すというアルゴリズムである⁸。その手続きを正確に述べると次のようになる。

最小次数法

Step 1: $V^0 := V, E^0 := E, S := \emptyset, k := 1$ とする。

Step 2: $G^{k-1} := (V^{k-1}, E^{k-1})$ とし、グラフ G^{k-1} において次数が最小の頂点を v_k とする。 $S := (S, v_k)$ とする。

Step 3: G^{k-1} から頂点 v_k を削除し、新しいグラフ $G^k := (V^k, E^k)$ をつくる。

Step 4: $k = k + 1$ とし、 $k > |V|$ 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 $S = (v_1, \dots, v_n)$ 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。

最小次数順序 S を PEO とするコーダル拡張を求めるには、次の Parter が提案した手法 [10] を用いればよい。ここで、最小次数順序を $S = (v_1, v_2, \dots, v_n)$ とする。

順番 S から $G = (V, E)$ のコーダル拡張を求めるアルゴリズム

Step 1: $k := 1$ とする。 $G' = G$ とし、 $E^* = E$ とする。

Step 2: $k > n$ ならば終了。そうでなければ頂点 v_k とそれに接続する枝をグラフ G' から削除する。

Step 3: $\text{Adj}_{G'}(v_i)$ が完全グラフとなるように枝を E^* に追加する。

Step 4: $k := k + 1$ とし、ステップ 2 へ。

このアルゴリズムにより得られるグラフ $G^* = (V, E^*)$ は、 S を PEO にもつグラフ G のコーダル拡張となる。

2.4 コーダル縮小を求めるアルゴリズム

一般のグラフに含まれる枝数最大のコーダル部分グラフを見つける問題 (最大コーダル部分グラフ問題) は NP 完全であり、さまざまな発見的解法が提案されている⁹。その基本的な考え方を説明する。

いま、グラフ $G = (V, E)$ が与えられ、 G の頂点には順番 $\{v_1, v_2, \dots, v_n\}$ が振られているとする。 $V^k := \{v_j \mid j \geq k\}$ とし、頂点集合を V^k 、枝集合を $E^k \subseteq V^k \times V^k$ とするグラフ $G^k = (V^k, E^k)$ を考える。グラフ G のコーダル部分グラフを求める基本的アイデアは、次のとおりである。

まず、1 頂点のみからなるグラフ $G^n := (\{v_n\}, \emptyset)$ はコーダルグラフである。次に、 $G^k = (V^k, E^k)$ が G のコーダル部分グラフであるとしよう。このとき、 $\{v_{k-1}, v_k, v_{k+1}, \dots, v_n\}$ を頂点とするコーダル部分グラフ $G^{k-1} = (V^{k-1}, E^{k-1})$ を次のように構成することができる。

Step 0: $G^{k-1} := G^k$

Step 1: G^{k-1} の頂点集合 V^{k-1} に頂点 v_{k-1} を追加する。

Step 2: v_{k-1} が単体的頂点になるように、次の条件を満たす G^k のクリーク $C \subseteq V^k$ の各頂点と頂点 v_{k-1} とを結ぶ枝を E^{k-1} に付け加える。

⁸グラフにおいて次数の少ない頂点は、対応する行列において非ゼロ要素の少ない列に対応する。

⁹与えられたグラフの枝数最大のコーダル縮小を求める問題は、与えられたグラフに含まれる枝数最大のコーダル部分グラフを求める問題と等価であることに注意。

【条件】

$V^k(v_{k-1}) := \text{Adj}_G(v_{k-1}) \cap V^k$ とする¹⁰ . C は, $V^k(v_{k-1})$ で誘導される G^k の部分グラフに含まれるクリークである .

こうして得られたグラフ G^{k-1} は, G のコーダルな部分グラフである . これを $k = n, n-1, n-2, \dots, 1$ とくり返して得られるグラフ $G^1 = (V^1, E^1)$ は, $V^1 = V$ を満たすグラフ G の部分グラフになる . グラフ G^1 は (v_1, v_2, \dots, v_n) を PEO にもつので, 定理 3 から G^1 はコーダルである . すなわち, G^1 は G のコーダル縮小である .

枝数のできるだけ多いコーダル縮小を求めるため, 頂点の選び方にさまざまな工夫がなされている . ここでは, Xue のアルゴリズム [13] を紹介する . このアルゴリズムは, 一般のグラフに対する最大クリーク問題への応用を念頭に考えられたアルゴリズムである . 表記の簡単のため, $\text{Suc}_G(v_i) := \{v_j \mid j > i \text{ and } v_j \in \text{Adj}_G(v_i)\}$ とする .

Edge-maximal chordal subgraph (Xue [13])

入力: グラフ $G = (V, E)$

出力: グラフ G のコーダル部分グラフ (コーダル縮小) $G' = (V, E')$

Step 0: $k := n$ とし, $V^k := \emptyset, E^k := \emptyset, peo := \emptyset, U := V$ とする . また, $\forall v \in V$ に対し, $t(v) = \emptyset, s(v) = 0$ とする .

Step 1: $k = 1$ なら終了 . $G^k := (V^k, E^k)$ は $peo = (v_1, \dots, v_n)$ を PEO にもつ枝数最大のコーダル部分グラフである .

Step 2: $s(v) = \max\{s(u) \mid u \in U\}$ を満たす $v \in U$ を 1 つ選ぶ . $V^{k-1} := V^k \cup \{v\}, peo := (v, peo), U := U \setminus \{v\}$ とする . $E^{k-1} := E^k \cup \{(v, u) \mid u = t(v) \text{ or } u \in \text{Adj}_G(v) \cap \text{Suc}_{G^k}(t(v))\}$ とする . 頂点 v のラベルを v_k とする .

Step 3: $\forall u \in \text{Adj}_G(v) \cap U$ に対し, $r_u := 1 + |\text{Suc}_{G^{k-1}}(v) \cap \text{Adj}_G(u)|$ とする .
もし, $r_u \geq s(u)$ なら, $t(u) := v, s(u) := r_u$ とする . $k := k-1$ とし, Step1 へ戻る .

注意 2 Step2 において $\max\{s(u) \mid u \in U\}$ を満たす頂点が複数あるとき, Xue はグラフ G において次数が最大となるものを選ぶことを推奨している [13] .

Xue のアルゴリズムで生成される G のコーダル縮小については, 次の性質が知られている .

定理 2.1 ([13], Theorem 3.1) $G = (V, E)$ に対して Xue のアルゴリズムを適用して得られたコーダル縮小を $G' = (V, E')$ とする . G' は $peo = (v_1, \dots, v_k)$ を PEO にもつ G のコーダル部分グラフの中で枝数最大のものである .

なお, Xue のアルゴリズムの計算量は, $O(|V| + \sum \deg(v) + \sum \deg^2(v)) \sim O(\Delta \times |E|)$ である [13] (ただし, $\Delta = \max\{\deg(v) \mid v \in V\}$ である .)

¹⁰すなわち, グラフ $G^k = (V^k, E^k)$ の頂点のうち, 与えられたグラフ $G = (V, E)$ において v_{k-1} と隣接した頂点の集合を $V^k(v_{k-1})$ とする .

3 コーダル部分グラフを用いた準ニュートン法

3.1 準ニュートン法とその問題点

この節では、準ニュートン法とその問題点について述べる。

準ニュートン法は、目的関数のヘッセ行列（もしくはその逆行列）の近似行列を用いて点列を生成する反復法である。\$x_k\$ を現在の反復点とし、\$B_k\$ を目的関数のヘッセ行列 \$\nabla^2 f(x_k)\$ の近似行列とする。ここで、\$B_k\$ は正定値対称行列であるとしよう。準ニュートン法では、まず探索方向 \$p_k\$ を

$$p_k = -B_k^{-1} \nabla f(x_k)$$

で定める¹¹。次に、\$p_k\$ 方向に直線探索を行って、次の反復点 \$x_{k+1}\$ を

$$x_{k+1} = x_k + \alpha_k p_k$$

により定める。ただし、\$\alpha_k\$ は非負の実数（ステップ幅）である。

行列 \$B_k\$ は各反復において適宜更新する。ここでは、\$B_k\$ の逆行列 \$H_k = B_k^{-1}\$ を更新することを考える¹²。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。ただし、

$$s_k = x_{k+1} - x_k, \quad y_k = \nabla f(x_{k+1}) - \nabla f(x_k) \quad (3.1)$$

である。一般に、セカント条件を満たす行列は無数に存在するので、いろいろな種類の更新公式が考えられており、その中でも BFGS 法と DFP 法が代表的である。

DFP 法では、次の最適化問題の解を \$H_{k+1}\$ とする [3]。

$$\begin{aligned} \min_H \quad & \psi(H_k^{-\frac{1}{2}} H H_k^{-\frac{1}{2}}) \\ \text{subject to} \quad & H y_k = s_k \\ & H = H^T \\ & H \geq 0 \end{aligned} \quad (3.2)$$

ただし、\$\psi: \mathbb{R}^{n \times n} \rightarrow \mathbb{R}\$ は

$$\psi(A) = \text{trace}(A) - \ln \det(A)$$

で定義される狭義凸関数である。また、\$H \geq 0\$ (\$H > 0\$) は、\$H\$ が半正定値（正定値）であることを表す。

特に、\$H\$ が正定値行列で、条件：

$$s_k^T y_k > 0 \quad (3.3)$$

を満たすとき¹³、問題 (3.2) の解は一意に定まり、その解 \$H_{k+1}^{DFP}\$ は次式で計算できる [3]。

$$H_{k+1}^{DFP} = H_k - \frac{H_k y_k (H_k y_k)^T}{y_k^T H_k y_k} + \frac{s_k s_k^T}{s_k^T y_k} \quad (3.4)$$

この更新公式は、DFP 公式と呼ばれる。

また、BFGS 法は、次式で与えられる更新公式（BFGS 公式）を用いて近似行列の更新を行う方法である。

$$H_{k+1}^{BFGS} = H_k - \frac{H_k y_k s_k^T + s_k (H_k y_k)^T}{s_k^T y_k} + \left(1 + \frac{y_k^T H_k y_k}{s_k^T y_k} \right) \frac{s_k s_k^T}{s_k^T y_k} \quad (3.5)$$

DFP 公式や BFGS 公式で求まる \$H_{k+1}\$ は、\$s_k (s_k)^T\$ などの影響で、\$\nabla^2 f(x)\$ が疎であっても一般には密な行列になる。したがって、行列 \$H_{k+1}\$ の保持には \$O(n^2)\$ のメモリが必要になる。ゆえに、次元数 \$n\$ が大きい問題に対してはこれらの方法を使うことができない。

¹¹探索方向 \$p_k\$ は、必ず目的関数の降下方向になる。なぜなら、\$B_k\$ の正定値性より \$\nabla f(x_k)^T p_k = -\nabla f(x_k) B_k^{-1} \nabla f(x_k) < 0\$ が成り立つからである。

¹²このとき、探索方向 \$p_k\$ は \$p_k = -H_k \nabla f(x_k)\$ と行列とベクトルの積で計算できる。

¹³\$f\$ が強凸関数のとき、条件 (3.3) は任意の 2 点 \$x_k, x_{k+1}\$ に対して成り立つ。また、\$f\$ が非凸であるとき、ステップ幅 \$\alpha_k\$ が Wolfe の条件を満たすようにすれば、条件 (3.3) は満たされることが知られている [9]。

3.2 コーダル拡張を用いた MCQN 法とその計算量

3.2.1 Ext-MCQN 法の概要

次元数 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。そこで、 H_{k+1}^{DFP} を与える問題 (3.2) の制約条件に対し、疎性の条件を加えた次の問題を考える。

$$\begin{aligned}
 \min_H \quad & \psi(H_k^{-\frac{1}{2}} H H_k^{-\frac{1}{2}}) \\
 \text{subject to} \quad & H y_k = s_k \\
 & H = H^T \\
 & H \geq 0 \\
 & (H^{-1})_{ij} = 0 \quad \forall (i, j) \notin F
 \end{aligned} \tag{3.6}$$

ここで、集合 F は、 $F \subseteq V \times V$ かつ $F \approx E := \{(i, j) \mid \text{ある } x \in \mathbb{R}^n \text{ に対して } [\nabla^2 f(x)]_{ij} \neq 0\}$ を満たすように選ぶ。ただし、 $V = \{1, 2, \dots, n\}$ である。本報告書を通して、(1) $(i, i) \in F, i = 1, 2, \dots, n$ (2) $(i, j) \in F \Rightarrow (j, i) \in F$ と仮定する。集合 F はなるべく $F = E$ となるように選ぶのが望ましい。

残念なことに、問題 (3.2) のように問題 (3.6) の最適解を陽に書き下すことができないので、問題 (3.6) の近似解を H_{k+1} としようというのが、MCQN 法の基本的な考え方である [15]。

(MCQN 法のプロトタイプ)

Step 1: 既存の準ニュートン法の更新公式 (DFP 公式 (3.4) や、BFGS 公式 (3.5) など) を用いて、 H_k から $(\bar{H}_{k+1})_{ij}, \forall (i, j) \in F$ の成分を計算しておく。

Step 2: $(\bar{H}_{k+1})_{ij}, (i, j) \in F$ を用いた最適化問題：

$$\begin{aligned}
 \min_H \quad & \psi(H_k^{-\frac{1}{2}} H H_k^{-\frac{1}{2}}) \\
 \text{subject to} \quad & H = H^T \\
 & H_{ij} = (\bar{H}_{k+1})_{ij} \quad \forall (i, j) \in F \\
 & H \geq 0 \\
 & (H^{-1})_{ij} = 0 \quad \forall (i, j) \notin F
 \end{aligned} \tag{3.7}$$

の最適解を H_{k+1} とする。

問題 (3.7) は、問題 (3.6) の制約条件に含まれるセカント条件 $H y_k = s_k$ を、 $H_{ij} = (\bar{H}_{k+1})_{ij}, \forall (i, j) \in F$ で置き換えた問題である。問題 (3.7) の解は、問題 (3.6) の近似解とみなすことができる ($F = V \times V$ のときは、問題 (3.7) は問題 (3.2) と等価である。) なお、今後 \bar{H}_{k+1} を、単に \bar{H} と書くことにする。

問題 (3.7) は、半正定値行列補完問題、すなわち、 $H_{ij} = \bar{H}_{ij}, \forall (i, j) \in F$ を満たし、 $(i, j) \notin F$ の部分の要素をすべて補ってできる半正定値対称行列 H を見つける問題である (半正定値行列補完については、付録 B を参照。) 問題 (3.7) も一般には簡単に解くことはできないが、 F が次の条件を満たすとき、その解を陽に表せることが知られている。

【コーダル条件】

頂点集合を V 、枝集合を $\bar{F} := F \setminus \{(i, i) \mid i = 1, \dots, n\}$ とする無向グラフ $G' = (V, \bar{F})$ がコーダルグラフとなる¹⁴。

問題の疎性を利用するためには $F = E$ とすることが望ましいのだが、ヘッセ行列の疎構造グラフ $G = (V, \bar{E})$ は一般にコーダルグラフとは限らない。そこで、[15] では、集合 F を次の条件を満たすように選ぶことが

¹⁴グラフ G' にはループはないことに注意。

提案されている．

$$\begin{aligned} \text{条件 1 } & F \supseteq E. \\ \text{条件 2 } & \text{集合 } F \text{ は, コーダル条件を満たす.} \end{aligned} \quad (3.8)$$

(3.8) の 2 条件を満たす F を求めることは, グラフ $G = (V, \bar{E})$ のコーダル拡張 $G' = (V, \bar{F})$ を求めることと等価である．本報告書では, グラフ G のコーダル拡張 G' を用いて構成した MCQN 法を, Ext-MCQN 法と呼ぶことにする．また, 今後単に「疎構造グラフ」と言ったときはヘッセ行列の疎構造グラフを指すことにする．疎性を利用するためには, なるべく $F \setminus E$ の要素が少なくなるように F を選ぶことが望ましい．言い換えると, 疎構造グラフ G の枝数最小のコーダル拡張を G' として用いることが望ましい．そのためには, 2.3 節で述べた方法などを用いてコーダル拡張 G' を求めればよい．

さて, コーダル条件を満たす集合 F が与えられたときの問題 (3.7) の解を具体的に与えよう．まず, 補題を 1 つ準備する．

補題 3.1 ([15], Theorem 2) $s_k^T y_k > 0$ と仮定する． H_k は正定値対称行列で, $(H_k^{-1})_{ij} = 0, \forall (i, j) \notin F$ と仮定する．このとき, 問題 (3.7) は次の最適化問題と等価になる：

$$\begin{aligned} \text{maximize } & \det(H) \\ \text{subject to } & H_{ij} = \bar{H}_{ij} \quad \forall (i, j) \in F \\ & H = H^T \\ & H \geq 0 \end{aligned} \quad (3.9)$$

さらに, この解は唯一つであり, 解として得られた行列 H は, $(H^{-1})_{ij} = 0, \forall (i, j) \notin F$ を満たす．

注意 3 問題 (3.9) は, $\bar{H}_{ij}, \forall (i, j) \in F$ が要素として与えられた部分行列の半正定値補完行列の中で, 行列式が最大となるものを求める問題である．

F がコーダル条件を満たすとき, 問題 (3.9) の解が陽に計算できる疎行列の積で表されることを示す． $G' = (V, \bar{F})$ がコーダルグラフであるとき, 2.1 節で述べたように, 次の 2 条件を満たす極大クリーク族 $\{C_r | r = 1, 2, \dots, l\}$ が存在する．

- $F = \bigcup_{r=1}^l C_r \times C_r$
- RIP (式 (2.1)): $r = 1, 2, \dots, l-1$ に対して

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

を満たす．

この極大クリーク族を用いて, $\{C_r\}$ の部分集合の族 $\{S_r | r = 1, 2, \dots, l\}, \{U_r | r = 1, 2, \dots, l\}$ を以下で定義する．

$$S_r := C_r \setminus (C_{r+1} \cup C_{r+2} \cup \dots \cup C_l), r = 1, \dots, l \quad (3.10)$$

$$U_r := C_r \cap (C_{r+1} \cup C_{r+2} \cup \dots \cup C_l), r = 1, \dots, l \quad (3.11)$$

ここで, $V = \bigcup_{r=1}^l S_r$ かつ $S_i \cap S_j = \emptyset$ であることに注意しておく． S_1, S_2, \dots, S_l の順にそれぞれの要素を取り出して並べたときに $1, 2, \dots, n$ となるように G' の頂点の番号をつけかえる¹⁵．そのような順序の付け替えを行う置換行列を P とする．

¹⁵定義により, S_1 の各要素は非空であり, S_1 に属する各頂点は G' の単体的頂点である．さらに, S_i の各要素は, $S_i \cup S_{i+1} \cup \dots \cup S_l$ により導かれる G' の部分グラフの単体的頂点である．したがって, S_1, S_2, \dots, S_l の順にそれぞれの要素を取り出して並べた順序は G' の PEO になる．こうして求めた PEO が $(1, 2, \dots, n)$ となるように頂点の番号を付け替えることになる．

このとき，問題 (3.9) の解 H^* は

$$H^* = P^T [L_1^{(k+1)}]^T [L_2^{(k+1)}]^T \cdots [L_{l-1}^{(k+1)}]^T D^{(k+1)} L_{i-1}^{(k+1)} \cdots L_2^{(k+1)} L_1^{(k+1)} P \quad (3.12)$$

で与えられる [4] . ここで , $r = 1, 2, \dots, l-1$ に対して

$$[L_r^{(k+1)}]_{ij} = \begin{cases} 1 & i = j \\ \left[(\bar{H}_{U_r, U_r})^{-1} \bar{H}_{U_r, S_r} \right]_{ij} & (i, j) \in U_r \times S_r \\ 0 & \text{otherwise} \end{cases} \quad (3.13)$$

であり ,

$$D^{(k+1)} = \begin{pmatrix} D_{S_1 S_1}^{(k+1)} & & & \\ & D_{S_2 S_2}^{(k+1)} & & \\ & & \ddots & \\ & & & D_{S_l S_l}^{(k+1)} \end{pmatrix} \quad (3.14)$$

である . ただし ,

$$D_{S_r S_r}^{(k+1)} = \begin{cases} \bar{H}_{S_r S_r} - \bar{H}_{S_r U_r} (\bar{H}_{U_r U_r})^{-1} \bar{H}_{U_r S_r} & r \leq l-1 \\ \bar{H}_{S_r S_r} & r = l \end{cases} \quad (3.15)$$

である¹⁶ .

補題 3.1 より , 式 (3.12) で得られる H^* は問題 (3.7) の最適解である . MCQN 法では , 式 (3.12) で得られる H^* を , 次回の近似行列 H_{k+1} とする . H_{k+1} は , $\bar{H}_{ij}, \forall (i, j) \in F$ のみを用いて計算されることに注意する . $L_r^{(k+1)}$ は三角行列であり , $D^{(k+1)}$ は l 個の行列が対角に並んだブロック対角行列である . F の要素が少ないとき , U_r, S_r は一般に小さくなるので¹⁷ , $L_r^{(k+1)}, D^{(k+1)}$ は疎な行列となる .

MCQN 法の実装の際 , H_k を陽に計算する必要はなく , $(H_k)_{ij}, \forall (i, j) \in F$ (と $L_r^{(k)}, D^{(k)}$) を記憶し , 各反復で更新すればよい (詳しくは , 3.3.1 節を参照されたい .) したがって , MCQN 法は従来の準ニュートン法よりも少ない計算量で実装することができる . さらに , F を条件 (3.8) を満たすように選んだとき , DFP 公式と同様の仮定の下で , Ext-MCQN 法によって生成される点列は局所的に超一次収束することが示されている [15] . しかし , 条件 (3.8) を満たす集合 F を用いた場合 , 問題によっては集合 E と比べて集合 F のサイズがかなり大きくなってしまふことがある . これは , 既存の更新公式 (BFGS 公式など) で更新すべき近似行列の要素数が大幅に増えてしまうことを意味する . ヘッセ行列の疎性を利用するという観点からは , このことは望ましくない .

3.2.2 Ext-MCQN 法の計算量

Ext-MCQN 法の行列更新に必要な計算量は ,

【 $(H_k)_{ij}, \forall (i, j) \in F$ のみをメモリに蓄えたとき】

- 領域計算量は $O(|F|)$
- 反復 1 回あたりの時間計算量は $O(\sum_{r=1}^l |C_r|^3)$

【 $(H_k)_{ij}, \forall (i, j) \in F$ および $((H_k)_{U_r, U_r})^{-1}, r = 1, 2, \dots, l-1$ をメモリに蓄えたとき】

- 領域計算量は $O(|F| + \sum_{r=1}^l |U_r|^2) \leq O(\sum_{r=1}^l |C_r|^2)$
- 反復 1 回あたりの時間計算量は $O(\sum_{r=1}^l |C_r|^2)$

¹⁶ただし , $n \times n$ 行列 A と $S, U \subseteq V$ に対し , A_{SU} は $A_{ij}, (i, j) \in S \times U$ を要素とする $|S| \times |U|$ 行列 (A の小行列) を表す .

¹⁷式 (3.10), (3.11) より , $S_r \cup U_r = C_r, r = 1, 2, \dots, l$ が成り立ち , さらに $F = \bigcup_{r=1}^l C_r \times C_r$ である .

である．詳細は 3.3.2 節で与える．Ext-MCQN 法の反復に必要な他の計算の計算量は，すべて上記の計算量よりも小さくなる．

なお，疎構造グラフ G のコーダル拡張 G' を求める操作の時間計算量は，2.3 節で述べた方法を用いた場合，最小次数順序の計算に $O(|V| + |\bar{F}|)$ ，コーダル拡張の計算に最悪の場合 $O(\sum \deg^2(v))$ がかかる．ヘッセ行列が疎な場合， $\deg(v)$ は小さな数となるので，この時間計算量は Ext-MCQN 法の行列更新にかかる時間計算量よりも少ない．

3.3 コーダル縮小を用いた MCQN 法とその計算量

3.3.1 Del-MCQN 法の概要

前節で紹介した Ext-MCQN 法では，目的関数のヘッセ行列の疎構造グラフ $G = (V, \bar{E})$ を考え，そのコーダル拡張 $G' = (V, \bar{F})$ を用いて近似行列の更新を行っていた．しかし，問題によっては集合 E と比べて集合 F のサイズがかなり大きくなってしまいう問題を抱えていた．そこで，本節では，疎構造グラフ $G = (V, \bar{E})$ のコーダル縮小を用いた MCQN 法 (Del-MCQN 法) を提案する．

まず，Ext-MCQN 法において集合 F に課していた条件 (3.8) を次のように変更する．

$$\begin{aligned} \text{条件 1} \quad & F \subseteq E. \\ \text{条件 2} \quad & \text{無向グラフ } G' = (V, \bar{F}) \text{ はコーダルグラフとなる.} \\ & (\text{ただし } \bar{F} := F \setminus \{(i, i) \mid i = 1, \dots, n\} \text{ である}) \end{aligned} \tag{3.16}$$

条件 (3.16) を満たす F を求めることは， $G = (V, \bar{E})$ のコーダル縮小を求めることと等価である．グラフ G のコーダル縮小を用いた場合，近似行列の非ゼロ要素数はヘッセ行列の非ゼロ要素数以下に抑えることができるので，Ext-MCQN 法よりもさらに少ない記憶容量で実装できる．しかし，グラフ G から枝を削ってしまうと，本来ヘッセ行列において非ゼロである成分を 0 としてしまうことになるため，近似行列 H_k はヘッセ行列 (の逆行列) を十分近似できていないおそれがある．このため，最急降下法などと同様に，条件数の悪い問題に対しては収束が極端に遅くなってしまうおそれがある．

ヘッセ行列の情報をできるだけ失わないようにするためには，なるべく $E \setminus F$ の要素を少なくすることが望ましい．そのためには，疎構造グラフ $G = (V, \bar{E})$ の枝数最大のコーダル部分グラフをコーダル縮小として用いればよいと考えられる．そこで，2.4 節で述べた Xue のアルゴリズムなどを用いてグラフ G のコーダル縮小を求め，それを用いて新しい MCQN 法を次のように構成する．

コーダル縮小を用いた MCQN 法 (Del-MCQN 法)

$V = \{1, 2, \dots, n\}$, $E = \{(i, j) \mid \text{ある } x \in \mathbb{R}^n \text{ に対して } [\nabla^2 f(x)]_{ij} \neq 0\}$, $\bar{E} = E \setminus \{(i, i) \mid i = 1, \dots, n\}$ とする .

Step 0: (コーダル縮小の計算と初期化)

- $G = (V, \bar{E})$ のコーダル縮小 $G' = (V, \bar{F})$ を求める (例えば, 2.4 節で述べた Xue のアルゴリズムを用いて求める)
- $G' = (V, \bar{F})$ の RIP を満たす極大クリーク族 $\{C_r \mid r = 1, 2, \dots, l\}$ を計算し, 式 (3.10), (3.11) を満たす C_r の部分集合の族 $\{S_r \mid r = 1, 2, \dots, l\}$ と $\{U_r \mid r = 1, 2, \dots, l\}$ を計算する .
(G' の RIP を満たす極大クリーク族を求めるアルゴリズムは, 付録 A.2 を参照)
- $F = \bar{F} \cup \{(i, i) \mid i = 1, \dots, n\}$ とする .
初期点 $x_0 \in \mathbb{R}^n$ と, $(H_0^{-1})_{ij} = 0, \forall (i, j) \notin F$ を満たす正定値対称行列 H_0 を選ぶ .
 $k := 0$ とする .

Step 1: $p_k = -H_k \nabla f(x_k)$ とする .

Step 2: (次回の反復点の決定)

- Armijo (または Wolfe) 条件を用いた直線探索によって, p_k 方向のステップ幅 α_k を求める .
- $x_{k+1} = x_k + \alpha_k p_k$ とおく .

Step 3: 停止条件が満たされていれば, x_{k+1} を解とみなして停止する . さもなければ Step 4 へいく .

Step 4: (近似行列の更新)

- 式 (3.1) により s_k と y_k を計算する .
- 既存の準ニュートン法の更新公式 (BFGS 公式, DFP 公式など) により, $\bar{H}_{ij}, \forall (i, j) \in F$ を求める .
- 式 (3.12) により, $H_{ij} = \bar{H}_{ij}, \forall (i, j) \in F$ を満たす正定値行列 H を求め, それを H_{k+1} とする .
- $k := k + 1$ として, Step1 へ戻る .

MCQN 法の実装の際には, H_k を陽に計算する必要はない . 実際, H_k が必要になるのは, Step2 で $p_k = -H_k \nabla f(x_k)$ を計算するときと, Step 4 で $\bar{H}_{ij}, \forall (i, j) \in F$ を求める際に $H_k y_k$ を計算するときだけである¹⁸ . 行列 H_k とあるベクトル $d \in \mathbb{R}^n$ の積 $H_k d$ の計算は, $(H_k)_{ij}, \forall (i, j) \in F$ を記憶しておけば, 次のように計算できる .

まず, 任意のベクトル $d \in \mathbb{R}^n$ に対し, $H_k d$ の計算は, 式 (3.12) を用いて,

$$H_k d = P^T [L_1^{(k)}]^T [L_2^{(k)}]^T \cdots [L_{l-1}^{(k)}]^T D L_{l-1}^{(k)} \cdots L_2^{(k)} L_1^{(k)} P d$$

¹⁸3.3.2 節で述べる . 例えば BFGS 公式 (3.17) を見よ .

で計算できることに注意する．これを右から順番に，

$$\begin{aligned}
q_1 &= L_1^{(k)} P q_0 \\
q_2 &= L_2^{(k)} q_1 \\
&\vdots \\
q_{l-1} &= L_{l-1}^{(k)} q_{l-2} \\
q_l &= D^{(k)} q_{l-1} \\
q_{l+1} &= [L_{l-1}^{(k)}]^T q_l \\
&\vdots \\
q_{2l-1} &= P^T [L_1^{(k)}]^T q_{2l-2}
\end{aligned}$$

と計算すれば， $H_k d = q_{2l-1}$ と求められる．

あるベクトル $w \in \mathbb{R}^n$ に対して，各 $L_r^{(k)} w$ の時間計算量は $O(|U_r| |S_r|)$ であり， $D^{(k)} w$ の時間計算量は $O(\sum_{r=1}^l |S_r|^2)$ であることから¹⁹， $H_k d$ の時間計算量は $O(\sum_{r=1}^l (|U_r| |S_r| + |S_r|^2)) = O(\sum_{r=1}^l |C_r|^2)$ である．

3.3.2 Del-MCQN 法の計算量

本小節では，Del-MCQN 法の反復 1 回あたりの計算量を評価する．

Del-MCQN 法の反復 1 回あたり，Step 2 の探索方向の時間計算量は， $O(\sum_{r=1}^l (|U_r| |S_r| + |S_r|^2)) = O(\sum_{r=1}^l |C_r|^2)$ である．また，Step 3 の時間計算量は，Step 2 と比べればほとんど無視できる．

Step 4 における Del-MCQN 法の行列更新に必要な反復 1 回あたりの時間計算量は，基本的には Ext-MCQN 法の計算量と同様に評価できる．まず， $\bar{H}_{ij}, \forall (i, j) \in F$ は従来の準ニュートン法の更新公式で求められる．BFGS 法で更新されるとすると，

$$\bar{H}_{ij} = (H_k)_{ij} + \rho s_i s_j - \frac{(H_k y_k)_i (s_k)_j + (s_k)_j (H_k y_k)_i}{s_k^T y_k} \quad \forall (i, j) \in F \quad (3.17)$$

で計算される．ただし，

$$\rho = \frac{1}{s_k^T y_k} + \frac{(y_k)^T H_k y_k}{(s_k^T y_k)^2}$$

である．よって， $H_k y_k$ が計算済みならば， $\bar{H}_{ij}, \forall (i, j) \in F$ の計算は $O(|F|)$ ができる． $H_k y_k$ の時間計算量は $O(\sum_{r=1}^l (|U_r| |S_r| + |S_r|^2))$ なので， $\bar{H}_{ij}, \forall (i, j) \in F$ の時間計算量は $O(|F| + \sum_{r=1}^l (|U_r| |S_r| + |S_r|^2)) = O(\sum_{r=1}^l |C_r|^2)$ である．

次に，式 (3.13) ~ (3.15) で与えた $L_r^{(k+1)}, r = 1, 2, \dots, l$ と $D^{(k+1)}$ の時間計算量を評価する．まず， $\bar{H}_{U_r U_r}, \bar{H}_{S_r U_r}, \bar{H}_{S_r S_r}, r = 1, 2, \dots, l$ の時間計算量はそれぞれ $O(|U_r|^2), O(|U_r| \times |S_r|), O(|S_r|^2)$ となる．また， $\bar{H}_{U_r U_r}^{-1}$ の時

¹⁹ $L_r^{(k)}, r = 1, 2, \dots, l-1$ は，

$$L_r^{(k)} = I + M_r^{(k)}, \quad M_r^{(k)} = \begin{cases} [L_r^{(k)}]_{ij} & (i, j) \in U_r \times S_r \\ 0 & \text{otherwise} \end{cases}$$

とかけるので， $L_r^{(k)} w = (I + M_r^{(k)}) w = w + M_r^{(k)} w$ となる． $M_r^{(k)}$ には非ゼロ要素が $|U_r| \times |S_r|$ 個しかないと考えると， $M_r^{(k)} w$ の計算は， $O(|U_r| |S_r|)$ ができる．ゆえに， $L_r w$ の時間計算量は $O(|U_r| |S_r|)$ である．

$D^{(k)} w$ の計算は，

$$D^{(k)} w = \begin{pmatrix} D_{S_1 S_1}^{(k)} w_{S_1} \\ D_{S_2 S_2}^{(k)} w_{S_2} \\ \vdots \\ D_{S_l S_l}^{(k)} w_{S_l} \end{pmatrix}$$

であることを考えると， $O(\sum_{r=1}^l |S_r|^2)$ ができる．ここで， w_{S_r} は $w_i, i \in S_r$ を要素とする $|S_r|$ 次元ベクトルである．

間計算量は $O(|U_r|^3)$ となる。したがって、 $D^{(k+1)}$ と $L_r^{(k+1)}$, $r = 1, 2, \dots, l$ の時間計算量は大まかに $O(\sum_{r=1}^l |C_r|^3)$ となる。なお、 $((H_k)_{U_r U_r})^{-1}$ をすべての $r = 1, 2, \dots, l$ について蓄えておくと、Sherman-Morrison の公式 [9] を使うことによって、時間計算量は $O(\sum_{r=1}^l |C_r|^2)$ に減らすことができる。

以上の評価から、Del-MCQN 法の計算量は

【 $(H_k)_{ij}, \forall (i, j) \in F$ のみをメモリに蓄えたとき】

- 領域計算量は $O(|F|)$
- 反復 1 回あたりの時間計算量は $O(\sum_{r=1}^l |C_r|^3)$

【 $(H_k)_{ij}, \forall (i, j) \in F$ および $((H_k)_{U_r U_r})^{-1}, r = 1, 2, \dots, l-1$ をメモリに蓄えたとき】

- 領域計算量は $O(|F| + \sum_{r=1}^l |U_r|^2) \leq O(\sum_{r=1}^l |C_r|^2)$
- 反復 1 回あたりの時間計算量は $O(\sum_{r=1}^l |C_r|^2)$

である。

ヘッセ行列が疎であるとき、一般に $|C_r| \ll n$ である。2.1 節の性質 4 から $l \leq n$ なので、 $\sum_{r=1}^n |C_r|^2 \ll n^2$ が成り立つ。特にヘッセ行列が三重対角行列であるときは、行列のサイズを n とすると、 $l = n$ かつ $r = 1, 2, \dots, n$ に対して $|C_r| = 2$ が成り立つ。このとき、 $(H_k)_{ij}, \forall (i, j) \in F$ 、およびすべての $r = 1, 2, \dots, l$ に対して $[(H_k)_{U_r U_r}]^{-1}$ をメモリに蓄えたとしても、領域計算量と反復 1 回あたりの時間計算量はともに $O(n)$ となる。

したがって、MCQN 法の計算量に最も効いてくるのは、疎構造グラフ G のコーダル拡張（またはコーダル縮小） G' に含まれる極大クリークの大きさ²⁰であることがわかる。コーダル縮小に含まれるクリークの大きさはコーダル拡張に含まれるクリークの大きさよりも小さいことから、反復 1 回あたりの Del-MCQN 法の計算量は時間計算量、領域計算量ともに Ext-MCQN 法よりも減らすことができる。

なお、疎構造グラフ G のコーダル縮小を求める操作は、Del-MCQN 法の前処理として行う。その時間計算量は、Xue のアルゴリズムを用いた場合、2.4 節で述べたとおり $O(|V| + \sum \deg(v) + \sum \deg^2(v)) \sim O(\Delta \times |\bar{F}|)$ である（ただし、 $\Delta = \max \{\deg(v) \mid v \in V\}$ は、グラフ $G = (V, \bar{F})$ の頂点の最大次数である。）ヘッセ行列が疎であるとき、 $|\bar{F}|$ は小さく、 $\deg(v) \ll n$ となるので、この時間計算量は Del-MCQN 法の行列更新にかかる時間計算量よりも少ない。

²⁰極大クリークを $C_r, r = 1, 2, \dots, l$ とすると、 $|C_r|$ のこと。

4 数値実験

本節では, 3.3 節で提案した Del-MCQN 法の数値実験結果を報告する. 実験は CPU が 3.4GHz の Pentium4, メモリが 3.5GB の計算機上で行い, アルゴリズムは MATLAB7.0 を用いて実装した. 本実験では, 制約なし 2 次計画問題

$$\text{minimize } f(x) = \frac{1}{2}x^T Ax + b^T x \quad (4.1)$$

を解き, 最適解が得られるまでの反復回数を比較した. 行列 A は, MATLAB の関数 `sprandsym` を用いて, 以下の式によって求めた.

$$A = \text{sprandsym}(n, nz, rc, option)$$

`sprandsym` は, 条件数がおおよそ $1/rc$ となり, 非ゼロ要素率 (= 非ゼロ要素の数 / n^2) がおおよそ nz となるような, $n \times n$ の正定値対称行列をランダムに生成する関数である. `option` は, 行列の生成方法を指定するための引数で, 今回は `option = 2` と指定した²¹. なお, ベクトル b は $[0, 1]^n$ からランダムに選んだ. すべての実験において, 初期点 x_0 は $x_0 := (100, \dots, 100)^T \in \mathbb{R}^n$ とし, 初期行列 H_0 は $H_0 := I$ (n 次単位行列) とした. ステップ幅 α_k は以下の手順で求めた.

Step 1: $\beta_k^{(0)} := 1, i := 0$ とおく.

Step 2: Armijo の条件

$$f(x_k + \beta_k^{(i)} p_k) - f(x_k) \leq 0.001 \times \beta_k^{(i)} \nabla f(x_k)^T p_k$$

を満たすならば, $\alpha_k := \beta_k^{(i)}$ とおいて終了. そうでなければ, Step3 へ.

Step 3: $\beta_k^{(i+1)} := 0.5 \times \beta_k^{(i)}, i := i + 1$ として Step2 へ.

また, $\bar{H}_{ij}, \forall (i, j) \in F$ は BFGS 公式 (3.17) を用いて求めた. H_{k+1} の正定値性を保証するため, $s_k^T y_k \leq 2.2 \times 10^{-16}$ のときは $H_{k+1} = H_k$ とした. アルゴリズムの終了条件には,

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

を用いた.

【実験 1】 Del-MCQN 法と Ext-MCQN 法の計算コストの比較

本実験では, Del-MCQN 法および Ext-MCQN 法の反復 1 回あたりの計算コストを比較する実験を行った.

G', G'' をそれぞれ G のコーダル縮小, コーダル拡張とし, G', G'' の極大クリーク族をそれぞれ $\{C'_r\}, \{C''_r\}$ とする. 3.3.2 節で述べたとおり, Del-MCQN 法および Ext-MCQN 法の反復 1 回あたりの時間計算量は, それぞれ $O(\sum_r |C'_r|^2), O(\sum_r |C''_r|^2)$ である²². Del-MCQN 法, Ext-MCQN 法の計算スキームは, 前処理の段階でそれぞれコーダル縮小, コーダル拡張を求めるところを除いては同じである. そこで, Del-MCQN 法, Ext-MCQN 法の反復 1 回あたりの計算コストをはかる指標として, 次で定義される値を導入する.

$$Del_cost = \sum_r |C'_r|^2 \quad (4.2)$$

$$Ext_cost = \sum_r |C''_r|^2 \quad (4.3)$$

本実験では, 行列 A のサイズ n を $n = 1000$ と固定した. 行列 A の非ゼロ要素率 nz を $nz=0.001 \sim 0.01$ まで 0.01 刻みで変化させ, 各 nz に対し, 関数 `sprandsym` の引数 rc を $rc = 0.01, 0.5, 0.1, 0.05, 0.01, 0.005, 0.001$ と変化させて実験を行った. 各 rc, nz に対して問題は 1 個ずつ生成し, Del_cost, Ext_cost の値と最適解を得

²¹ `option = 2` としたとき, 関数 `sprandsym` で生成される行列の条件数は正確には $1/rc$ とならない. したがって, 実験の際は生成された行列の条件数も記録するようにした.

²² $(H_k)_{ij}, \forall (i, j) \in F$, および $[(H_k)_{U_r U_r}]^{-1}, \forall r = 1, 2, \dots, l$ をメモリに蓄えたとき

るまでに必要な反復回数を求めた．なお，疎構造グラフのコーダル縮小は，Xue のアルゴリズムを用いて求めた．コーダル拡張は 2.3 節で述べた手法を用いて求めた．

結果を表 1，表 2 に示す．表 1 は， $rc = 0.01, 0.5, 0.1, 0.05, 0.01, 0.005, 0.001$ と 7 回試行したときの計算コストの平均である²³．また，表 2 は，問題 (4.1) の最適解が得られるまでの Del-MCQN 法，Ext-MCQN 法の反復回数である．

表 1: 反復 1 回あたりの計算コストの比較 (行列 A のサイズ $n = 1000$)

nz	Del_cost	Ext_cost
0.001	1108.5	1108.5
0.002	2682.2	2912.8
0.003	3595.2	39281.3
0.004	3974.7	168242.3
0.005	4179.0	398744.2
0.006	4277.2	744299.2
0.007	4436.8	951421.8
0.008	4697.2	1323809.0
0.009	4924.2	1441110.8
0.01	5114.5	1661677.5

(注) Del_cost, Ext_cost は式 (4.2) で定義した値．

まず，表 1 の結果を検討する． $nz = 0.001$ のとき，行列 A の各行には非ゼロ要素が平均 1 個しかないことから，行列 A はほぼ対角行列である．したがって，行列 A の疎構造グラフはほとんどの場合コーダルグラフとなるため，コーダル縮小とコーダル拡張はほぼ一致する．よって，表 1 において $nz = 0.001$ のとき， Del_cost と Ext_cost の値が等しくなっているのは妥当である． nz の値を大きくしていくと， Del_cost の値はゆるやかに増えていくが， Ext_cost の値は急激に増加し， $nz = 0.01$ のとき， Ext_cost の値は Del_cost の値の 320 倍ほどである．また，表 2 の結果から，Ext-MCQN 法の反復回数は条件数を大きくしてもそれほど増加していないのに対し，Del-MCQN 法の反復回数は条件数を大きくすると，大きく増加することがわかる．1 回の反復にかかるコストは Del-MCQN 法のほうがはるかに小さいため，非ゼロ要素数がそれほど少なくなく，条件数がそれほど悪くなければ，全体の計算コストは Del-MCQN 法のほうが小さいと考えられる．

【実験 2】コーダル縮小，コーダル拡張の枝数と MCQN 法の反復回数の関係

本実験では，行列 A のサイズ n を $n = 100$ と固定し，関数 sprandsym の引数 rc を $rc = 0.01$ と固定した．非ゼロ要素率 nz を $nz = 0.01 \sim 0.10$ まで 0.01 刻みで変化させて実験を行った．各 nz に対して問題を 10 個ずつ生成し，最適解を得るまでに必要な反復回数を求めた．

コーダル縮小の枝数と MCQN 法の反復回数の関係を調べるため，疎構造グラフのコーダル縮小には，次の 2 つの場合を用いた．

1. Xue のアルゴリズムを用いて求めた場合
2. 疎構造グラフから枝をすべて取り除いてしまった場合，すなわち n 個の頂点からなる枝をもたないグラフをコーダル縮小として用いた場合²⁴

また，疎構造グラフのコーダル拡張は，2.3 節で述べた手法を用いて求めた．

なお，ヘッセ行列 $\nabla^2 f(x) (= A)$ の非ゼロ要素数 $|E|$ と，その疎構造グラフ $G = (V, \bar{E})$ の枝数との間には，

$$|E| = 2 \times (\text{グラフ } G \text{ の枝数}) + n$$

²³ nz がほぼ等しければ， rc を変化させても疎構造グラフの構造は変わらない

²⁴ 近似行列の対角成分のみを既存の準ニュートン法の更新公式 (ここでは BFGS 公式) で更新することになる．

表 2: 反復回数の比較 (行列 A のサイズ $n = 1000$)

n_z	rc	条件数	反復回数		n_z	rc	条件数	反復回数	
			Del	Ext				Del	Ext
0.001	0.5	2.14	26	26	0.006	0.5	2.45	23	29
	0.1	8.62	38	38		0.1	9.02	51	38
	0.05	22.79	62	62		0.05	23.71	109	72
	0.01	140.58	157	157		0.01	85.64	215	136
	0.005	359.10	178	178		0.005	220.37	754	183
	0.001	1645.71	205	205		0.001	1091.81	4670	343
0.002	0.5	2.85	28	28	0.007	0.5	2.14	26	28
	0.1	13.69	60	59		0.1	7.86	47	40
	0.05	27.68	66	69		0.05	21.44	72	66
	0.01	164.49	159	159		0.01	110.27	191	131
	0.005	384.85	399	177		0.005	256.87	443	179
	0.001	1194.10	265	265		0.001	868.05	1443	331
0.003	0.5	2.42	48	42	0.008	0.5	2.16	30	32
	0.1	12.48	63	58		0.1	8.90	59	40
	0.05	36.13	68	64		0.05	24.38	87	67
	0.01	140.37	436	152		0.01	73.93	230	137
	0.005	304.06	350	201		0.005	368.26	575	181
	0.001	2013.00	536	310		0.001	1057.77	2557	347
0.004	0.5	2.63	31	58	0.009	0.5	2.18	35	26
	0.1	11.13	52	42		0.1	12.25	63	46
	0.05	22.31	76	67		0.05	19.80	87	62
	0.01	144.46	304	155		0.01	114.44	428	150
	0.005	275.64	903	189		0.005	215.05	567	186
	0.001	1554.37	1643	329		0.001	943.79	2554	351
0.005	0.5	2.37	21	67	0.01	0.5	2.24	42	30
	0.1	11.92	55	40		0.1	9.53	62	45
	0.05	22.17	80	72		0.05	18.51	73	56
	0.01	120.66	412	144		0.01	96.98	288	141
	0.005	268.30	568	194		0.005	197.90	519	182
	0.001	942.34	1301	338		0.001	1307.61	2312	334

(注) Del は Del-MCQN 法, Ext は Ext-MCQN 法である.

の関係式が成り立つ²⁵. 同様に, 既存の更新公式 (ここでは BFGS 公式) で更新すべき H_k の要素の数 $|F|$ と, グラフ $G' = (V, \bar{E})$ の枝数との間には,

$$|F| = 2 \times (\text{グラフ } G' \text{ の枝数}) + n$$

の関係式が成り立つ. 本実験では G, G' の枝数ではなく, $|E|, |F|$ を記録した.

結果を付録 C の表 3 に示す. 結果は, 各 n_z の値に対して条件数の小さい順に並べ替えて示した.

まず, 表 3 からコーダル縮小, コーダル拡張の枝数と MCQN 法の反復回数を比較する. $n_z = 0.01$ のとき, 行列 A の各行には平均 1 個しか非ゼロ要素がないことから, 先ほども述べたように行列 A の疎構造グラフはほとんどの場合コーダルグラフとなるため, 疎構造グラフの枝数とそのコーダル縮小, コーダル拡張の枝数はほぼ等しくなる. つまり, コーダル縮小を用いた場合の $|F|$ とコーダル拡張を用いた場合の $|F|$, そして行列 A の非ゼロ要素数 $|E|$ はほぼ等しくなる (実際, この実験では 10 回の試行すべてで一致している.) 行列 A の非ゼロ要素率を大きくすると $|E|$ はしだいに大きくなるが, コーダル拡張を用いた場合の $|F|$ の増加速度はそれよりも速い. 実際, $n_z = 0.1$ のときには $|F|$ は $|E|$ の約 3 倍である. 一方, コーダル縮小を

²⁵行列 A は正定値なので, A の対角成分はすべて正

用いた場合の $|F|$ は, n_z の値を大きくしてもそれほど大きくならず, $n_z = 0.1$ のときには $|E|$ の約半分, コーダル拡張を用いた場合の $|F|$ の約 $1/6$ になっている. このことから, Del-MCQN 法が必要とするメモリは, Ext-MCQN 法のそれよりも大幅に少なくなることがわかる.

次に反復回数を比較すると, すべての場合において Ext-MCQN 法が一番少なく, 次いで Del-MCQN 法, 対角成分だけを用いた MCQN 法の順であることがわかる. n_z の値を固定し, 条件数を大きくしていくと, Ext-MCQN 法の反復回数はほとんど変わらないが, Del-MCQN 法の反復回数は, しだいに増加する. 一方, n_z の値を大きくしても, Ext-MCQN 法, Del-MCQN 法の反復回数は若干増加する程度で, ほとんど変わらない. Del-MCQN 法と Ext-MCQN 法の反復回数の差は, 最も悪い時で 13 倍程度である. このことから, 反復回数という観点から見ると, Del-MCQN 法は, Ext-MCQN 法よりも劣っている. n_z の値を大きくしていくにつれて Del-MCQN 法の反復回数と対角成分のみを用いた MCQN 法の反復回数の差が小さくなっていく理由には, 関数 sprandsym で生成される行列の性質によると考えられる.

5 まとめと今後の課題

本報告書では, コーダル部分グラフを用いた MCQN 法 (Del-MCQN 法) を提案した. 理論的には Del-MCQN 法は Ext-MCQN 法よりも領域計算量, 反復 1 回あたりの時間計算量が小さいことを示した. また, 疎構造グラフのコーダル部分グラフを求めるアルゴリズムとして, 枝数最大のコーダル部分グラフを求める発見的解法として知られる Xue のアルゴリズムを用いて数値実験を行った. その結果から,

- 行列の条件数を大きくしたり, 非ゼロ要素率を大きくすると, Del-MCQN 法の反復回数は増加する.
- 反復回数の観点からは, Del-MCQN 法は Ext-MCQN 法より劣っているが, 反復 1 回あたりの計算にかかるコストの観点からは Ext-MCQN 法よりも優れている. したがって, 条件数がそれほど悪くない問題では Del-MCQN 法のほうが全体の計算量も小さいといえる.

ことがわかった.

まだ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.

- 複雑な問題での数値実験
本数値実験では, 凸 2 次計画問題に対して実験を行った. 今後は, より複雑な問題 (例えば, CUTER のテスト問題 [6]) に対して提案手法を適用し, その振る舞いを調べる必要がある.
- コーダル縮小を用いた MCQN 法と, 他のアルゴリズムとの計算量, 計算の比較
本数値実験では, Ext-MCQN 法との比較実験のみを行ったが, 他の手法とも比較してみる必要がある. また, 今回は MATLAB を用いてアルゴリズムを実装したが, 今後は C や Fortran で実装を行い, 計算にかかる時間を比較してみる必要がある.
- 疎構造グラフの枝数にもっとも近いコーダルグラフの利用
Ext-MCQN 法では, ヘッセ行列の疎構造グラフのコーダル拡張を, Del-MCQN 法では, コーダル縮小を用いていたが, 疎構造グラフにいくつか枝をつけくわえたり削ったりしてコーダルグラフにしたもの²⁶を用いて MCQN 法を適用することも考えられる. 疎構造グラフにつけくわえたり削ったりした枝の数が少なければ, ヘッセ行列の疎構造をよりよく保存できると考えられる.

²⁶一般に, このようなコーダルグラフを見つける問題は chordal editing 問題として知られている.

謝辞

まず、本報告書作成にあたり、細部にいたるまで熱心なご指導をいただいた山下信雄助教授に心より感謝申し上げます。また、日ごろからご教授くださり、本研究に対しても貴重なご指摘をいただいた福嶋雅夫教授に厚くお礼申し上げます。林俊介助手には、夜遅くまで実験の相談に乗っていただくなど、大変お世話になりました。深く感謝いたしております。福嶋研究室の皆様には、たくさんの助言と励ましの言葉をいただきました。ありがとうございました。

参考文献

- [1] J. R. S. Blair and B. W. Peyton. An introduction to chordal graphs and clique trees. In J. R. Gilbert A. George and J. W. H. Liu, editors, *Graph Theory and Sparse Matrix Computation*, pp. 1–29. Springer-Verlag, New York, 1993.
- [2] P. M. Dearing, D. R. Shier, and D. D. Warner. Maximal chordal subgraphs. *Discrete Applied Mathematics*, Vol. 20, pp. 181–190, 1988.
- [3] R. Fletcher. A new variational result for quasi-newton formulae. *SIAM Journal on Optimization*, Vol. 1, No. 1, pp. 18–21, 1991.
- [4] M. Fukuda, M. Kojima, K. Murota, and K. Nakata. Exploiting sparsity in semidefinite programming via matrix completion I: General framework. *SIAM Journal on Optimization*, Vol. 11, pp. 647–674, 2000.
- [5] 福嶋雅夫. 数理計画入門. 朝倉書店, 1996.
- [6] N. I. M. Gould, D. Orban, and P. L. Toint. Cuter and sifdec: A constrained and unconstrained testing environment, revisited. *ACM Transactions on Mathematical Software*, Vol. 20, pp. 373–394, 2003.
- [7] R. Grone, C. R. Johnson, E. M. Sá, and H. Wolkowicz. Positive definite completions of partial hermitian matrices. *Linear Algebra and its Applications*, Vol. 58, pp. 109–124, 1984.
- [8] A. Natanzon, R. Shamir, and R. Sharan. Complexity classification of some modification problems. *Discrete Applied Mathematics*, Vol. 113, pp. 109–128, 2001.
- [9] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer-Verlag New York, Inc., New York, 1999.
- [10] S. V. Parter. The use of linear graphs in gauss elimination. *SIAM Review*, Vol. 3, pp. 119–130, 1961.
- [11] J. D. Rose. A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations. In R. C. Reed, editor, *Graph Theory and Computing*, pp. 183–217. Academic Press, New York, 1972.
- [12] R. E. Tarjan and M. Yannakakis. Simple linear-time algorithms to test chordality of graphs, test acyclicity of hyper graphs, and selectively reduce acyclic hypergraphs. *SIAM Journal on Computing*, Vol. 13, pp. 566–579, 1984.
- [13] J. Xue. Edge-maximal triangulate subgraphs and heuristics for the maximum clique problem. *Networks*, Vol. 24, pp. 109–120, 1994.
- [14] 矢部博. 工学基礎 最適化とその応用. 数理工学社, 2006.
- [15] N. Yamashita. Sparse quasi-newton updates with positive definite matrix completion. Technical Report 008, Applied Mathematics and Physics, Kyoto University, 2005.

付録

A コーダグラフに関するアルゴリズム

A.1 MCS アルゴリズム

コーダグラフの PEO は, Tarjan と Yannakakis が提案した Maximum Cardinality Search アルゴリズム (以下 MCS 法) を用いれば, $O(|V| + |E|)$ で求めることができる [12]. そのアルゴリズムを以下に示す. 表記の簡単のため, $\mathcal{L}_i := \{v_i, v_{i+1}, \dots, v_n\}$ とかく.

MCS 法

Step 1: $\mathcal{L}_{n+1} := \emptyset, i := n$ とする.

Step 2: $|\text{Adj}(v) \cap \mathcal{L}_{i+1}|$ が最大となるような頂点 $v \in V - \mathcal{L}_{i+1}$ を選ぶ.

Step 3: $\alpha(v) := i, \mathcal{L}_i := \mathcal{L}_{i+1} \cup \{v_i\}$ とする.

Step 4: $i := i - 1$ とし, Step 2 へ.

この MCS 法によって計算される PEO は α である.

A.2 RIP を満たす極大クリーク族とクリーク木を求めるアルゴリズム

コーダグラフとその PEO から, RIP を満たす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.

PEO から RIP を満たす極大クリーク族とクリーク木を求めるアルゴリズム

Step 1: $r := 1, C_1 := \{v_n\}$, 極大クリーク族 $\mathcal{K}_n := \{C_1\}$, $\text{parent}(C_1) := \emptyset, i := n - 1$ とする.

Step 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 $C_q \in \mathcal{K}_{i+1}$ を求める.

Step 3: $C_q = A_i$ ならば, $C_r := C_q \cup \{v_i\}$ とする.

さもなければ, $r := r + 1, C_r := A_i \cup \{v_i\}$, $\text{parent}(C_r) := C_q, \mathcal{K}_i := \mathcal{K}_{i+1} \cup \{C_r\}$ とする.

Step 4: $i := i - 1$ とし, ステップ 2 へ.

このアルゴリズムでは, 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. クリーク木は各頂点 (クリーク) の親頂点 (クリーク) を表す関数 parent によって表現されている.

B 半正定値行列補完

本節では, 一部の要素だけが明記された行列のことを, 部分行列と呼ぶことにする. $n \times n$ 部分行列 \bar{X} は, \bar{X}_{ij} の集まりとして与えられる. $V = \{1, 2, \dots, n\}$ とし, $F \subseteq V \times V$ とする. \bar{X} には, $(i, j) \in F$ のところだけ要素が明記されているとしよう. 以下では, 部分行列 \bar{X} は対称行列, すなわち $(i, j) \in F \Rightarrow (j, i) \in F$ とする.

部分対称行列 \bar{X} の補完とは, \bar{X} の要素をすべて備えている対称行列を意味する. つまり, $X_{ij} = \bar{X}_{ij}, \forall (i, j) \in F$ を満たし, $(i, j) \notin F$ の部分の要素をすべて補ってできる行列 X が \bar{X} の行列補完である.

半正定値行列補完問題とは, 与えられた部分対称行列の補完である半正定値対称行列を見つける問題の

ことである .

頂点集合を V , 枝集合を $\bar{F} := F \setminus \{(i, i) \mid i = 1, \dots, n\}$ とするグラフ $G = (V, \bar{F})$ を考える²⁷ . その極大クリーク族を , $\{C_r \subseteq V \mid r = 1, 2, \dots, l\}$ とする . \bar{X} が (半) 正定値行列補完を持つための必要条件は , 「すべての $r = 1, 2, \dots, l$ に対して , 各 \bar{X}_{C_r, C_r} が (半) 正定値行列である」ことが知られている . ただし , \bar{X}_{C_r, C_r} は , G の極大クリーク C_r をもとにして作られた \bar{X} の小行列である . \bar{X}_{C_r, C_r} のすべての要素は明記されていることに注意しよう . この条件を **clique positive definite condition (CPDC)** という . すなわち , 極大クリーク族をもとにして作られるすべての小行列が CPDC を満たす場合には , その他の部分を適当に決めることで (半) 正定値行列が得られるのである . さらに , $G = (V, \bar{F})$ がコーダルグラフの場合には , その逆も成り立つ .

定理 B.1 (半正定値行列補完の必要十分条件 [7]) $G = (V, \bar{F})$ をコーダルグラフとする . このとき , 部分対称行列 \bar{X} が半正定値行列 X で補完可能であるための必要十分条件は , 極大クリーク族をもとに作られる各小行列 \bar{X}_{C_r, C_r} が CPDC を満たすことである .

さらに , $G = (V, \bar{F})$ がコーダルグラフであり , 部分対称行列 \bar{X} が CPDC を満たすならば , 以下の (1), (2) が成り立つ .

(1) \bar{X} の半正定値補完行列の中で , 行列式が最大のもので X とすると , それは一意に定まる .

すなわち

$$\begin{aligned} \max \quad & \det(X) \\ \text{subject to} \quad & X_{ij} = \bar{X}_{ij}, \forall (i, j) \in F \\ & X = X^T \\ & X \geq 0 \end{aligned} \tag{B.1}$$

の解は唯一つである .

(2)(B.1) の解として得られた補完行列 X は , $X_{ij}^{-1} = 0, \forall (i, j) \notin F$ を満たす .

注意 4 DFP 公式を与える問題 (3.2) の最適解を \bar{H} とする . \bar{H} の $(i, j) \in F$ の成分だけが与えられた部分行列を H^{QN} とする . $s_k^T y_k > 0$ かつ H_k は正定値対称を仮定すると , H^{QN} も正定値対称である . したがって , $G = (V, \bar{F})$ の極大クリーク族 $\{C_r \mid r = 1, 2, \dots, l\}$ をもとにして作られる H^{QN} の小行列 $H_{C_r, C_r}^{QN}, r = 1, 2, \dots, l$ は正定値対称である . よって , CPDC が満たされる . ゆえに , 問題 (3.7) の解として得られる行列 H は , $H_{ij}^{-1} = 0, \forall (i, j) \notin F$ を満たす .

²⁷ グラフからループを除いている

C 4節【実験2】の結果

表 3: コーダル縮小, コーダル拡張の枝数と反復回数との関係 (行列 A の次元 $n = 100$)

n	n_z		条件数	ヘッセ行列の 非ゼロ要素数	$ F $			反復回数		
					Del	Diag	Ext	Del	Diag	Ext
100	0.01	試行 1	180.68	112	112	100	112	61	364	61
		試行 2	199.30	112	112	100	112	59	241	59
		試行 3	237.22	112	112	100	112	62	580	62
		試行 4	302.46	112	112	100	112	62	985	62
		試行 5	326.12	112	112	100	112	60	1487	60
		試行 6	328.12	112	112	100	112	61	1208	61
		試行 7	423.25	112	112	100	112	62	1889	62
		試行 8	425.51	112	112	100	112	62	1419	62
		試行 9	444.46	112	112	100	112	63	1861	63
		試行 10	455.24	112	112	100	112	62	410	62
100	0.02	試行 1	155.82	218	218	100	218	68	420	68
		試行 2	195.56	220	216	100	238	69	709	72
		試行 3	206.88	222	220	100	224	67	418	67
		試行 4	230.57	220	218	100	222	61	588	61
		試行 5	260.10	222	222	100	222	66	720	66
		試行 6	265.74	222	222	100	222	66	310	66
		試行 7	279.02	222	220	100	244	66	650	66
		試行 8	279.86	222	220	100	226	148	845	62
		試行 9	288.76	218	216	100	220	77	741	68
		試行 10	383.53	222	216	100	242	115	694	68
100	0.03	試行 1	193.42	332	286	100	502	88	467	73
		試行 2	214.09	330	280	100	504	138	656	72
		試行 3	253.69	330	288	100	452	161	913	74
		試行 4	270.97	328	290	100	432	131	319	70
		試行 5	286.73	332	288	100	462	341	1200	78
		試行 6	291.43	332	288	100	476	101	1000	73
		試行 7	298.11	328	286	100	510	275	511	75
		試行 8	343.52	332	292	100	498	130	712	76
		試行 9	395.93	332	292	100	442	292	606	76
		試行 10	557.23	330	288	100	450	372	730	73
100	0.04	試行 1	131.87	440	332	100	922	192	654	78
		試行 2	179.11	426	336	100	756	120	396	74
		試行 3	237.43	440	334	100	832	225	309	80
		試行 4	243.53	438	336	100	824	392	713	77
		試行 5	260.26	442	330	100	908	398	801	78
		試行 6	260.38	428	322	100	860	430	578	82
		試行 7	279.60	438	326	100	794	247	503	77
		試行 8	454.59	434	312	100	830	557	588	82
		試行 9	505.90	436	328	100	834	582	734	76
		試行 10	650.66	440	314	100	904	209	938	83
100	0.05	試行 1	156.76	544	376	100	1256	215	702	87
		試行 2	178.49	538	346	100	1340	304	498	86
		試行 3	184.39	542	366	100	1336	703	686	85
		試行 4	205.28	542	354	100	1292	352	532	82
		試行 5	226.63	530	374	100	1190	140	256	82
		試行 6	245.21	536	356	100	1268	171	359	82
		試行 7	255.59	544	352	100	1342	169	275	79
		試行 8	257.34	540	370	100	1292	263	542	88
		試行 9	313.28	538	356	100	1328	518	627	81
		試行 10	384.89	538	348	100	1268	191	559	69
100	0.06	試行 1	127.11	648	374	100	1698	271	370	80
		試行 2	150.84	646	366	100	1932	275	286	86
		試行 3	166.04	660	386	100	1890	217	332	88
		試行 4	171.41	652	372	100	1890	139	478	81
		試行 5	173.16	644	378	100	1692	209	297	83
		試行 6	175.83	644	382	100	1788	167	257	77
		試行 7	179.87	650	360	100	1834	154	341	81
		試行 8	207.58	630	370	100	1756	157	568	85
		試行 9	261.36	650	400	100	1734	413	479	82
		試行 10	450.50	646	376	100	1630	1139	1441	86
100	0.07	試行 1	124.97	752	418	100	2134	177	465	83
		試行 2	143.01	756	384	100	2344	225	294	81
		試行 3	201.01	752	424	100	2224	228	300	80
		試行 4	224.67	748	414	100	2108	257	304	81
		試行 5	228.05	754	412	100	2202	292	329	83
		試行 6	240.36	742	408	100	2112	141	322	84
		試行 7	260.27	752	402	100	2260	346	372	88
		試行 8	275.88	754	396	100	2202	324	400	85
		試行 9	340.40	754	404	100	2364	408	450	83
		試行 10	372.16	750	402	100	2118	413	460	89
100	0.08	試行 1	137.19	848	454	100	2474	261	357	76

n	nc		条件数	ヘッセ行列の 非ゼロ要素数	F			反復回数		
					Del	Diag	Ext	Del	Diag	Ext
		試行 2	143.52	842	448	100	2488	201	378	83
		試行 3	163.02	842	444	100	2560	267	398	81
		試行 4	165.89	842	434	100	2778	316	458	85
		試行 5	175.26	848	452	100	2852	467	504	81
		試行 6	187.27	836	414	100	2560	402	547	84
		試行 7	204.29	844	454	100	2452	269	516	87
		試行 8	210.25	824	438	100	2616	425	604	82
		試行 9	219.90	830	442	100	2786	228	424	81
		試行 10	238.38	838	442	100	2462	237	726	88
		100	0.09	試行 1	146.05	910	466	100	2658	205
試行 2	159.35			936	468	100	2828	161	487	82
試行 3	169.45			932	484	100	2962	205	398	77
試行 4	183.05			904	462	100	2714	122	480	86
試行 5	200.64			914	464	100	2660	388	437	77
試行 6	208.00			934	460	100	2926	308	413	81
試行 7	219.08			930	486	100	2742	433	530	84
試行 8	296.80			908	460	100	2522	245	430	72
試行 9	312.00			918	446	100	2998	409	425	84
試行 10	420.97			918	466	100	2682	305	707	86
100	0.1	試行 1	123.92	1010	490	100	3412	321	351	81
		試行 2	168.49	1018	488	100	2996	359	505	80
		試行 3	170.12	1000	474	100	3180	502	514	84
		試行 4	190.47	1012	482	100	3156	329	445	86
		試行 5	201.76	1006	474	100	3220	525	658	81
		試行 6	250.53	1002	488	100	3056	381	521	82
		試行 7	271.90	1008	478	100	2870	635	721	83
		試行 8	277.90	1004	504	100	3088	320	500	81
		試行 9	293.81	1008	488	100	3132	220	637	76
		試行 10	330.38	990	470	100	2866	322	748	82

(注) |F|: 近似行列 H_k の要素のうち、既存の準ニュートン法の更新公式 (ここでは BFGS 公式) で更新すべき要素の数、Xue: コーダル縮小を Xue のアルゴリズムを用いて求めた場合、Diag: 対角成分のみを用いた場合、Ext: コーダル拡張を用いた場合である。