数理解析研究所講究録
第1931巻 2015年 47-66

無制約最適化問題に対する準ニュートン・パターンサーチ法

東京理科大学・理学部数理情報科学科　矢部　博
キャノンITソリューションズ株式会社 稲葉洋介
Hiroshi Yabe
Department of Mathematical Information Science, Tokyo University of Science
Inaba Yosuke
Canon IT Solutions Inc.

1　はじめに

本論文では, 以下の無制約最小化問題について考える.

$$\min\ f(x) \tag{1.1}$$

ただし, $f:R^{n}\rightarrow R$ は十分になめらかな関数とする. (1.1) を解くためによく使用されている反復法は

$$x^{(k+1)}=x^{(k)}+\alpha^{(k)}d^{(k)}$$

によって点列 $\{x^{(k)}\}$ を生成するものである. ここで, $\alpha^{(k)}>0$ をステップ幅, $d^{(k)}\in R^{n}$ を探索方向と呼ぶ. 目的関数の導関数が利用できないとき, または, 計算コストがかかるときには, 微分を用いることを前提とした解法はそのままでは使用できない. そこで, 直接探索法のような微分を用いない解法を考える必要があると言える. こうした理由から, 近年, 目的関数の微分を用いない解法に注目が集まっている [9, 4]. その中の一つにパターンサーチ法と呼ばれる解法がある. パターンサーチ法は1960年代に Hooke and Jeeves, Rosenbrock らによって提案されたものであり [10], 単純かつ実用的な解法であるため, 今日でも広く使われている. 近年, 多くの研究者が無制約最適化問題に対するパターンサーチ法に再び注目するようになり, いろいろな研究が行われてきた. 例えば, Dennis and Torczon [8] , Torczon [15] , Coope and Price [5, 6], Byatt, Coope and Price [3] らの研究が挙げられる. これらの解法は微分を用いることなく, グリッドを用いることで新たな点を探す特徴がある. しかし, 単純かつ実用的ではあるが, 一般的に微分を用いる勾配法と比較して収束が遅いという欠点を持つ. その理由として, パターンサーチ法では反復ごとに多くの探索方向を計算する必要があるからである. この状況を解決するために多くの研究者は様々な技法を用いることで, パターンサーチ法を修正してきた. 一般的にパターンサーチ法を改善する上で2つの課題がある. すなわち, 「反復点採用判断基準の選択」と「探索方向の集合の定義」である. 前者に対して, フィルターに基づいた解法 [17] が使われている. 後者に対しては, 現時点での探索方向の集合が R^{n} の基底で構成される. R^{n} の全空間を探索することは可能ではあるが, 多くの探索方向を持つことになり, 結果的にアルゴリズムの効率性を下げることになる. そこで近年, 正の基底, 共役性, そして並列処理がこの問題に対して使われるようになった. その中でも, 特に正の基底がよく使われている [6, 16, 18].

パターンサーチ法の効率を改善する方法の1つに, パターンサーチ法と他の解法を組み合わせる手法がある. 特に準ニュートン法の考えを組み入れたパターンサーチ法に関する研究が進められて

いる. このアプローチでは, ヘッセ行列の逆行列の近似行列 $H^{(k)}$ の分解行列が, パターンサーチの過程で必要となる正の基底を作るのに利用される. また, 準ニュートン法をパターンサーチ法に組み込むことによって, 収束を加速することが可能になることが期待される. こうした理由から, Wu and Sun [16] は BFGS 公式に基づいた準ニュートン・パターンサーチ法を提案し, その後, Wu and Sun [18] は修正対称ランクワン公式に基づいた準ニュートン・パターンサーチ法を提案した. 本研究では, BFGS 公式などの代表的な更新公式を含む Broyden 公式族に着目して, 公式族の形で準ニュートン・パターンサーチ法を提案する.

本論文の構成について述べる. 第 2 節では Coope and Price [6] が提案したグリッドに基づいた方法を紹介する. 第 3 節では準ニュートン・パターンサーチ法について触れる. まずはじめに 3.1 節で勾配法としての準ニュートン法を紹介し, 3.2 節で Broyden 公式族を用いた準ニュートン・パターンサーチ法を提案する. さらに 3.3 節で Wu and Sun [18] が提案した修正対称ランクワン公式を用いた準ニュートン・パターンサーチ法を紹介する. そして最後に, 第 4 節で数値実験結果を報告する.

以下では, I は単位行列とし, $||\cdot||$ はベクトルもしくは行列に対する 2 ノルムとする.

2 パターンサーチ法

本節では Coope and Price [6] が提案したグリッドに基づいた方法を紹介する. パターンサーチ法は目的関数の導関数の情報を用いずに点列 $\{x^{(k)}\}$ を生成する手法である. 反復ごとに, 目的関数は以下で定義する試行点で評価され, 現在の点より関数値が下がる点を見つけることが各反復での目的になる. もし条件を満たす点が見つかった場合は新しい点として用いて, この反復を「成功」と呼ぶ. 見つからない場合は「失敗」とし, 試行点を更新する.

2.1 グリッドに基づいた方法と正の基底

グリッドに基づいた方法とは, 一連のグリッドにおける目的関数値を調べることによって問題 (1.1) の最適解を求めるアルゴリズムのことである. グリッド $\mathcal{G}^{(m)}$ は n 個の線形独立な基底ベクトル $V^{(m)}$ によって次のように定義される.

$$\mathcal{G}^{(m)} = \{x \in R^n \ : \ x = x_0^{(m)} + h^{(m)} \sum_{i=1}^{n} \eta_i v_i^{(m)}\},$$

ただし

$$V^{(m)} = \{v_i^{(m)} \in R^n \ : \ i = 1, \cdots, n\}$$

である. また, $h^{(m)}$ は正のスカラー, η_i は整数とする. パラメータ $h^{(m)}$ はメッシュサイズと呼ばれており, 収束性を確立するために, m が増加するにつれてメッシュがより良くなるように調整される. $V^{(m)}$ における基底ベクトルはグリッド $\mathcal{G}^{(m)}$ の軸と平行である.

集合 $V^{(m)}$ は正の基底 $V_+^{(m)}$ を作るのに使われる. ただし, $V_+^{(m)}$ が正の基底であるためには次の 2 つの条件が要求される.

(i) R^n の全てのベクトルは, $V_+^{(m)}$ のベクトルの非負の線形結合によって表すことが出来る.

(ii) $V_+^{(m)}$ に含まれるベクトルは, $V_+^{(m)}$ の残りのベクトルの非負の線形結合によって表すことが出来ない.

$\{v_1, v_2, \ldots, v_n\}$ を R^n の一組の基底としたとき, 正の基底の例として

$$\{v_1, v_2, \ldots, v_n, -\sum_{i=1}^{n} v_i\} \tag{2.1}$$

あるいは,

$$\{v_1, v_2, \ldots, v_n, -v_1, -v_2, \ldots, -v_n\}$$

などがあげられる.

ここで, $\eta^{(m)}$ を $V_+^{(m)}$ の基数とする. 以下では, $V_+^{(m)}$ の最初の n 個の元は $V^{(m)}$ の要素とし, 残りの要素は $V^{(m)}$ の元の整数の線形結合によって以下のように与えられるものと仮定する.

$$v_j^{(m)} = \sum_{i=1}^{n} \xi_{ij}^{(m)} v_i^{(m)},\ j = n+1, \ldots, \eta^{(m)}, \tag{2.2}$$

ただし, $x \in \mathcal{G}^{(m)}$, $v \in V_+^{(m)}$ ならば, $x + h^{(m)}v \in \mathcal{G}^{(m)}$ となるように整数 $\xi_{ij}^{(m)}$ は選ばれる. 式 (2.2) では $V_+^{(m)}$ の元が特別な順番に並んでいると仮定する. このとき, (2.2) を満たす正の基底は「順序付けられた正の基底」と呼ばれる.

順序づけられた正の基底を用いた場合のグリッド上の探索における終了条件は, 次の定理によって与えられる.

定理 2.1 *([6])* ベクトルの集合 V_+ が正の基底であるとき,

$$g^T v \geq 0 \ \ \forall v \in V_+ \ \Rightarrow \ g = 0$$

が成り立つ.

上記の定理は, 以下の定義を動機づける.

定義 2.1 (グリッド極小解) 次の式が成り立つとき, グリッド $\mathcal{G}^{(m)}$ 上の点 x は正の基底 $V_+^{(m)}$ に関してグリッド極小解であるという.

$$f(x + h^{(m)} v_i) \geq f(x) \ \ \forall v_i \in V_+^{(m)}$$

収束性を確立するために, グリッド極小解を定義するのに使われる順序付けられた正の基底に関していくつかの条件が課される. 次の定義は, これらの条件が順序付けられた正の基底の要素間の線形関係として簡単に表現できることを示している.

定義 2.2 (構造的同値性) *2* つの順序づけられた正の基底 $\{v_1, \ldots, v_p\}$ と $\{w_1, \ldots, w_p\}$ が構造的同値であるとは, 任意の $j > n$ に対して, $v_j = \sum_{i=1}^{n} \zeta_{ij} v_i$ と $w_j = \sum_{i=1}^{n} \zeta_{ij} w_i$ が成り立つことである.

2.2 アルゴリズムの概要とその収束性

前節で述べてきた「順序付けられた正の基底」と「グリッドに基づいた方法」を組み込んだアルゴリズムの概要を説明していく.

以下のアルゴリズムは外部反復と内部反復から構成されている. 外部反復 (Step1-3) では, グリッドを選び停止条件を満たすかどうかを調べる. 内部反復 (Step2) では, $V_+^{(m)}$ のメンバーを用いて新しい点を探す. グリッド極小解が見つかったら内部反復を終了し, 外部反復において新しいグリッドを選択する.

アルゴリズム 2.1
$m = 0, k = 0$ とおく. $x_0^{(0)}$ は初期点 $x^{(0)}$ とする.
停止条件が満たされるまで, Step1 から step3 を実行する.

Step1. $h^{(m)}$, $V_+^{(m)}$ を選ぶ. $i = 1$, $num = 0$, $\eta^{(m)} = |\ V_+^{(m)}\ |$ とする.

Step2. num が $\eta^{(m)}$ に達するまで，以下の内部反復（(a) と (b)）を繰り返す．

(a) グリッド $\mathcal{G}^{(m)}$ における有限個の点 $(x^{(k)}+h^{(m)}v_i^{(m)}$ を含む）において目的関数値を計算する．もし $x^{(k)}$ より関数値が下がる点があった場合，その中の最小点を $x^{(k+1)}$ とし，$k=k+1$, $num=0$ とする．存在しない場合は $num=num+1$ とする．

(b) $i=i+1$ とする．もし $i>\eta^{(m)}$ ならば，$i=1$ とする．

Step3. グリッド極小解を $\hat{x}^{(m)}=x^{(k)}$ とする．直線探索法などの有限回手順を実行することで得られる現時点で最も関数値の低い点 $x_0^{(m+1)}$ を $x^{(k+1)}=x_0^{(m+1)}$ とおく．そして $k=k+1$, $m=m+1$ とする．

このアルゴリズムにおいて，num は $V_+^{(m)}$ を用いた探索に続けて失敗した回数である．$num=\eta^{(m)}$ のとき，グリッド極小解 $\hat{x}^{(m)}$ が見つかり，グリッド $\mathcal{G}^{(m)}$ における探索が終了する．
アルゴリズム 2.1 の収束性については，次の定理で与えられる．

定理 2.2 ([6]) アルゴリズム 2.1 において，以下の条件が成り立つと仮定する

1. 点列 $\{x^{(k)}\}$ は有界である．

2. f は連続的微分可能である．

3. 全ての m,i に対して $|det(v_1^{(m)},\ldots,v_n^{(m)})|\geq\kappa$ と $||v_i^{(m)}||\leq K$ を満たす正の定数 κ, K が存在する．

4. $\lim_{m\to\infty}h^{(m)}=0$.

5. B を順序づけられた正の基底 $\{V_+^{(m)}\}_{m=1}^{\infty}$ の列としたとき，B の各メンバーがその有限部分集合に含まれる適当なメンバーと構造的同値であるような B の有限部分集合が存在する．

このとき，グリッド極小解の点列 $\{\hat{x}^{(m)}\}$ は無限に多くのメンバーを持ち，そして，$\{\hat{x}^{(m)}\}$ の任意の集積点 $\hat{x}^{(\infty)}$ は f の停留点となる．

定理 2.2 は，グリッド極小解の点列がどのように生成されるかについてほとんど制限していない．そのかわり，点列が有界かつ無限個のメンバーをもつことの仮定が必要となる．仮定 3 は，適切に $V_+^{(m)}$ を選ぶことで容易に満たされる．仮定 4 を満たす単純な方法は，グリッド極小解が見つかる度にメッシュサイズを半分にすることである．仮定 5 は実用上のものである．
定理 2.2 は一般定理なので，いろいろなアルゴリズムに適用できることを注意しておく．その一方で，この定理は生成される点列の部分列の収束性しか保証していない．そこで点列全体の収束性を達成するために，次にアルゴリズム 2.1 に比べてもっと限定されたアルゴリズム 2.2 を紹介する．このアルゴリズムでは，$x^{(k)}$ における"best drop"を与えるような $V_+^{(m)}$ のメンバー $z^{(k)}$ に沿って探索が実行される．$V_+^{(m)}$ の best drop メンバー $z^{(k)}$ は

$$f(x^{(k)}+h^{(m(k))}z^{(k)})\leq f(x^{(k)}+h^{(m(k))}v)\quad\forall v\in V_+^{(m(k))} \tag{2.3}$$

を満たす．ただし $m(k)$ は $x^{(k)}$ を位置付けるグリッドの個数である．$z^{(k)}$ を決定するためには $\eta^{(m(k))}$ 回の関数評価が必要となる．それぞれの $z^{(k)}$ に沿った探索は，次式で定義される点ごとに f の評価をしなければならない．

$$\tilde{x}_i=x^{(k)}+\alpha_i h^{(m(k))}z^{(k)},\quad i=0,\ldots,$$

ただし，$\alpha_0=1$ であり，整数列 $\{\alpha_i\}$, $i\geq1$, は次式を満たす．

$$\alpha_{i-1}+1\leq\alpha_i\leq\beta\alpha_{i-1},\quad\beta\geq2$$

この探索は，$f(\tilde{x}_l)\leq f(\tilde{x}_{l+1})$ となるような整数 $l\geq0$ が見つかったときにのみ終了する．
以上の考えに基づいたアルゴリズムは次のように記述できる．

アルゴリズム 2.2
$m=0, k=0$ とおく. $x_0^{(0)}$ は初期点 $x^{(0)}$ とする.
停止条件が満たされるまで, Step1 から step3 までを実行する.

Step1. $h^{(m)}$, $V_+^{(m)}$ を選ぶ. $i=1$, $num=0$, $\eta^{(m)}=\mid V_+^{(m)}\mid$ とする.

Step2. $x^{(k)}$ がグリッド極小解になるまで, 以下の内部反復 ((a), (b), (c)) を繰り返す.

(a) (2.3) を満たす best drop 方向 $z^{(k)}$ を計算する. もし $f(x^{(k)}) \leq f(x^{(k)}+h^{(m)}z^{(k)})$ を満たすならば step2 を終了し, $x^{(k)}$ をグリッド極小解とする.

(b) $\alpha_0=1$ から始めて, $\alpha_{j+1}\in[\alpha_j+1\ ,\ \beta\alpha_j]$ のルールに従って作られる整数 α_1 , α_2 , ... に対して

$$f(x^{(k)}+\alpha_{l+1}h^{(m)}z^{(k)}) \geq f(x^{(k)}+\alpha_l h^{(m)}z^{(k)})$$

を満たす整数 α_l を選ぶ.

(c) 有限個のグリッド点において目的関数値を計算し, それらの点と $x^{(k)}+\alpha_l h^{(m)}s^{(k)}$ の中で最小値を与える点を $x^{(k+1)}$ として選ぶ. そして $k=k+1$ とする.

Step3. グリッド極小解を $\hat{x}^{(m)}=x^{(k)}$ とする. 直線探索法等の有限回手順を実行することで得られる現時点で最も関数値の低い点 $x_0^{(m+1)}$ を $x^{(k+1)}=x_0^{(m+1)}$ とおく. そして $k=k+1$, $m=m+1$ とする.

以下の定理は, アルゴリズム 2.2 で生成される点列の収束性を保証している.

定理 2.3 ([6]) 定理 2.2 の仮定が成り立つとき, アルゴリズム *2.2* で生成される点列は目的関数 f の停留点に収束する.

3 準ニュートン・パターンサーチ法

3.1 準ニュートン法

無制約最小化問題 (1.1) を解くための有効な勾配法の 1 つとして, 準ニュートン法がよく知られている [19, 20]. k 回目の反復における近似解 $x^{(k)}$ において $H^{(k)}\approx[\nabla^2 f(x^{(k)})]^{-1}, g^{(k)}=\nabla f(x^{(k)})$ が与えられたとき, 探索方向 $d^{(k)}$ は

$$d^{(k)}=-H^{(k)}g^{(k)}$$

によって求められる. そして, $d^{(k)}$ に沿って直線探索を行うことで新しい点を得る:

$$x^{(k+1)}=x^{(k)}-\alpha^{(k)}H^{(k)}g^{(k)}$$

ただし, $\alpha^{(k)}>0$ は以下の Wolfe 条件を満たすステップ幅である.

$$f(x^{(k)})-f(x^{(k)}-\alpha^{(k)}H^{(k)}g^{(k)}) \ \geq\ \sigma\alpha^{(k)}g^{(k)T}H^{(k)}g^{(k)}, \tag{3.1}$$

$$g(x^{(k)}-\alpha^{(k)}H^{(k)}g^{(k)})^T H^{(k)}g^{(k)} \ \leq\ \tau g^{(k)T}H^{(k)}g^{(k)} \tag{3.2}$$

ここで $\sigma\in(0,1/2)$, $\tau\in(\sigma,1)$ であり, $g(x)\equiv\nabla f(x)$ である. 準ニュートン法では, 行列 $H^{(k)}$ は以下のセカント条件を満たすように更新される.

$$H^{(k+1)}y^{(k)}=s^{(k)}$$

ただし $y^{(k)} = g^{(k+1)} - g^{(k)}$, $s^{(k)} = x^{(k+1)} - x^{(k)}$ である. $H^{(k)}$ から $H^{(k+1)}$ への更新公式はいろいろあるが, 代表的なものでは BFGS 公式, DFP 公式, 対称ランクワン (SR1) 公式等が挙げられる. これらの公式は Broyden 公式族としてまとめられる.

$$\begin{aligned} H^{(k+1)} &= H^{(k)} - \frac{H^{(k)}y^{(k)}{y^{(k)}}^T H^{(k)}}{{y^{(k)}}^T H^{(k)} y^{(k)}} + \frac{s^{(k)}{s^{(k)}}^T}{{s^{(k)}}^T y^{(k)}} \\ &\quad + \phi^{(k)}({y^{(k)}}^T H^{(k)} y^{(k)}) u^{(k)}{u^{(k)}}^T \end{aligned} \tag{3.3}$$

ただし

$$u^{(k)} = \frac{H^{(k)}y^{(k)}}{{y^{(k)}}^T H^{(k)} y^{(k)}} - \frac{s^{(k)}}{{s^{(k)}}^T y^{(k)}}$$

であり, $\phi^{(k)}$ は実パラメータである. ここで, $\phi^{(k)} = 1$ のとき BFGS 公式

$$H^{(k+1)} = H^{(k)} - \frac{H^{(k)}y^{(k)}{s^{(k)}}^T + s^{(k)}{y^{(k)}}^T H^{(k)}}{{s^{(k)}}^T y^{(k)}} + \left(1 + \frac{{y^{(k)}}^T H^{(k)} y^{(k)}}{{s^{(k)}}^T y^{(k)}}\right) \frac{s^{(k)}{s^{(k)}}^T}{{s^{(k)}}^T y^{(k)}}, \tag{3.4}$$

$\phi^{(k)} = 0$ のとき DFP 公式

$$H^{(k+1)} = H^{(k)} - \frac{H^{(k)}y^{(k)}{y^{(k)}}^T H^{(k)}}{{y^{(k)}}^T H^{(k)} y^{(k)}} + \frac{s^{(k)}{s^{(k)}}^T}{{s^{(k)}}^T y^{(k)}},$$

$\phi^{(k)} = \frac{{s^{(k)}}^T y^{(k)}}{(s^{(k)} - H^{(k)}y^{(k)})^T y^{(k)}}$ のとき SR1 公式

$$H^{(k+1)} = H^{(k)} + \frac{(s^{(k)} - H^{(k)}y^{(k)})(s^{(k)} - H^{(k)}y^{(k)})^T}{(s^{(k)} - H^{(k)}y^{(k)})^T y^{(k)}}$$

をそれぞれ得る. Broyden 公式族は, パラメータ $\phi^{(k)}$ が $0 \leq \phi^{(k)} \leq 1$ を満たすときに, $H^{(k)}$ が正定値対称で ${s^{(k)}}^T y^{(k)} > 0$ という仮定のもとで正定値対称性を保存することが知られているので, BFGS 公式と DFP 公式は正定値対称性を保存する. なお, ${s^{(k)}}^T y^{(k)} > 0$ という仮定は直線探索で Wolfe 条件を課すことで満たされる. その一方で, SR1 公式は, $H^{(k)}$ が正定値であっても $H^{(k+1)}$ が正定値であるとは限らないため, 降下方向が得られる保証がない. この問題点を解決するため, Sun [14] は次の修正対称ランクワン公式を提案した.

$$H^{(k+1)} = \theta H^{(k)} + \frac{(s^{(k)} - \theta H^{(k)}y^{(k)})(s^{(k)} - \theta H^{(k)}y^{(k)})^T}{(s^{(k)} - \theta H^{(k)}y^{(k)})^T y^{(k)}} \tag{3.5}$$

この公式において, $\theta > 0$, ${s^{(k)}}^T y^{(k)} > 0$, $H^{(k)}$ が正定値対称行列であるとき, $H^{(k+1)}$ が正定値行列になるための必要十分条件は

$$\theta \notin \left[\frac{{s^{(k)}}^T y^{(k)}}{{y^{(k)}}^T H^{(k)} y^{(k)}}, \frac{{s^{(k)}}^T {H^{(k)}}^{-1} s^{(k)}}{{s^{(k)}}^T y^{(k)}}\right] \tag{3.6}$$

が成り立つことであることが示されている.

3.2 準ニュートン・パターンサーチ法（Broyden 公式族）

パラメータを $0 \leq \phi^{(k)} \leq 1$ を満たすように選んだ場合の Broyden 公式族 (3.3) では $H^{(k)}$ は正定値対称性を保存するので, $H^{(k)} = L^{(k)}{L^{(k)}}^T$ と分解できる. ただし $L^{(k)} = \left[l_1^{(k)} l_2^{(k)} \ldots l_n^{(k)}\right]$ は n 次

正則行列であり，$l_i^{(k)} \in R^n (i = 1, \ldots, n)$ である．このとき, Broyden 公式族の分解型公式は以下のようになる [1, 2].

$$\begin{aligned} L^{(k+1)} &= L^{(k)} + \left(q^{(k)} - \frac{1 - \sqrt{\phi^{(k)}}}{{y^{(k)}}^T H^{(k)} y^{(k)}} H^{(k)} y^{(k)} \right) {y^{(k)}}^T L^{(k)} \\ &\quad + \frac{\sqrt{\phi^{(k)} \delta^{(k)}}}{{s^{(k)}}^T y^{(k)}} s^{(k)} {s^{(k)}}^T {L^{(k)}}^{-T}, \end{aligned} \tag{3.7}$$

ただし，

$$\begin{aligned} q^{(k)} &= \left(\frac{(1 - \sqrt{\phi^{(k)}})\sqrt{\delta^{(k)}}}{{y^{(k)}}^T H^{(k)} y^{(k)}} - \frac{\sqrt{\phi^{(k)}}}{{s^{(k)}}^T y^{(k)}} \right) s^{(k)}, \\ \delta^{(k)} &= \left((1 - \phi^{(k)}) \frac{{s^{(k)}}^T y^{(k)}}{{y^{(k)}}^T H^{(k)} y^{(k)}} + \phi^{(k)} \frac{{s^{(k)}}^T B^{(k)} s^{(k)}}{{s^{(k)}}^T y^{(k)}} \right)^{-1}, \end{aligned}$$

$$H^{(k)} = L^{(k)} {L^{(k)}}^T, \quad B^{(k)} = (H^{(k)})^{-1} = {L^{(k)}}^{-T} {L^{(k)}}^{-1}$$

である．この式において，特に $\phi^{(k)} = 1$ とおけば BFGS 公式 (3.4) の分解型公式が次のように得られる [7, 20, 16].

$$L^{(k+1)} = L^{(k)} + \frac{s^{(k)}}{{s^{(k)}}^T y^{(k)}} \left(\sqrt{\frac{{s^{(k)}}^T y^{(k)}}{{s^{(k)}}^T (L^{(k)} {L^{(k)}}^T)^{-1} s^{(k)}}} (L^{(k)} {L^{(k)}}^T)^{-1} s^{(k)} - y^{(k)} \right)^T L^{(k)} \tag{3.8}$$

式 (3.7) には $(L^{(k)} {L^{(k)}}^T)^{-1} s^{(k)}$ が含まれているが, ${H^{(k)}}^{-1} = B^{(k)}$, $B^{(k)} d^{(k)} = -g^{(k)}$ より

$$\begin{aligned} (L^{(k)} {L^{(k)}}^T)^{-1} s^{(k)} &= B^{(k)} s^{(k)} \\ &= \alpha^{(k)} B^{(k)} d^{(k)} \\ &= -\alpha^{(k)} g^{(k)} \end{aligned}$$

と変形することで逆行列の計算を回避することができる．ただし，$\alpha^{(k)}$ は直線探索によって得られるステップ幅である．ここで, $\hat{g}^{(k)} = {L^{(k)}}^T g^{(k)}$ とすると, その第 i 成分は

$$\hat{g}_i^{(k)} = {l_i^{(k)}}^T g^{(k)}, \ i = 1, 2, \ldots, n \tag{3.9}$$

となる．$\hat{g}_i^{(k)}$ は点 $x^{(k)}$ における $l_i^{(k)}$ 方向の $f(x)$ の方向微係数である．このとき, 準ニュートン方向は以下のようにして得られる．

$$\begin{aligned} d^{(k)} &= -H^{(k)} g^{(k)} \\ &= -L^{(k)} {L^{(k)}}^T g^{(k)} \\ &= -L^{(k)} \hat{g}^{(k)} \end{aligned}$$

式 (3.9) において, $\hat{g}^{(k)}$ の要素は有限差分法による近似計算を用いて導くことが出来る．例えば前進差分, 中心差分は以下のとおりである．ただし, δ_i はきざみ幅である．

前進差分：

$$\hat{g}_i^{(k)} = \frac{f(x^{(k)} + \delta_i l_i^{(k)}) - f(x^{(k)})}{\delta_i} + \mathcal{O}(\delta_i) \tag{3.10}$$

中心差分：

$$\hat{g}_i^{(k)} = \frac{f(x^{(k)} + \delta_i l_i^{(k)}) - f(x^{(k)} - \delta_i l_i^{(k)})}{2\delta_i} + \mathcal{O}(\delta_i^2) \tag{3.11}$$

さらに

$$\hat{y}^{(k)} = L^{(k)^T} y^{(k)} = \breve{g}^{(k+1)} - \hat{g}^{(k)} \tag{3.12}$$

とおく. ただし, $\breve{g}^{(k+1)} = L^{(k)^T} g^{(k+1)}$ である. このとき,

$$y^{(k)^T} H^{(k)} y^{(k)} = y^{(k)^T} L^{(k)} L^{(k)^T} y^{(k)} = (\hat{y}^{(k)})^T \hat{y}^{(k)}, \tag{3.13}$$

$$s^{(k)^T} y^{(k)} = \alpha d^{(k)^T} y^{(k)} = -\alpha g^{(k)^T} L^{(k)} L^{(k)^T} y^{(k)} = -\alpha (\hat{g}^{(k)})^T \hat{y}^{(k)}, \tag{3.14}$$

$$\begin{aligned} s^{(k)^T} H^{(k)^{-1}} s^{(k)} &= \alpha^2 d^{(k)^T} B^{(k)} d^{(k)} \\ &= \alpha^2 g^{(k)^T} L^{(k)} L^{(k)^T} L^{(k)^{-T}} L^{(k)^{-1}} L^{(k)} L^{(k)^T} g^{(k)} \\ &= \alpha^2 (\hat{g}^{(k)})^T \hat{g}^{(k)} \end{aligned} \tag{3.15}$$

のような式変形を利用すれば，微分を用いない Broyden 公式族の分解型公式が得られる．例えば、BFGS 公式の分解型公式 (3.8) は以下のように書き換えられる．

$$\begin{aligned} L^{(k+1)} &= L^{(k)} + \frac{s^{(k)}}{s^{(k)^T} y^{(k)}} \left(\sqrt{\frac{s^{(k)^T} y^{(k)}}{s^{(k)^T} (L^{(k)} L^{(k)^T})^{-1} s^{(k)}}} (L^{(k)} L^{(k)^T})^{-1} s^{(k)} - y^{(k)} \right)^T L^{(k)} \\ &= L^{(k)} + \sqrt{\frac{s^{(k)^T} y^{(k)}}{s^{(k)^T} (L^{(k)} L^{(k)^T})^{-1} s^{(k)}}} \frac{(-\alpha^{(k)} s^{(k)} g^{(k)^T} L^{(k)})}{s^{(k)^T} y^{(k)}} - \frac{s^{(k)} y^{(k)^T} L^{(k)}}{s^{(k)^T} y^{(k)}} \\ &= L^{(k)} + \sqrt{\frac{-\hat{g}^{(k)^T} \hat{y}^{(k)}}{\alpha^{(k)} \hat{g}^{(k)^T} \hat{g}^{(k)}}} \frac{s^{(k)} \hat{g}^{(k)^T}}{\hat{g}^{(k)^T} \hat{y}^{(k)}} + \frac{s^{(k)} \hat{y}^{(k)^T}}{\alpha^{(k)} \hat{g}^{(k)^T} \hat{y}^{(k)}} \end{aligned} \tag{3.16}$$

以下では，アルゴリズム 2.1 と Broyden 公式族の分解型公式を組み合わせたアルゴリズムについて述べる．このアルゴリズムでは $f(x)$ の方向微係数を有限差分法によって計算し，それから準ニュートン方向を生成する．また，準ニュートン方向に沿った直線探索は新しい点を探すのと同時に $L^{(k)}$ を更新する役割をもつ．以下のアルゴリズムにおける $x^{(m)}$ は許容パターンサーチ点を表し，k は反復回数とする．

アルゴリズム 3.1
初期設定: $k = 0,\ m = 0,\ h^{(0)} = 1,\ L^{(0)} = I$ とする．初期点 $x^{(0)}$ を与え，(2.1) に基づいて $L^{(0)}$ から正の基底 $\tilde{V}_+^{(0)}$ を生成する．$V_+^{(0)} = \tilde{V}_+^{(0)}$, $F^{(0)} = \xi$ とする．ξ, ϵ, ϵ_1, ϵ_2, ϵ_3 は正の定数とする．$0 < \sigma < \tau < 1,\ 0 < \sigma < \frac{1}{2}$ となる定数 σ, τ を与える．

メッシュサイズが ϵ より小さくなるまで Step1 から Step10 を繰り返す．

Step1. メッシュサイズ $h^{(k)}$ を選ぶ．$i = 1$, $num = 0$ とし，$\eta^{(k)}$ は行列 $V_+^{(k)}$ の列の数とする．

点 $x^{(k)}$, メッシュサイズ $h^{(k)}$, 正の基底 $V_+^{(k)}$ を用いてグリッド $\mathcal{G}^{(m)}$ を定義する．

Step2. num が $\eta^{(k)}$ より小さいとき，以下の手順 ((a) と (b)) を繰り返す．

(a) $\{x^{(k)} + h^{(k)} v_{+i}^{(k)}\}$ $(i = 1, \ldots, n+1)$ を含むグリッド $\mathcal{G}^{(m)}$ における有限個の点での目的関数値を計算する．ただし，$v_{+i}^{(k)}$ は $V_+^{(m)}$ の第 i 列ベクトルを表す．

$(a\text{-}1)$ もし $f(x_{tempt}) < f(x^{(k)}) - (h^{(k)})^2$ を満たすような x_{tempt} が存在するならば, $x^{(k+1)} = x_{tempt}$, $h^{(k+1)} = \phi h^{(k)}$ とする. ただし ϕ は自然数とする. もし $h^{(k+1)} > F^{(m)}$ を満たすならば $h^{(k+1)} = h^{(k)}$ とおく. $V_+^{(k+1)} = V_+^{(k)}$, $k = k+1$, $num = 0$ とする.

$(a\text{-}2)$ そうでなければ, $num = num + 1$ とする.

(b) $i = i + 1$ とする.
もし $i > \eta^{(k)}$ ならば $i = 1$ とする.

Step3. $\tilde{x}^{(m)} = x^{(k)}$, $\tilde{h}^{(m)} = h^{(k)}$ とおく.

Step4. (3.10) または (3.11) より $\tilde{x}^{(m)}$ での $\hat{g}^{(m)}$ を計算する.

Step5. $d^{(m)} = -L^{(m)}\hat{g}^{(m)}$ とする.

Step6. 直線探索によって, 次式を満たす $d^{(m)}$ 方向のステップ幅 $\alpha^{(m)}$ を決める. (Wolfe条件 (3.1), (3.2) に対応する.)

$$\begin{aligned} f(\tilde{x}^{(m)} + \alpha^{(m)} d^{(m)}) &\leq f(\tilde{x}^{(m)}) - \sigma\alpha^{(m)} \hat{g}^{(m)^T} \hat{g}^{(m)}, \\ \tau \hat{g}^{(m)^T} d^{(m)} &\leq \hat{g}(\tilde{x}^{(m)} + \alpha^{(m)} d^{(m)})^T d^{(m)} \end{aligned}$$

Step7. $x^{(k+1)} = \tilde{x}^{(m)} + \alpha^{(m)} d^{(m)}$ とおく. (3.10) または (3.11) を用いて $x^{(k+1)}$ における勾配ベクトル g_{tempt} を計算し, $\hat{y}^{(m)} = g_{tempt} - \hat{g}^{(m)}$ とする. もし

$$||\hat{y}^{(m)}|| < \epsilon_1$$

を満たすならばアルゴリズムを停止する. そうでなければ Step8 へ行く.

Step8. もし

$$(\hat{g}^{(m)})^T \hat{y}^{(m)} > -\epsilon_2 ||\hat{g}^{(m)}|| ||\hat{y}^{(m)}||$$

を満たすならば, $L^{(m+1)} = L^{(m)}$, $V_+^{(m+1)} = V_+^{(m)}$ とし, Step10 に行く.

Step9. *Broyden* 公式族の分解型公式を用いて $L^{(m)}$ を更新する. 例えば, *BFGS* 公式の分解型公式の場合には

$$L^{(m+1)} = L^{(m)} + \sqrt{\frac{-\hat{g}^{(m)^T} \hat{y}^{(m)}}{\alpha^{(m)} \hat{g}^{(m)^T} \hat{g}^{(m)}}} \frac{s^{(m)} \hat{g}^{(m)^T}}{\hat{g}^{(m)^T} \hat{y}^{(m)}} + \frac{s^{(m)} \hat{y}^{(m)^T}}{\alpha^{(m)} \hat{g}^{(m)^T} \hat{y}^{(m)}}$$

を用いて $L^{(m)}$ を更新する. ただし, $s^{(m)} = x^{(k+1)} - \tilde{x}^{(m)}$ とする.

Step10. $h^{(k+1)} = \frac{1}{2} h^{(k)}$ とおき, $\tilde{V}_+^{(m+1)} = \left[L^{(m+1)}, -L^{(m+1)} e\right]$ で $\tilde{V}_+^{(m)}$ を更新する. $V_+^{(k+1)} = \tilde{V}_+^{(m+1)}$ とし, $F^{(m+1)} = \zeta F^{(m)}$ とおく. ただし $\zeta \in (0,1)$ である. $k = k+1$, $m = m+1$ とおく.

上記のアルゴリズムについてもう少し詳しく説明する. このアルゴリズムは外部反復と内部反復の 2 つから構成されている. 内部反復 (Step 2) では正の基底 $V_+^{(k)}$ の全てのメンバーに沿って新しい点を探す. num は探索に続けて失敗した回数である. $num = \eta^{(k)}$ のとき, 新しい点が見つけられなかったことを意味する. 内部反復が終了したら外部反復に移る. Step3 では中間情報 $\tilde{x}^{(m)}$, $\tilde{h}^{(m)}$ を与える. Step4-Step6 では探索方向を導き, Wolfe 条件を用いた直線探索によってステップ幅 $\alpha^{(m)}$ を求める. その後, $x^{(k)}$ を更新する. Step8 で $H^{(k+1)}$ の正定値性を確認している. 特に, このアルゴリズムでは有限差分法を用いていることにより常に $s^{(k)^T} y^{(k)}$ が正であることが言えないため, セーフガードを入れている. Step9 で $L^{(m)}$ を更新する. Step10 で $V_+^{(k)}$, $F^{(m)}$ および, m, k を更新する. 外部反復では, グリッドメッシュサイズ, 準ニュートン方向, ステップ幅を生成し, その後, L を更新する.

3.3 準ニュートン・パターンサーチ法（修正対称ランクワン公式）

本節では, パターンサーチ法への修正対称ランクワン公式 (3.5) の適用について紹介する．この公式は前述の Broyden 公式族には含まれないので，Wu and Sun [18] に倣う.

$$a = {y^{(k)}}^T H^{(k)} y^{(k)},\ b = {s^{(k)}}^T y^{(k)},\ c = {s^{(k)}}^T {H^{(k)}}^{-1} s^{(k)} \tag{3.17}$$

とおいたとき，修正対称ランクワン公式 (3.5) は以下のような分解型公式で表現できる.

$$H^{(k)} = L^{(k)} {L^{(k)}}^T,$$

$$L^{(k+1)} = \sqrt{\theta}\left[I + u^{(k)} {v^{(k)}}^T\right] L^{(k)}. \tag{3.18}$$

ただし

$$u^{(k)} = \mu(s^{(k)} - \theta H^{(k)} y^{(k)}), \tag{3.19}$$

$$v^{(k)} = \frac{B^{(k)} s^{(k)} - \theta y^{(k)}}{\theta}, \tag{3.20}$$

$$\mu = \frac{-\theta \pm \sqrt{\frac{c\theta - b\theta^2}{b - a\theta}}}{c - 2b\theta + a\theta^2} \tag{3.21}$$

である．ここで, (3.20) で定義された $v^{(k)}$ に対して

$$\omega^{(k)} = {L^{(k)}}^T v^{(k)} \tag{3.22}$$

とおいたとき, (3.19) の $u^{(k)}$ は次のようになる

$$u^{(k)} = \theta \mu L^{(k)} \omega^{(k)}$$

したがって (3.18) は

$$L^{(k+1)} = \sqrt{\theta} L^{(k)} \left[I + \theta \mu \omega^{(k)} {\omega^{(k)}}^T\right] \tag{3.23}$$

となる. また, (3.22) は以下のように書き換えられる.

$$\begin{aligned} \omega^{(k)} &= {L^{(k)}}^T v^{(k)} \\ &= L^{(k)T} \left(\frac{B^{(k)} s^{(k)} - \theta y^{(k)}}{\theta} \right) \\ &= L^{(k)T} \left(\frac{-\alpha g^{(k)} - \theta y^{(k)}}{\theta} \right) \\ &= -\left[\hat{y}^{(k)} + \left(\frac{\alpha}{\theta} \hat{g}^{(k)} \right) \right] \end{aligned} \tag{3.24}$$

式変形 (3.13), (3.14), (3.15) の場合と同様にして，(3.17) の a, b, c を (3.12) と $\hat{g}^{(k)} = {L^{(k)}}^T g^{(k)}$ を用いて以下のように書き換える.

$$\begin{aligned} a &= (\hat{y}^{(k)})^T \hat{y}^{(k)}, \\ b &= -\alpha (\hat{g}^{(k)})^T \hat{y}^{(k)}, \\ c &= \alpha^2 (\hat{g}^{(k)})^T \hat{g}^{(k)} \end{aligned}$$

パラメータ θ の決め方として，Osborne and Sun [13] は行列 $\tilde{H}_{k+1} = H^{(k)^{-\frac{1}{2}}} H^{(k+1)} H^{(k)^{-\frac{1}{2}}}$ の条件数が最小になるような θ として，

$$\theta_1 = \frac{c}{b} - \sqrt{\left(\frac{c}{b}\right)^2 - \frac{c}{a}}, \quad \theta_2 = \frac{c}{b} + \sqrt{\left(\frac{c}{b}\right)^2 - \frac{c}{a}} \tag{3.25}$$

を与えた．

以上に基づいて, Wu and Sun [18] は修正対称ランクワン公式を用いたパターンサーチ法として次のアルゴリズムを提案した．

アルゴリズム 3.2
初期設定: $k = 0,\ m = 0,\ h^{(0)} = 1,\ L^{(0)} = I$ とする．初期点 $x^{(0)}$ を与え,(2.1) に基づいて $L^{(0)}$ から正の基底 $\tilde{V}_{+}^{(0)}$ を生成する．$V_{+}^{(0)} = \tilde{V}_{+}^{(0)}$, $F^{(0)} = \xi$ とする．ξ, ϵ, ϵ_1, ϵ_2, ϵ_3 は正の定数とする．$\sigma \in (0, \frac{1}{2})$ とする．

メッシュサイズが ϵ より小さくなるまで Step1 から Step13 を繰り返す．

Step1. メッシュサイズ $h^{(k)}$ を選ぶ．$i = 1$, $num = 0$ とし, $\eta^{(k)}$ は行列 $V_{+}^{(k)}$ の列の数とする．

点 $x^{(k)}$, メッシュサイズ $h^{(k)}$, 正の基底 $V_{+}^{(k)}$ を用いてグリッド $\mathcal{G}^{(m)}$ を定義する．

Step2. num が $\eta^{(k)}$ より小さいとき, 以下の手順 ((*a*) と (*b*)) を繰り返す．

(*a*) $\{x^{(k)} + h^{(k)} v_{+i}^{(k)}\}$ $(i = 1, \ldots, n+1)$ を含むグリッド $\mathcal{G}^{(m)}$ における有限個の点での目的関数値を計算する．ただし, $v_{+i}^{(k)}$ は $V_{+}^{(k)}$ の第 i 列ベクトルを表わす．

(*a*-1) もし $f(x_{tempt}) < f(x^{(k)}) - (h^{(k)})^2$ を満たすような x_{tempt} が存在するならば, $x^{(k+1)} = x_{tempt}$, $h^{(k+1)} = \phi h^{(k)}$ とする．ただし ϕ は自然数とする．もし $h^{(k+1)} > F^{(m)}$ を満たすならば $h^{(k+1)} = h^{(k)}$ とおく．$V_{+}^{(k+1)} = V_{+}^{(k)}$, $k = k + 1$, $num = 0$ とする．

(*a*-2) そうでなければ, $num = num + 1$ とする．

(*b*) $i = i + 1$ とする．
もし $i > \eta^{(k)}$ ならば $i = 1$ とする．

Step3. $\tilde{x}^{(m)} = x^{(k)}$, $\tilde{h}^{(m)} = h^{(k)}$ とおく．

Step4. (3.10) または (3.11) より $\tilde{x}^{(m)}$ での $\hat{g}^{(m)}$ を計算する．

Step5. $d^{(m)} = -L^{(m)} \hat{g}^{(m)}$ とする．

Step6. $d^{(m)}$ 方向に沿った直線探索を実行して, $x^{(k+1)}$ を探す．ここでは次式を満たす直線探索によって, $d^{(m)}$ 方向のステップ幅 $\alpha^{(m)}$ を決める．

$$f(\tilde{x}^{(m)} + \alpha^{(m)} d^{(m)}) \leq f(\tilde{x}^{(m)}) - \sigma \alpha \hat{g}^{(m)^T} \hat{g}^{(m)}$$

Step7. $x^{(k+1)} = \tilde{x}^{(m)} + \alpha^{(m)} d^{(m)}$ とおく．(3.10) または (3.11) を用いて $x^{(k+1)}$ における勾配ベクトル g_{tempt} を計算し, $\hat{y}^{(m)} = g_{tempt} - \hat{g}^{(m)}$ とする．もし

$$||\hat{y}^{(m)}|| < \epsilon_1$$

を満たすならばアルゴリズムを停止する．そうでなければ Step8 へ行く．

Step8. もし

$$(\hat{g}^{(m)})^T \hat{y}^{(m)} > -\epsilon ||\hat{g}^{(m)}|| ||\hat{y}^{(m)}||$$

を満たすならば, $L^{(m+1)} = L^{(m)}$, $V_{+}^{(m+1)} = V_{+}^{(m)}$ とし, Step13 に行く．

Step9. もし

$$-(\alpha\hat{g}^{(m)}+\hat{y}^{(m)})^T\hat{y}^{(m)} > \epsilon_2||\alpha\hat{g}^{(m)}+\hat{y}^{(m)}||||\hat{y}^{(m)}||$$

を満たすならば, $\theta=1$ とし Step12 へ行く. そうでなければ Step10 へ行く.

Step10. $a=(\hat{y}^{(m)})^T\hat{y}^{(m)}$, $b=-\alpha(\hat{g}^{(m)})^T\hat{y}^{(m)}$, $\gamma=\frac{a}{b}$ とする. もし

$$||L^{(m)}(\hat{y}^{(m)}+\alpha\gamma\hat{g}^{(m)})|| \le \epsilon_3$$

を満たすならば,

$$L^{(m+1)}=\frac{1}{\sqrt{\gamma}}L^{(m)}$$

とし Step13 へ行く. そうでなければ, (3.25) を用いて θ_1, θ_2 を計算する.

Step11. 次の不等式が成り立つとき $\theta=\theta_1$ とおき, 成り立たないときは $\theta=\theta_2$ とおく.

$$\mathrm{Tr}(L^{(k+1)}L^{(k+1)^T})_{\theta=\theta_1} < \mathrm{Tr}(L^{(k+1)}L^{(k+1)^T})_{\theta=\theta_2} \tag{3.26}$$

Step12. (3.21), (3.24) を用いて μ, $\omega^{(m)}$ を計算し, 更新公式 (3.23) を用いて $L^{(m)}$ を更新して $L^{(m+1)}$ を生成する.

Step13. $h^{(k+1)}=\frac{1}{2}h^{(k)}$ とおき, $\tilde{V}_+^{(m+1)}=\left[L^{(m+1)},-L^{(m+1)}e\right]$ で $\tilde{V}_+^{(m)}$ を更新する. $V_+^{(k+1)}=\tilde{V}_+^{(m+1)}$ とし, $F^{(m+1)}=\zeta F^{(m)}$ とおく. ただし, $\zeta\in(0,1)$ とする. $k=k+1$, $m=m+1$ とおく.

4 数値実験

本節では, BFGS 公式 (3.16) および修正対称ランクワン公式 (3.23) を用いた準ニュートン・パターンサーチ法の数値実験結果について報告する. 特に, BFGS 公式に初期サイジングを使用した以下のアルゴリズム 4.1 を提案する. なおアルゴリズム 3.2 では, パラメータ θ の選択範囲に条件 (3.6) が課されているため, こうした初期サイジングが活用できないことを強調しておく.

アルゴリズム 4.1
初期設定: $k=0$, $m=0$, $h^{(0)}=1$, $L^{(0)}=I$ とする. 初期点 $x^{(0)}$ を与え, (2.1) に基づいて $L^{(0)}$ から正の基底 $\tilde{V}_+^{(0)}$ を作る. $V_+^{(0)}=\tilde{V}_+^{(0)}$, $F^{(0)}=\xi$ とする. ξ, ϵ, ϵ_1, ϵ_2, ϵ_3 は正の定数とする. $0<\sigma<\tau<1$, $0<\sigma<\frac{1}{2}$ となる定数 σ, τ を与える.

メッシュサイズが ϵ より小さくなるまで Step1 から Step11 を繰り返す.

Step1. メッシュサイズ $h^{(k)}$ を選ぶ. $i=1$, $num=0$ とし, $\eta^{(k)}$ は行列 $V_+^{(k)}$ の列の数とする.

点 $x^{(k)}$, メッシュサイズ $h^{(k)}$, 正の基底 $V_+^{(k)}$ を用いてグリッド $\mathcal{G}^{(m)}$ を定義する.

Step2. num が $\eta^{(k)}$ より小さいとき, 以下の手順 ((a) と (b)) を繰り返す.

(a) $\{x^{(k)}+h^{(k)}v_{+i}^{(k)}\}$ $(i=1,\dots,n+1)$ を含むグリッド $\mathcal{G}^{(m)}$ における有限個の点での目的関数値を計算する. ただし, $v_{+i}^{(k)}$ は $V_+^{(k)}$ の第 i 列ベクトルを表す.

$(a$-1) もし $f(x_{tempt})<f(x^{(k)})-(h^{(k)})^2$ を満たすような x_{tempt} が存在するならば, $x^{(k+1)}=x_{tempt}$, $h^{(k+1)}=\phi h^{(k)}$ とする. ただし ϕ は自然数とする. もし $h^{(k+1)}>F^{(m)}$ を満たすならば $h^{(k+1)}=h^{(k)}$ とおく. $V_+^{(k+1)}=V_+^{(k)}$, $k=k+1$, $num=0$ とする.

$(a$-2) そうでなければ, $num=num+1$ とする.

(b) $i=i+1$ とする.
もし $i>\eta^{(k)}$ ならば $i=1$ とする.

Step3. $\tilde{x}^{(m)} = x^{(k)}$, $\tilde{h}^{(m)} = h^{(k)}$ とおく.

Step4. (3.10) または (3.11) より $\tilde{x}^{(m)}$ での $\hat{g}^{(m)}$ を計算する.

Step5. $d^{(m)} = -L^{(m)}\hat{g}^{(m)}$ とする.

Step6. 直線探索によって, 次式を満たす $d^{(m)}$ 方向のステップ幅 $\alpha^{(m)}$ を決める. (Wolfe 条件 (3.1), (3.2) に対応する.)

$$\begin{aligned} f(\tilde{x}^{(m)} + \alpha^{(m)} d^{(m)}) &\le f(\tilde{x}^{(m)}) - \sigma\alpha^{(m)} \hat{g}^{(m)^T} \hat{g}^{(m)}, \\ \tau \hat{g}^{(m)^T} d^{(m)} &\le \hat{g}(\tilde{x}^{(m)} + \alpha^{(m)} d^{(m)})^T d^{(m)} \end{aligned}$$

Step7. $x^{(k+1)} = \tilde{x}^{(m)} + \alpha^{(m)} d^{(m)}$ とおく. (3.10) または (3.11) を用いて $x^{(k+1)}$ における勾配ベクトル g_{tempt} を計算し, $\hat{y}^{(m)} = g_{tempt} - \hat{g}^{(m)}$ とする. もし

$$||\hat{y}^{(m)}|| \le \epsilon_1$$

を満たすならばアルゴリズムを停止する. そうでなければ Step8 へ行く.

Step8. もし

$$(\hat{g}^{(m)})^T \hat{y}^{(m)} > -\epsilon_2 ||\hat{g}^{(m)}|| ||\hat{y}^{(m)}||$$

を満たすならば, $L^{(m+1)} = L^{(m)}$, $V_+^{(m+1)} = V_+^{(m)}$ とし, Step11 に行く.

Step9. $m = 0$ のとき, $\gamma_0 = \frac{-\alpha^{(0)}(\hat{g}^{(0)})^T \hat{y}^{(0)}}{(\hat{y}^{(0)})^T \hat{y}^{(0)}}$, または $\gamma_0 = \frac{-\alpha^{(0)}(\hat{g}^{(0)})^T \hat{g}^{(0)}}{(\hat{g}^{(0)})^T \hat{y}^{(0)}}$ を計算する. $\gamma_0 > 0$ のとき, 初期サイジングを実行して

$$L^{(0)} = \sqrt{\gamma_0} L^{(0)}$$

とおく.

Step10. 分解型 BFGS 公式

$$L^{(m+1)} = L^{(m)} + \sqrt{\frac{-\hat{g}^{(m)^T} \hat{y}^{(m)}}{\alpha^{(m)} \hat{g}^{(m)^T} \hat{g}^{(m)}}} \frac{s^{(m)} \hat{g}^{(m)^T}}{\hat{g}^{(m)^T} \hat{y}^{(m)}} + \frac{s^{(m)} \hat{y}^{(m)^T}}{\alpha^{(m)} \hat{g}^{(m)^T} \hat{y}^{(m)}}$$

を用いて $L^{(m)}$ を更新する. ただし, $s^{(m)} = x^{(k+1)} - \tilde{x}^{(m)}$ とする.

Step11. $h^{(k+1)} = \frac{1}{2} h^{(k)}$ とおき, $\tilde{V}_+^{(m+1)} = [L^{(m+1)}, -L^{(m+1)} e]$ で $\tilde{V}_+^{(m)}$ を更新する. $V_+^{(k+1)} = \tilde{V}_+^{(m+1)}$ とし, $F^{(m+1)} = \zeta F^{(m)}$ とおく. ただし $\zeta \in (0,1)$ である. $k = k+1$, $m = m+1$ とおく.

本論文の数値実験で扱ったテスト問題およびその初期点を表 1 にまとめておく. これらのテスト問題は Moré, Garbow and Hillstrom [11] から引用した. 各々のテスト問題に対して, Wu and Sun [18] によって提案されたアルゴリズム 3.2 の数値実験結果を表 2 に与える. ただし, この表での N_t は反復回数, N_f は関数評価回数, f_t は x_t における関数値, f^* は最小解 x^* における最小値とする. ただし, 反復回数はアルゴリズムにおいて点 x が更新された回数のことであり, 関数評価回数は目的関数とその勾配ベクトルが呼び出された回数を意味する. (勾配ベクトルの成分の差分近似は回数に入れていない。) 同様にアルゴリズム 3.1(BFGS) の数値実験結果を表 3 に, 初期サイジングを使用したアルゴリズム 4.1 の数値実験結果を表 4, 5 にそれぞれ与える.

表 2 から表 5 において, アルゴリズム 3.2 は問題 TF.5, TF.9, TF.12, TF.15, TF.21 で, アルゴリズム 3.1(BFGS) は問題 TF.2, TF.15, TF.18, TF.21 で, アルゴリズム 4.1 は問題 TF.12, TF.15 でそれぞれ失敗している. アルゴリズム 3.2 に比べてアルゴリズム 3.1(BFGS) の計算効率の方がよい場合が多いことがわかるが, 逆のケースも散見される. ただし, アルゴリズム 3.2 ではパラメータ θ の計算が必要なので, 他の解法に比べて計算時間がかかった. 一方、初期サイジングを使用したアルゴリズム 4.1 はいずれのサイジングパラメータの場合でも比較的計算効率が安定していると思われる. 初期サイジングの効果が現れているといえる. しかしながら, テスト問題によっては γ_0 がゼロに近い値になることもあり, 計算効率が悪くなる場合もあった.

表 1: テスト問題一覧（問題名, 次元, 初期点）

	名前	次元	初期点
TF.1	Beale	2	$x^{(0)} = (1,\ 1)$
TF.2	Rosenbrock	2	$x^{(0)} = (-1.2,\ 1)$
TF.3	Extended Powell	4	$x^{(0)} = (3,\ -1,\ 0,\ 1)$
TF.4	Freudenstein and Roth function	2	$x^{(0)} = (0.5,\ -2)$
TF.5	Jennrich and Sampson function	2	$x^{(0)} = (0.3,\ 0.4)$
TF.6	Brown Badly Scaled function	2	$x^{(0)} = (1,\ 1)$
TF.7	Broyden Tridiagonal function	10	$x^{(0)} = (-1,\ \ldots,\ -1)$
TF.8	Brown and Dennis function	4	$x^{(0)} = (25,\ 5,\ -5,\ -1)$
TF.9	Wood function	4	$x^{(0)} = (-3,\ -1,\ -3,\ -1)$
TF.10	Tridia	50	$x^{(0)} = (1,\ \ldots,\ 1)$
TF.11	Box three-dimensional function	3	$x^{(0)} = (0,\ 10,\ 20)$
TF.12	Powell badly scaled function	2	$x^{(0)} = (0,\ 1)$
TF.13	Bard function	3	$x^{(0)} = (1,\ 1,\ 1)$
TF.14	Gaussian function	3	$x^{(0)} = (0.4,\ 1,\ 0)$
TF.15	Meyer function	3	$x^{(0)} = (0.02,\ 4000,\ 250)$
TF.16	Powell singular function	4	$x^{(0)} = (3,\ -1,\ 0,\ 1)$
TF.17	Kowalik and Osborne function	4	$x^{(0)} = (0.25,\ 0.39,\ 0.415,\ 0.39)$
TF.18	Extended Rosenbrock function	50	$x^{(0)} = (-1.2,\ 1,\ \ldots,\ -1.2,\ 1)$
		100	$x^{(0)} = (-1.2,\ 1,\ \ldots,\ -1.2,\ 1)$
		1000	$x^{(0)} = (-1.2,\ 1,\ \ldots,\ -1.2,\ 1)$
TF.19	Penalty function 1	4	$x^{(0)} = (1,\ 2,\ 3,\ 4)$
		10	$x^{(0)} = (1,\ 2,\ \ldots,\ 10)$
TF.20	Penalty function 2	4	$x^{(0)} = (1/2,\ \ldots,\ 1/2)$
		10	$x^{(0)} = (1/2,\ \ldots,\ 1/2)$
TF.21	Extended Wood function	20	$x^{(0)} = (-3,\ -1,\ -3,\ -1,\ \ldots,\ -3,\ -1)$
		100	$x^{(0)} = (-3,\ -1,\ -3,\ -1,\ \ldots,\ -3,\ -1)$
		1000	$x^{(0)} = (-3,\ -1,\ -3,\ -1,\ \ldots,\ -3,\ -1)$
TF.22	Linear function-rank 1	5	$x^{(0)} = (1,\ 1,\ 1,\ 1,\ 1)$
TF.23	Discrete boundary value function	5	$x^{(0)} = t_i(t_i - 1),\ t_i = \frac{i}{n+1}$
		10	$x^{(0)} = t_i(t_i - 1),\ t_i = \frac{i}{n+1}$
TF.24	Variably dimensioned function	4	$x^{(0)} = 1 - \frac{i}{n}$

表 2: アルゴリズム 3.2 の数値実験結果

Function	Dimension	N_t	N_f	f_t	f^*
TF.1	$N = 2$	25	73	4.48×10^{-19}	0
TF.2	$N = 2$	45	111	7.32×10^{-24}	0
TF.3	$N = 4$	81	283	8.92×10^{-9}	0
TF.4	$N = 2$	21	63	48.98425	48.9842
TF.5	$N = 2$	27	72	725.7843	125.279
TF.6	$N = 2$	23	59	6.60×10^{-27}	0
TF.7	$N = 10$	102	745	6.90×10^{-15}	0
TF.8	$N = 4$	82	284	85823.2	85822.2
TF.9	$N = 4$	85	287	1.2739	0
TF.10	$N = 50$	2	101	0	0
TF.11	$N = 3$	47	167	2.21×10^{-23}	0
TF.12	$N = 2$	5	17	0.1351	0
TF.13	$N = 3$	57	201	0.008241	8.21×10^{-3}
TF.14	$N = 3$	2	12	2.34×10^{-6}	1.12×10^{-8}
TF.15	$N = 3$	61	196	111980.8	87.94
TF.16	$N = 4$	81	283	8.91×10^{-9}	0
TF.17	$N = 4$	1	8	0.0053	1.027×10^{-3}
TF.18	$N = 50$	72	2843	9.82×10^{-5}	0
	$N = 100$	74	5645	1.12×10^{-4}	0
	$N = 1000$	73	56044	1.22×10^{-4}	0
TF.19	$N = 4$	71	266	5.61×10^{-5}	2.25×10^{-5}
	$N = 10$	84	634	0.000129	7.09×10^{-5}
TF.20	$N = 4$	74	269	1.12×10^{-5}	9.38×10^{-6}
	$N = 10$	74	605	0.000296	2.94×10^{-4}
TF.21	$N = 20$	100	1191	2.858145	0
	$N = 100$	87	5658	4.469738	0
	$N = 1000$	95	56066	2.930536	0
TF.22	$N = 5$	5	32	2.142857	2.142857
TF.23	$N = 5$	26	273	5.59×10^{-27}	0
	$N = 10$	64	684	1.58×10^{-10}	0
TF.24	$N = 4$	89	284	7.20×10^{-4}	0

表 3: アルゴリズム 3.1(BFGS) の数値実験結果

Function	Dimension	N_t	N_f	f_t	f^*
TF.1	$N=2$	28	79	2.19×10^{-20}	0
TF.2	$N=2$	3	9	0.516117	0
TF.3	$N=4$	71	266	2.81×10^{-9}	0
TF.4	$N=2$	18	57	48.98425	48.98425
TF.5	$N=2$	20	59	125.2793	125.279
TF.6	$N=2$	24	66	3.99×10^{-28}	0
TF.7	$N=10$	109	752	5.70×10^{-7}	0
TF.8	$N=4$	55	216	85822.2	85822.2
TF.9	$N=4$	74	276	1.51×10^{-6}	0
TF.10	$N=50$	2	101	0	0
TF.11	$N=3$	40	165	1.83×10^{-20}	0
TF.12	$N=2$	56	139	1.30×10^{-5}	0
TF.13	$N=3$	37	152	0.008215	8.21×10^{-3}
TF.14	$N=3$	2	12	2.34×10^{-6}	1.12×10^{-8}
TF.15	$N=3$	71	215	121710.6	87.94
TF.16	$N=4$	71	266	2.81×10^{-9}	0
TF.17	$N=4$	1	8	0.0053	1.027×10^{-3}
TF.18	$N=50$	108	2879	1.4765	0
	$N=100$	105	5676	3.2313	0
	$N=1000$	194	56165	11.834	0
TF.19	$N=4$	58	260	5.78×10^{-5}	2.25×10^{-5}
	$N=10$	72	603	7.09×10^{-5}	7.09×10^{-5}
TF.20	$N=4$	63	258	0.000296	9.38×10^{-6}
	$N=10$	85	616	0.000836	2.94×10^{-4}
TF.21	$N=20$	120	1250	16.14394	0
	$N=100$	154	5924	5991.369	0
	$N=1000$	176	58416	43730.66	0
TF.22	$N=5$	5	32	2.142857	2.142857
TF.23	$N=5$	12	168	1.66×10^{-18}	0
	$N=10$	36	519	9.36×10^{-21}	0
TF.24	$N=4$	69	230	4.00×10^{-19}	0

表 4: アルゴリズム 4.1($\gamma_0 = \frac{-\alpha^{(0)}(\hat{g}^{(0)})^T\hat{y}^{(0)}}{(\hat{y}^{(0)})^T\hat{y}^{(0)}}$) の数値実験結果

Function	Dimension	N_t	N_f	f_t	f^*
TF.1	$N=2$	26	74	8.54×10^{-19}	0
TF.2	$N=2$	42	108	2.06×10^{-23}	0
TF.3	$N=4$	71	266	2.81×10^{-9}	0
TF.4	$N=2$	19	61	48.98425	48.98425
TF.5	$N=2$	19	55	125.279	125.279
TF.6	$N=2$	24	66	3.99×10^{-28}	0
TF.7	$N=10$	32	492	6.70×10^{-19}	0
TF.8	$N=4$	55	216	85822.2	85822.2
TF.9	$N=4$	74	276	1.51×10^{-6}	0
TF.10	$N=50$	2	101	0	0
TF.11	$N=3$	40	165	1.83×10^{-20}	0
TF.12	$N=2$	5	17	0.1351	0
TF.13	$N=3$	37	152	0.008241	8.21×10^{-3}
TF.14	$N=3$	2	12	2.34×10^{-6}	1.12×10^{-8}
TF.15	$N=3$	71	217	123337.0	87.94
TF.16	$N=4$	71	266	2.81×10^{-9}	0
TF.17	$N=4$	1	8	0.0053	1.027×10^{-3}
TF.18	$N=50$	87	2759	4.52×10^{-4}	0
	$N=100$	84	5456	6.61×10^{-4}	0
	$N=1000$	105	54077	7.15×10^{-3}	0
TF.19	$N=4$	49	251	2.25×10^{-5}	2.25×10^{-5}
	$N=10$	152	702	7.09×10^{-5}	7.09×10^{-5}
TF.20	$N=4$	46	200	9.38×10^{-6}	9.38×10^{-6}
	$N=10$	80	611	0.000295	2.94×10^{-4}
TF.21	$N=20$	97	1227	0.001522	0
	$N=100$	88	5858	0.004237	0
	$N=1000$	83	58053	0.024979	0
TF.22	$N=5$	5	32	2.142857	2.142857
TF.23	$N=4$	71	266	5.61×10^{-5}	0
	$N=10$	84	634	0.000129	0
TF.24	$N=4$	1	8	0.0053	0

表 5: アルゴリズム 4.1($\gamma_0 = \frac{-\alpha^{(0)}(\hat{g}^{(0)})^T\hat{g}^{(0)}}{(\hat{g}^{(0)})^T\hat{y}^{(0)}}$) の数値実験結果

Function	Dimension	N_t	N_f	f_t	f^*
TF.1	$N=2$	26	74	8.54×10^{-19}	0
TF.2	$N=2$	40	106	3.21×10^{-23}	0
TF.3	$N=4$	71	266	2.81×10^{-9}	0
TF.4	$N=2$	19	61	48.98425	48.98425
TF.5	$N=2$	19	55	125.279	125.279
TF.6	$N=2$	24	66	3.99×10^{-28}	0
TF.7	$N=10$	42	525	1.01×10^{-20}	0
TF.8	$N=4$	55	216	85822.2	85822.2
TF.9	$N=4$	74	276	1.51×10^{-6}	0
TF.10	$N=50$	2	101	0	0
TF.11	$N=3$	40	165	1.83×10^{-20}	0
TF.12	$N=2$	5	17	0.1351	0
TF.13	$N=3$	37	152	0.008241	8.21×10^{-3}
TF.14	$N=3$	2	12	2.34×10^{-6}	1.12×10^{-8}
TF.15	$N=3$	71	217	123337.0	87.94
TF.16	$N=4$	71	266	2.81×10^{-9}	0
TF.17	$N=4$	1	8	0.0053	1.027×10^{-3}
TF.18	$N=50$	78	2750	1.25×10^{-3}	0
	$N=100$	80	5452	2.55×10^{-3}	0
	$N=1000$	103	54075	2.57×10^{-2}	0
TF.19	$N=4$	49	251	2.25×10^{-5}	2.25×10^{-5}
	$N=10$	152	702	7.09×10^{-5}	7.09×10^{-5}
TF.20	$N=4$	53	248	9.38×10^{-6}	9.38×10^{-6}
	$N=10$	78	609	0.000294	2.94×10^{-4}
TF.21	$N=20$	98	1228	0.001418	0
	$N=100$	88	5858	0.004509	0
	$N=1000$	90	58060	0.030905	0
TF.22	$N=5$	5	32	2.142857	2.142857
TF.23	$N=4$	71	266	5.61×10^{-5}	0
	$N=10$	84	634	0.000129	0
TF.24	$N=4$	1	8	0.0053	0

5 まとめ

本論文では, Coope and Price [6] が提案したパターンサーチ法の１つである「グリッドに基づいた方法」と, 準ニュートン法の代表的な更新公式を含む Broyden 公式族を組み合わせた解法を提案した. そして, BFGS 公式, 初期サイジング付き BFGS 公式、修正対称ランクワン公式をそれぞれ用いたパターンサーチ法について数値実験を行い, 計算効率を比較した. 初期サイジング付き BFGS 公式を用いた場合が, 比較的安定しているように思われる. 今後の課題としては, Broyden 公式族に含まれる他のメンバーを用いたパターンサーチ法についても数値実験を行っていく必要がある. また, Oren and Luenberger [12] 流のサイジングを取り込んだ場合の数値的な挙動にも興味がある.

謝辞 本研究の一部は日本学術振興会科学研究費補助金・基盤研究 (C)(25330030) の支援を受けて行われた.

参考文献

[1] ASNOP 研究会, パソコン FORTRAN 版 非線形最適化プログラミング, 日刊工業新聞社, 1991.

[2] K. W. Brodlie, A. R Gourlay and J. Greenstadt, Rank-one and rank-two corrections to positive definite matrices expressed in product form, *Journal of the Institute of Mathematics and its Applications*, **11** (1973), pp.73-82.

[3] D. Byatt, I. D. Coope and C. J. Price, Conjugate grids for unconstrained optimizaation, *Computational Optimization and Applications*, **29** (2004), pp.49-68.

[4] A. R. Conn, K. Scheinberg and L. N. Vicente, *Introduction to Derivative-Free Optimization*, MPS-SIAM Series on Optimization, SIAM, Philadelphia, 2009.

[5] I. D. Coope and C. J. Price, A direct search conjugate directions algorithm for unconstrained minimization, *ANZIAM Journal* , **42** (2000), pp.C478-498.

[6] I. D. Coope and C. J. Price, On the convergence of grid-based methods for unconstrained optimization, *SIAM Journal on Optimization*, **11** (2001), pp.859-869.

[7] J. E. Dennis, Jr. and R. B. Schnabel, A new derivation of symmetric positive definite secant updates, in *Nonlinear Programing 4* , O.L. Mangasarian, R.R. Meyer and S.M. Robinson eds., Academic Press, 1981, pp.167-199

[8] J. E. Dennis, Jr. and V. Torczon, Direct search methods on parallel machines, *SIAM Journal on Optimization*, **1** (1991), pp.448-474.

[9] T. G. Kolda, R. M. Lewis and V. Torczon, Optimization by direct search: New perspectives on some classical and modern methods, *SIAM Review*, **45** (2003), pp. 385-482.

[10] J. コワリック, M. R. オスボーン（山本善之, 小山健夫 共訳）,「非線形最適化問題」, 培風館, 1970.

[11] J. J. Moré, B. S. Garbow and K. E. Hillstrom, Testing unconstrained optimization software, *ACM Transactions on Mathematical Software*, **7** (1981), pp.17-41.

[12] S. S. Oren and D. G. Luenberger, Self-scaling variable metric (SSVM) algorithms, Part 1: Criteria and sufficient conditions for scaling a class of algorithms, *Management Science*, **20** (1974), pp.845-862.

[13] M. R. Osborne and L. P. Sun, A new approach to symmetric rank-one updating, *IMA Journal of Numerical Analysis*, **19** (1999), pp.497-507.

[14] L. P. Sun, An approach to scaling symmetric rank-one update, *Pacific Journal of Optimization*, **2** (2006), pp.105-118.

[15] V. Torczon, On the convergence of pattern search algorithms, *SIAM Journal on Optimization*, **7** (1997), pp.1-25.

[16] T. Wu and L. P. Sun, A quasi-Newton based pattern search algorithm for unconstrained optimization, *Applied Mathematics and Computation*, **183** (2006), pp. 685-694.

[17] T. Wu and L. P. Sun, A filter-based pattern search method for unconstrained optimization, *Numerical Mathematics. A Journal of Chinese Universities(English Series)*, **15** (2006), pp.209-216.

[18] T. Wu and L. P. Sun, A new quasi-Newton pattern search method based on symmetric rank-one update for unconstrained optimization, *Computers and Mathematics with Applications*, **55** (2008), pp.1201-1214.

[19] 矢部博, 「最適化とその応用」, 数理工学社, 2006.

[20] 矢部博, 八巻直一, 「非線形計画法」, 応用数値計算ライブラリ, 朝倉書店, 1999.