数理解析研究所講究録
第 1879 巻 2014 年 125-133

信頼領域法を利用した非線形半正定値計画問題に対する主双対内点法

株式会社 NTT データ数理システム * 原田耕平 (Kouhei Harada)
NTT DATA Mathematical Systems Inc.
株式会社 NTT データ数理システム　山下浩 (Hiroshi Yamashita)
NTT DATA Mathematical Systems Inc.
東京理科大学　矢部博 (Hiroshi Yabe)
Tokyo University of Science

概要

本稿は，非線形半正定値計画問題に対する主双対内点法アルゴリズムの報告である．非線形計画問題に対する主双対内点法のアルゴリズムとしては，直線探索法に拠る方法と，信頼領域法の考え方を利用する方法が存在する．これらそれぞれに対して，非線形半正定値計画問題の拡張が存在するが，これらのアルゴリズム間の特徴差は，必ずしも非線形計画問題の場合と同じではない．本稿では，信頼領域法を利用した非線形半正定値計画問題に対する主双対内点法アルゴリズム自体をまず紹介し，さらに直線探索法を利用するアルゴリズムとの性質比較を試みる．

1 概要

本稿では，非線形半正定値計画問題に対する，主双対内点法アルゴリズムを比較報告する．(線形の) 半正定値計画問題に対する研究は以前より盛んであり，多数の成果が得られている．一方で，非線形半正定値計画問題に対する研究は，それに比べると数少ないものの，近年このクラスの問題群に対するアルゴリズムが幾つか報告されている．代表的な方法として，乗数法を拡張した [6]，部分問題として，線形化した SDP を解く [2, 1, 3, 4] などが挙げられる．[2, 1, 3] は SQP の拡張と見なす事ができ，[2] では局所超一次収束性が，[1] では大域的収束性が証明されている．両者は直線探索に基づく方法を利用しているが，[3] では，フィルター法の考え方を利用している．[4] は SLP の拡張と見なせるが，大域的収束性を示すために信頼領域法の考え方を利用している．

主双対内点法を非線形半正定値計画問題に拡張する方法としては，我々の知る限り [12] が初めての試みである．このアルゴリズムは直線探索法を利用するもので，局所的な性質については [10] に記されている．最近では，この方向性の研究として，[5, 8] なども提案されている．一方，同じ主双対内点法の枠組みでも，信頼領域法の考え方を利用する方法として [11] が提案されている．本稿では，このアルゴリズムの特徴，及び [12] との比較について説明する．

本稿の構成は以下の通りである．第 2 章では，本稿にて考察する非線形半正定値計画問題を定義し，その最適性の条件等，アルゴリズムの説明において必要な情報を記載する．第 3 章では，[11] に記載されているアルゴリズムを説明する．ここでは，専らその手続きに着目し，その正当性に関しては深入りしない．第 4 章では，アルゴリズムの幾つかのバリエーションに関して議論する．特に，非線形計画問題では有用であった方法が，必ずしも非線形半正定値計画問題では効果的とは限らない事に注意する．第 5 章では，

* 〒160-0016 東京都新宿区信濃町 35 番地　信濃町煉瓦館 1 階

直線探索を利用する方法 [12] と，信頼領域法の考え方を利用する方法 [11] との比較を行う．ここでも，非線形計画問題における性質が，非線形半正定値計画問題では担保されない事に注意する．

2 準備

本稿では，以下の非線形半正定値計画問題を考える．
Problem 1

$$\text{minimize} \quad f(x) \tag{1}$$
$$\text{subject to} \quad g(x)=0 \tag{2}$$
$$X(x) \succeq O \tag{3}$$

ここで， $x \in \mathbb{R}^n$ であり f, g, X は連続かつ十分滑らかで，それぞれ $f:\mathbb{R}^n \to \mathbb{R}$, $g:\mathbb{R}^n \to \mathbb{R}^m$, $X:\mathbb{R}^n \to \mathbb{S}^p$ であるとする．ただし， $\mathbb{S}^p$ は p 次の対称行列全体からなる集合である．また，本問題に対するラグランジュ関数は，以下で与えられる．

$$L(w)=f(x)-y^T g(x)-\langle X(x), Z\rangle$$

ここで， $w=(x,y,Z)$ であり， $y \in \mathbb{R}^m$, $Z \in \mathbb{S}^p$ はそれぞれ制約 (2), (3) に対するラグランジュ乗数である．さらに，本問題に対する最適性の条件（ KKT 条件）は，以下で与えられる [12].

$$r_0(w) \equiv \begin{pmatrix} \nabla_x L(w) \\ g(x) \\ X(x)Z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

本稿で紹介するアルゴリズムは，相補性条件 $X(x)Z=0$ の代わりに，パラメータ $\mu>0$ を導入して滑らかにした条件 $X(x)Z=\mu I$ を満たす点を， $\mu \to 0$ に対して逐次求める．この様に相補性条件を置き換えた KKT 条件を，以下 barrier KKT(BKKT) 条件と呼ぶ．

$$r(w) \equiv \begin{pmatrix} \nabla_x L(w) \\ g(x) \\ X(x)Z-\mu I \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

BKKT 条件を満たす点を求める際に，以下の探索方向とメリット関数を用いる．探索方向には，スケーリングされた BKKT 条件に対する Newton 方向と，$\nabla_{xx}L(w)$ を何らかの正定値行列 D に置き換えた方向（以下 SD 方向）の二つを用いる．スケーリングに HRVW/KSH/M 方向を利用した場合，これらは具体的には以下の形で求められる．

$$\begin{pmatrix} \nabla_{xx}L(w_k)+H_k & -A_0(x_k)^T \\ -A_0(x_k) & O \end{pmatrix} \begin{pmatrix} \Delta x_k^{NW} \\ \Delta y_k^{NW} \end{pmatrix} = \begin{pmatrix} -\nabla f(x_k)+A_0(x_k)^T y+\mu \mathcal{A}^*(x_k) X_k^{-1} \\ g(x_k) \end{pmatrix} \tag{4}$$

$$\Delta Z_k^{NW}=\mu X_k^{-1}-Z_k-\frac{1}{2}(X_k^{-1}\Delta X_k^{NW}Z+Z\Delta X_k^{NW}X_k^{-1}) \tag{5}$$

$$\begin{pmatrix} D+H_k & -A_0(x_k)^T \\ -A_0(x_k) & O \end{pmatrix} \begin{pmatrix} \Delta x_k^{SD} \\ \Delta y_k^{SD} \end{pmatrix} = \begin{pmatrix} -\nabla f(x_k)+A_0(x_k)^T y+\mu \mathcal{A}^*(x_k) X_k^{-1} \\ g(x_k) \end{pmatrix} \tag{6}$$

$$\Delta Z_k^{SD}=\mu X_k^{-1}-Z_k-\frac{1}{2}(X_k^{-1}\Delta X_k^{SD}Z+Z\Delta X_k^{SD}X_k^{-1}) \tag{7}$$

ただし， $A_0(x):\mathbb{R}^m \to \mathbb{R}^n, \mathcal{A}^*(x):\mathbb{S}^p \to \mathbb{R}^n$ は，それぞれ以下で定まる．

$$A_0(x)=\begin{pmatrix} \nabla g_1(x)^T \\ \vdots \\ \nabla g_m(x)^T \end{pmatrix}, \mathcal{A}(x)^* Z=\begin{pmatrix} \langle A_1(x), Z\rangle \\ \vdots \\ \langle A_n(x), Z\rangle \end{pmatrix}$$

ここで, $A_i \equiv \frac{\partial X(x)}{\partial x_i}$ である. また, H は以下の様に定まる.

$$H_{ij} = Tr\left(A_i(x)X^{-1}A_j(x)Z\right)$$

一方, メリット関数 F は, 以下の二つの関数 F_{BP}, F_{PD} を組み合わせて作られる. それぞれ, Primal barrier penalty function, Primal-dual barrier function と呼ぶ. ここで $u = (x, Z)$ と定める.

$$\begin{aligned}
F_{BP}(x) &\equiv f(x) - \mu \log(\det X) + \rho \|g(x)\|_1 \\
F_{PD}(u) &\equiv \langle X, Z\rangle - \mu \log(\det X \det Z) \\
F(u) &\equiv F_{BP}(x) + F_{PD}(u)
\end{aligned}$$

本アルゴリズムでは, メリット関数そのものだけでなく, これらの一次近似 F_l, 二次近似 F_q 並びにそれぞれの変化量 ΔF_l, ΔF_q も利用する. それらは具体的には以下の形で定められる. 特に, F_q は後述の内部反復において, モデル二次関数として機能する.

$$\begin{aligned}
F_l(u, \Delta u) &\equiv F(u) + \Delta F_l(u, \Delta u) \\
\Delta F_l(u, \Delta u) &\equiv \Delta F_{BPl}(x, \Delta x) + \Delta F_{PDl}(u, \Delta u) \\
\Delta F_{BPl}(x, \Delta x) &\equiv f(x)^T \Delta x - \mu Tr(X^{-1}\Delta X) + \rho(\|g(x) + A_0(x)\Delta x\|_1 - \|g(x)\|_1) \\
\Delta F_{PDl}(u, \Delta u) &\equiv Tr(\Delta X Z + X \Delta Z - \mu X^{-1}\Delta X - \mu Z^{-1}\Delta Z) \\
F_q(u, \Delta u) &\equiv F_l(u, \Delta u) + \frac{1}{2}(\Delta u)^T(\nabla_{xx}L(w) + H)\Delta u \\
\Delta F_q(u, \Delta u) &\equiv F_q(u, \Delta u) - F(u)
\end{aligned}$$

3 アルゴリズム

本稿にて紹介するアルゴリズムは, 外部反復と内部反復の二つの反復から構成される. 外部反復は [NLSDP] に対する Barrier KKT 条件 (以下 BKKT 条件) を満たす点を, 逐次的に求める操作であり, 適当な仮定の下でこの反復は, KKT 条件を満たす点へ大域的収束する [12]. 外部反復の具体的な手続きを以下に述べる.

外部反復

Step0. 十分小さな $\epsilon > 0$, 正の定数 $M_c > 0$, 単調減少列 $\{\mu_k\}, \mu_k \downarrow 0$ を与える. $k = 0$ とする.
Step1. $\|r(w_{k+1}, \mu_k)\| \le M_c \mu_k$ を満たす w_{k+1} を求める.
Step2. 終了条件 $\|r_0(w_{k+1})\| \le \epsilon$ を満たせば, 停止する.
Step3. $k = k + 1$ とし, Step1 に戻る.

一方の内部反復は, 外部反復の各反復において BKKT 条件を満たす点を求める操作である. 上述の手続きの中では Step1 に相当する. この内部反復において, 信頼領域法の考え方が利用される. 内部反復の具体的な手続きを以下に述べる.

内部反復

Step0. 正の定数 $M_c > 0, M_1 > 0, \mu > 0, \rho > 0, \gamma \in (0,1)$ を与える． $k = 0$ とする．

Step1. 終了条件 $\|r(w_{k+1}, \mu)\| \leq M_c\mu$ を満たせば，停止する．

Step2. 方程式系 (4)(5) より，探索方向 Δw_k^{NW} を，方程式系 (6)(7) より，探索方向 Δw_k^{SD} を求める． $\|\Delta u_k^{NW}\| \leq M_1 \|\Delta u_k^{SD}\|$ を満たさない場合，この関係を満たす様に (4) を修正する．

Step3. 以下の条件を満たす探索方向 s_k を求める．

$$\begin{aligned}
\Delta u_k &= \nu \Delta u_k^{SD} + (1-\nu)\Delta u_k^{NW},\ \nu \in [0,1] \\
s_k &= \alpha^*(u_k, \Delta u_k)\Delta u_k \\
\|s_k\| &= \delta_k \\
(1-\gamma)\lambda_{min} U(u_k) &\leq \lambda_{min} U(u_k + s_k) \\
\Delta F_q(u_k, s_k) &\leq \frac{1}{2} F_q(u_k, \alpha^*(u_k, \Delta u_k^{SD})\Delta u_k^{SD})
\end{aligned}$$

Step4. 以下に基づいて，信頼領域 δ_k を更新する．

$$\begin{aligned}
&\textit{If} \quad \Delta F(u_k, s_k) > \frac{1}{4}\Delta F_q(u_k, s_k),\ \textit{then} \quad \delta_{k+1} = \frac{1}{2}\delta_k \\
&\textit{ElseIf} \quad \Delta F(u_k, s_k) \leq \frac{3}{4}\Delta F_q(u_k, s_k),\ \textit{then} \quad \delta_{k+1} = 2\delta_k \\
&\textit{Otherwise} \quad \delta_{k+1} = \delta_k
\end{aligned}$$

Step5. 探索方向 s_k 分進める事によりメリット関数が降下するとき，即ち $\Delta F(u_k, s_k) < 0$ が満たされる場合，$u_{k+1} = u_k + s_k$, $y_{k+1} = y_k + \Delta y_k$ とする．そうでなければ，点を更新しない．即ち， $w_{k+1} = w_k$ である．

Step6. $k = k+1$ として， Step1 に戻る．

内部反復の Step3 における α^* は，点 u_k と方向 Δu_k が固定された際に，モデル二次関数 $F_q(u_k, \alpha\Delta u_k)$ が最小となる様な $\alpha \geq 0$ である．即ち，

(8)
$$F_q(u_k, \alpha^*(u_k, \Delta u_k)\Delta u_k) \leq F_q(u_k, \alpha\Delta u_k),\ \forall \alpha \geq 0$$

が成り立つ．ところで， $\nu = 1$ の場合， $s_k = \alpha^*(u_k, \Delta u_k^{SD})\Delta u_k^{SD}$ となる事から，少なくとも $\nu = 1$ とすれば， Step3 を満たす探索方向が存在する事に注意する．

この手続きを適用する事により，適当な仮定の下で，ある部分列 K と $\hat{u} = (\hat{x}, \hat{Z})$ が存在し，

(9)
$$\lim_{k \to +\infty,\, k \in K} \Delta u_k^{SD} = 0$$

(10)
$$\lim_{k \to +\infty,\, k \in K} \Delta u_k^{NW} = 0$$

(11)
$$\lim_{k \to +\infty,\, k \in K} u_k = \hat{u}$$

となる．

ところで， Δw_k^{NW} は，方程式系 (4)(5) の解であるから，以下の関係が成り立つ．

$$\begin{aligned}
\nabla f(x_k) + (\nabla_{xx} L(w_k) + H_k)\Delta x_k^{NW} - A_0(x_k)^T(y_k + \Delta y_k) - \mu \mathcal{A}^*(x_k)X_k^{-1} &= 0 \\
g(x_k) + A_0(x_k)\Delta x_k^{NW} &= 0 \\
\mu X_k^{-1} - Z_k - \frac{1}{2}(X_k^{-1}\Delta X_k^{NW} Z + Z\Delta X_k^{NW} X_k^{-1}) - \Delta Z_k^{NW} &= 0
\end{aligned}$$

(9)(10)(11) より，適当な仮定の下で， $y_k + \Delta y_k$ も $k \in K$ に関して収束し，その値を $\hat{y}$ とすれば，以下を得る．

$$\begin{aligned}
\nabla f(\hat{x}) - A_0(\hat{x})^T\hat{y} - \mu \mathcal{A}^*(\hat{x})X(\hat{x})^{-1} &= 0 \\
g(\hat{x}) &= 0 \\
\mu X(\hat{x})^{-1} - \hat{Z} &= 0
\end{aligned}$$

これにより，内部反復が BKKT 条件を満たす点を生成する事が確かめられる．

4 主双対バリア関数と Box 制約

BKKT 条件の中で，特に相補性条件 $X(x)Z = \mu I$ を満たすために導入された項が，主双対バリア関数 F_{PD} である．しかしながら，相補性条件を満たすという目的に対して，主双対バリア関数の導入は唯一の方策では無い．非線形計画問題に対する同種のアルゴリズムである [9, 13] においては，主双対バリア関数を導入する代わりに，Box 制約と呼ばれるアイデアを採用する事で，相補性条件を満たすことを試みている．

Box 制約のアイデアを，非線形半正定値計画問題に拡張した場合，内部反復におけるメリット関数は $F_{BPl}(x_k)$ に相当する部分のみから構成される．この場合，Δx_k のステップ α_{xk} と ΔZ_k のステップ α_{zk} は同じでは無い事に注意する．α_{zk} は，相補性条件が満たされるように，以下の Box 制約を満たすように定められる．

(12) $$c_{L_k} I \preceq X(x_k + \alpha_{xk}\Delta x_k)^{\frac{1}{2}}(Z_k + \alpha_{zk}\Delta Z_k)X(x_k + \alpha_{xk}\Delta x_k)^{\frac{1}{2}} \preceq c_{U_k} I$$

また，これを満たす様に α_{zk} が定められた後は，c_{L_k}, c_{U_k} を，以下を満たす様に更新する．

$$0 < c_{L_k} < \mu < c_{U_k}$$

本ルールに基づき α_{zk} を定めると，相補性条件が満たされる [11]．ただし，Box 制約 (12) を満たす様な，α_{zk} の決め方には，幾つかの方法が存在する．

我々は Box 制約を利用する方法について，二通りの実装を行い，これらと主双対バリア関数を用いる方法との比較実験を行った．これらの比較実験の詳細は，次章で述べる．本章では，各実装の方法について述べる．

4.1 方法 1

$$\begin{aligned}
c_L &= \min\{\frac{\mu}{M_L}, \lambda_{min}(X + \alpha_x \Delta X)\lambda_{min}(Z)\} \\
c_U &= \max\{\mu M_U, \lambda_{max}(X + \alpha_x \Delta X)\lambda_{max}(Z)\} \\
\alpha_z &= \min\left\{1 - \frac{c_L}{\lambda_{min}(X + \alpha_x \Delta X)\lambda_{min}(Z)}, \frac{c_U}{\lambda_{max}(X + \alpha_x \Delta X)\lambda_{max}(Z)} - 1, 1\right\}
\end{aligned}$$

但し，$M_L > 1, M_U > 1$ はそれぞれ所与の定数である．また，第一項，第二項は Box 外部への方向の場合のみ有効とする．Box の境界上であっても，Box 内部への方向である場合は，ステップ幅は 1 とする．さらに，境界上かつ Box 外部への方向の場合，ステップ幅は 0 とする．

方法 1 において計算すべき値は，$X + \alpha_x \Delta X, Z, Z^{-1}\Delta Z$ の三種の最大最小固有値なので，計算量は後述の方法 2 より少ない．一方で $\lambda_{min}(X + \alpha_x \Delta X)\lambda_{min}(Z), \lambda_{max}(X + \alpha_x \Delta X)\lambda_{max}(Z)$ を用いている事から Box 制約の評価としてはギャップが大きい．このため，実装上は $M_L = M_U$ を相当大きな値に設定する必要がある．

4.2 方法 2

$$
\begin{aligned}
c_L &= \min\{\frac{\mu}{M_L}, \lambda_{min}((X+\alpha_x\Delta X)Z)\} \\
c_U &= \min\{M_U, \lambda_{max}((X+\alpha_x\Delta X)Z)\} \\
\alpha_z &= \min\left\{-\frac{1}{\lambda_{min}(V_L^{-\frac{1}{2}}\Delta Z V_L^{-\frac{1}{2}})}, \frac{1}{\lambda_{max}(V_U^{-\frac{1}{2}}\Delta Z V_U^{-\frac{1}{2}})}, 1\right\} \\
V_L &= Z - c_L(X+\alpha_x\Delta X)^{-1} \\
V_U &= c_U(X+\alpha_x\Delta X)^{-1} - Z
\end{aligned}
$$

方法 1 と同じく，$M_L > 1, M_U > 1$ はそれぞれ所与の定数である．方法 2 は，方法 1 と比べると計算量が多くなるが，$\lambda_{min}((X+\alpha_x\Delta X)Z), \lambda_{max}((X+\alpha_x\Delta X)Z)$ を用いているため，Box 制約の評価としては，より厳格である．

4.3 実験結果

本節では，第 3 章で紹介したアルゴリズムに対して，相補性条件の取り扱いを変えた場合，どの様に性能差が生じるのか，各実装の実験結果を報告する．以下のテーブルにて p1, p2 と記載されているアルゴリズムは，それぞれ第 4 章の方法 1,2 に対応しており，Box 制約を利用しているものである．また，pd と記載されているアルゴリズムは第 3 章で紹介した元来のアルゴリズムであり，主双対バリア関数を利用している．性能比較の実験には，制御系の非線形半正定値計画問題のベンチマーク問題集である Compleib[7] の SOF H_2 problem を利用した．

Problem 2 (SOF H_2 problem)

$$
\begin{aligned}
\text{minimize} \quad & Tr(X) \\
\text{subject to} \quad & Q \succ O \\
& A(F)Q + QA(F)^T + B_1B_1^T \preceq O \\
& \begin{pmatrix} X & C(F)Q \\ QC(F)^T & Q \end{pmatrix} \succeq O
\end{aligned}
$$

$X \in \mathbb{R}^{n_z \times n_z}, F \in \mathbb{R}^{n_u \times n_y}, Q \in \mathbb{R}^{n_x \times n_x}$ が変数行列．X, Q は対称行列だが，F は非対称行列で正方とは限らない．$A \in \mathbb{R}^{n_x \times n_x}, B \in \mathbb{R}^{n_x \times n_u}, B_1 \in \mathbb{R}^{n_x \times n_w}, C \in \mathbb{R}^{n_y \times n_x}, C_1 \in \mathbb{R}^{n_z \times n_x}, D_{11} \in \mathbb{R}^{n_z \times n_w}, D_{12} \in \mathbb{R}^{n_z \times n_u}, D_{21} \in \mathbb{R}^{n_y \times n_w}$ が定数行列．$A(F), B(F), C(F), D(F)$ は次の様に定義される．

$$
\begin{aligned}
A(F) &= A + BFC \\
B(F) &= B_1 + BFD_{21} \\
C(F) &= C_1 + D_{12}FC \\
D(F) &= D_{11} + D_{12}FD_{21}
\end{aligned}
$$

本問題に対しては，初期値として内点許容解を得る事が容易ではない．何らかの解が得られたベンチマーク問題に対する結果を以下に示す．反復の終了条件は KKT 条件の残差が $1.0e-6$ 以下か否かで判定する．以下のテーブルにおける itr とは Newton 方程式の計算回数を意味する [*1]．また，time は計算全体に必要とした計算時間（秒）を意味する．問題に出現する 3 種の半正定値制約それぞれの次元は，$n_x, n_x, n_x + n_z$ により求められる．

[*1] μ の更新回数を意味するものではない事に注意

SOF H2 problem												
問題名	変数	n_x	n_y	n_u	n_w	n_z	itr(p1)	time(p1)	itr(p2)	time(p2)	itr(pd)	time(pd)
AC1	27	5	3	3	3	2	*	*	*	*	58	0.13
AC2	39	5	3	3	3	5	*	*	*	*	56	0.16
AC3	38	5	4	2	5	5	*	*	*	*	74	0.24
AC8	53	9	5	1	10	2	50	0.32	*	*	50	0.35
AC15	37	4	3	2	4	6	*	*	*	*	294	1.07
AC16	39	4	4	2	4	6	*	*	*	*	61	0.15
AC17	22	4	2	1	4	4	*	*	*	*	104	0.25
HE1	15	4	1	2	2	2	*	*	82	0.15	*	*
HE2	24	4	2	2	4	4	*	*	*	*	26	0.05
REA1	26	4	3	2	4	4	*	*	*	*	48	0.11
REA2	24	4	2	2	4	4	*	*	*	*	257	0.63
REA3	159	12	3	1	12	12	88	1.8	*	*	44	0.93
DIS1	88	8	4	4	1	8	*	*	37	0.31	36	0.29
DIS2	16	3	2	2	3	3	53	0.07	53	0.08	106	0.15
DIS3	58	6	4	4	6	6	*	*	*	*	198	0.9
DIS4	66	6	6	4	6	6	29	0.16	29	0.15	22	0.11
AGS	160	12	2	2	12	12	46	1.18	*	*	19	0.44
BDT1	96	11	3	3	1	6	*	*	*	*	65	0.92
EB1	59	10	1	1	2	2	40	0.29	41	0.3	32	0.22
EB2	59	10	1	1	2	2	84	0.62	77	0.57	*	*
PSM	49	7	3	2	2	5	172	0.67	104	0.4	48	0.17
NN2	7	2	1	1	2	2	20	0.02	20	0.02	17	0.02
NN11	157	16	5	3	3	3	*	*	*	*	40	1.32
NN15	20	3	2	2	1	4	29	0.04	29	0.04	66	0.15
NN16	62	8	4	4	8	4	*	*	*	*	277	2.52

これらを見る限り，非線形計画問題の場合と異なり， Box 制約を利用する方法は主双対バリア関数を利用する方法と比べて著しく性能が劣っている．

5 直線探索法との比較

本稿で紹介した，信頼領域法を用いる方法では，内部反復での大域的収束性を担保するために信頼領域法の考え方を利用したが，この方法とは別に直線探索法を利用する方法も考えられる [12]．非線形計画問題においても，内部反復に信頼領域法を利用する方法 [13] と，直線探索法を利用する方法 [9] がそれぞれ存在する．そのため，これらを非線形半正定値計画問題へと拡張する事は自然である．

しかしながら，両アルゴリズムの性能差という意味においては，非線形半正定値計画問題のそれは，非線形計画問題の場合とイコールではない．非線形計画問題においては，ほぼ全ての場合において信頼領域法を利用する方法の方が良いパフォーマンスを発揮するが，非線形半正定値計画問題の場合，直線探索法の方が有利なケースが存在する．本章では，この根拠を詳しく説明する．

直線探索を利用する方法では，探索方向がメリット関数を降下させる事を担保するため，ラグランジュ関数のヘッセ行列を正定値化する事が要求される．これを保証するための方法として，準ニュートン法 (Quasi-Newton method)，レベンバーグ・マーカート法 (Levenberg-Marquardt method) 等が存在する．しかし，いずれの方法も大規模問題に向いているとは言い難い．何故なら，準ニュートン法は次数 n の密行列を取り扱う必要があり，レベンバーグ・マーカート法では，次数 n の行列の固有値問題に準じる処理が必要となるからである．

実際，非線形計画問題に対する直線探索法 [9] では，前者の準ニュートン法を用いているが，そのパフォーマンスは [13] に大きく劣る．

ところが，非線形半正定値計画問題では，探索方向に対するステップ幅の上限を求める操作に際して，別途次数 p の固有値問題を解く必要がある．このため，非線形計画問題においては，計算コストの面から不利であった，レベンバーグ・マーカート法の欠点が，相対的に緩和される．特に $n \le p$ の場合，この傾向は如実に表れる．

さらに，上述の探索方向に対するステップ幅の上限を求める操作は，直線探索を利用する方法の場合は一度で済むが，信頼領域を利用する方法の場合，複数回必要な場合がある [*2]．非線形計画問題の場合，この処理に要する時間は問題とならないが，非線形半正定値計画問題の場合，この差は無視できない．

以上を纏めると，行列のサイズ p が大きく変数の数 n が小さい場合，信頼領域法よりも，レベンバーグ・マーカート法を利用した直線探索法の方が有利であることが示唆される．以下の実験結果は，その裏付けとなるものである．

ここで紹介する人為的な問題は，K 個の行列 $A^1, \cdots, A^K$ に対して，フロベニウスノルムの最小化を同時に実施するものである．但し，各行列 A^k を近似する行列 X^k は，全て同じフロベニウスノルムを持たねばならず，またその値は小さいほど良い．加えて，近似する行列 $X^1, \cdots, X^K$ には全ての自由度がある訳では無く，動かす事が可能な範囲 S は限られる．さらに，近似行列内における対角成分の値は全て等しいとする [*3]．

$$
\begin{array}{ll}
\text{minimize} & \displaystyle\sum_{k=1}^{K} \left\| X^k - A^k \right\|_F + \sum_{k=1}^{K} \left\| X^k \right\|_F \\
\text{subject to} & X^k \succeq O \\
& X^k_{ij} = v_k,\ i = j \\
& X^k_{ij} = 0,\ (i,j,k) \notin S,\ i \neq j \\
& \left\| X^{k1} \right\|_F = \left\| X^{k2} \right\|_F, \forall (k1, k2)
\end{array}
$$

以下に，この問題に対する実験結果を述べる．

n	p	K	itr(LM)	time(LM)	itr(trust)	time(trust)
116	1000	3	60	348.5	39	454.7
297	300	5	52	22.0	29	24.4
821	100	10	18	5.1	26	3.6
1502	50	20	19	25.5	22	1.7
1975	20	30	18	49.51	27	1.2

この表を見ると，行列次元 p が変数の数 n より大きい場合には，直線探索法の方が計算時間が少ない事が分かる．上述の例では大きな差は無いが，反復回数が信頼領域法の方がより少ない事を鑑みると，1 反復辺りに直せば直線探索法の優位性は明白である．一方で，変数の数 n が行列次元 p より大きな場合には，信頼領域法の方が計算時間は少ない．

参考文献

[1] Correa,R., Ramirez,H., A global algorithm for nonlinear semidefinite programming, *SIAM Journal on Optimization*, 15.1, 303-318, 2004.

[2] Fares,B., Noll,D., Apkarian,P., Robust control via sequential semidefinite programming. *SIAM Journal on Control and Optimization*, 40(6), 1791-1820, 2002.

[*2] 第三章内部反復 Step3 を参照

[*3] これは，変数の数 n が行列次元 p より小さくなる例を構築するためである．

[3] Gomez,W., Hector,R., A filter algorithm for nonlinear semidefinite programming, *Computational Applied Mathematics* 29.2, 297-328, 2010.

[4] Kanzow,C., Nagel,C., Kato,H., Fukushima,M., Successive linearization methods for nonlinear semidefinite programs. *Computational Optimization and Applications*, 31(3), 251-273, 2005.

[5] Kato,A., Yabe,H., Yamashita,H., An interior point method with a primal-dual quadratic barrier penalty function for nonlinear semidefintie programming, *Technical Report.*

[6] Kocvara,M., Stingl,M., PENNON: A code for convex nonlinear and semidefinite programming. *Optimization methods and software*, 18(3), 317-333, 2003.

[7] Leibfritz, F. "COMPleib: COnstrained Matrix optimization Problem library.", 2006.

[8] Yamakawa,N., Yamashita,N., A differentiable merit function for the shifted perturbed KKT conditions of the nonlinear semidefinite programming, *Technical Report.*

[9] Yamashita,H., A globally convergent primal-dual interior point method for constrained optimization, *Optimization Methods and Software*, Volume10-2, 443-469, 1998.

[10] Yamashita,H., Yabe,H., Local and superlinear convergence of a primal-dual interior point method for nonlinear semidefinite programming. *Mathematical programming*, 132(1-2), 1-30, 2012.

[11] Yamashita,H., Yabe,H., Harada,K., A primal-dual interior point trust region method for large scale nonlinear semidefintie programming, *Technical Report.*

[12] Yamashita,H., Yabe,H., Harada,K., A primal-dual interior point method for nonlinear semidefintie programming, *Mathematical Programming*, Volume135, 89-121, 2012.

[13] Yamashita,H., Yabe,H., Tanabe,T., A globally and superlinearly convergent primal-dual interior point trust region method for large scale constrained optimization, *Mathematical Programming*, Volume102, 111-151, 2005.