単体制約と L_1 正則化項をもつ凸計画問題に対する近接勾配法

門元 崇

摘要

大規模な凸計画問題は信号処理や統計など様々な分野で現れる．その中でも単体制約や L_1 正則化は応用上重要である．単体制約は $\sum_{i=1}^{n} x_i = 1, x_i \geq 0 (i = 1, \ldots, n)$ と表される制約で，確率などを表す際に用いられる．一方，L_1 正則化項を目的関数に加えると，疎な（0 となる要素を多く含むような）最適解を与えることができる．

大規模な凸計画問題に対する代表的な解法として，内点法が挙げられる．内点法は高精度な解を高速に得ることができる手法である．しかし目的関数のヘッセ行列が密な場合には計算が困難になることがある．近年，このような問題を解く手法として，勾配射影法や近接勾配法といった目的関数の勾配を用いた手法が注目を集めている．このような手法は高精度な解を求めるために多くの反復が必要となる反面，低精度な解であれば高速に求まることが知られている．したがって，勾配に基づく手法は解の精度があまり要求されない大規模な問題に対して有効である．その中でも近接勾配法は L_1 正則化項などの特殊な構造をもつ問題に対して，その構造を利用して効率良く解を求めることができる．これまでに L_1 正則化項をもつ制約なしの凸計画問題に対する近接勾配法はよく研究されてきている．しかし，制約をもつ場合への拡張は行われていない．

本報告書では L_1 正則化項と単体制約の両方をもつような問題について考える．このような問題に対して近接勾配法の効率の良い実装方法を提案する．さらに数値実験を行い，問題の規模を大きくした場合などの計算時間を計測することで提案手法の有用性を検証する．

目次

1 序論

近年，様々な分野で 1 万変数を越えるような大規模な凸計画問題を解く手法が求められている．そのような分野の例として，信号処理 [3] や機械学習 [1] などが挙げられる．大規模な凸計画問題に対する代表的な解法としては，内点法 [7] がある．内点法は目的関数のヘッセ行列が疎な場合に高精度な解を高速に求めることができる．しかし，上記のような応用例では一般にヘッセ行列は密になることが多く，内点法の適用が困難である．また，そのような問題では低精度な解でも十分役立つことが多い．

以下では，上記のような応用例 [1, 3] を数多く含む次の構造を持った凸計画問題を考える．

$$\begin{aligned} \min \quad & F(x) \equiv f(x) + g(x) \\ \text{s.t.} \quad & x \in Q \end{aligned} \tag{1}$$

ここで集合 $Q \subseteq \mathbb{R}^n$ は空でない閉凸集合とする．関数 $f : Q \to \mathbb{R}$ は微分可能な凸関数，関数 $g : Q \to \mathbb{R}$ は具体的な形がわかっている特殊な形をもつ凸関数である．例えば $g(x) = \|x\|$ や $g(x) = \sum_{i=1}^n |x_i|$ などが挙げられる．ここで，g の微分可能性は仮定していないことに注意する．このような問題に対して，劣勾配射影法 [2, 4]，近接点法 [10, 4, 9]，近接勾配法 [3, 11] などの手法が提案されている．

劣勾配射影法は目的関数の劣勾配を利用した手法である．劣勾配射影法の 1 回の反復の計算は比較的簡単で，素早く終えることができる．しかしながら，内点法などのニュートン型の手法と比べて収束が遅いことが知られている [2]．劣勾配射影法は一般に F が微分可能なときは 1 次収束するが，微分不可能なときは 1 次収束ですら保証されていない．一方で近接点法とは，問題 (1) の目的関数を正則化した部分問題を逐次的に解く手法である．ステップ幅を適切に選ぶことで超一次収束することが知られている [9]．近接点法の欠点は，部分問題を正確に解くには非常に時間がかかることがあるということである．そこで問題 (1) において g が特別な構造をもつ場合には，その構造を利用して g に対しては近接点法を，f の最小化には劣勾配射影法を用いた，近接勾配法が提案されている．近接勾配法は g が微分不可能であっても F が微分可能な場合の劣勾配射影法と同等の収束性をもつことが知られている [11]．

Beck と Teboulle [3] は $g(x) = \sum_{i=1}^n |x_i|$, $Q = \mathbb{R}^n$ の場合に対する近接勾配法とその高速化法を考え，その収束性を証明している．このときの部分問題は $O(n)$ で計算できる．しかし $Q = \mathbb{R}^n$ でないときは，この部分問題の解法は適用できない．

本報告書では特に，ポートフォリオ問題や機械学習，統計的な分野などで現れる，単体制約や L_1 正則化項を持つ問題について考える．つまり関数 g と集合 Q をそれぞれ $g(x) = \sum_{i=1}^n |x_i - c_i|$, $Q = \{x \in \mathbb{R}^n \mid \sum_{i=1}^n x_i = 1, x_i \geq 0\,(i = 1, \ldots, n)\}$ とした次の問題を考える．

$$\begin{aligned} \min \quad & f(x) + \sum_{i=1}^n |x_i - c_i| \\ \text{s.t.} \quad & \sum_{i=1}^n x_i = 1 \\ & x_i \geq 0 \quad (i = 1, \ldots, n) \end{aligned} \tag{2}$$

ここで $c \in \mathbb{R}^n$ は定数ベクトルとする．単体制約は応用上，確率分布やポートフォリオ問題における資産配分などを表す．一方，L_1 正則化項を含むことによって $x_i = c_i$ となるような i を多く含む解を得ることができる．本報告書ではこのような問題に対して近接勾配法を適用することを考え，その際に現れる部分問題を効率的に解くためのアルゴリズムを提案する．

本報告書の構成は以下の通りである．2 節では劣勾配射影法，近接点法および近接勾配法の紹介を行い，その収束性について説明する．3 節では問題 (2) に対して近接勾配法を適用した際に，部分問題を効率的に解くアルゴリズムを提案する．4 節では f が 2 次関数であるような場合に対して数値実験を行い，提案アルゴリズムの有用性を調べる．最後に 5 節で結論と今後の課題について述べる．

本報告書において log は対数を表し，便宜上 $0 \log 0 = 0$ として扱うものとする．また集合 Q に対して $\operatorname{int} Q$ は Q の内部を表すものとする．

2 凸計画問題に対する近接勾配法

この節では近接勾配法について紹介する．近接勾配法とは凸計画問題を解く方法の一種であり，劣勾配射影法と近接点法を組み合わせた手法である．以下ではまず劣勾配射影法と近接点法について紹介し，次に近接勾配法の説明を行う．

次の問題について考える．

$$\begin{aligned} \min \quad & F(x) \\ \text{s.t.} \quad & x \in Q \end{aligned} \tag{3}$$

ここで関数 $F : Q \to \mathbb{R}$ は連続な凸関数，集合 $Q \subseteq \mathbb{R}^n$ は空でない閉凸集合である．また，問題 (3) は最適解を持つものとする．

微分不可能な凸計画問題に対する有効な解法として，劣勾配射影法が挙げられる．$\partial F(x)$ を F の劣勾配 [8] とし，$F'(x^k) \in \partial F(x^k)$ とする．$F(x)$ が各点で微分可能である場合は，$\partial F(x) = \{\nabla F(x)\}$ となる．劣勾配射影法は次のように点列 $\{x^k\}$ を生成する方法である．

$$\begin{aligned} x^{k+1} &= \pi_Q(x^k - t_k F'(x^k)) \\ t_k &> 0 \end{aligned} \tag{4}$$

ここで t_k はステップ幅を表す．また $\pi_Q(\cdot)$ は集合 Q 上の射影である．π_Q としてユークリッド射影 $\pi_Q(z) = \underset{x \in Q}{\operatorname{argmin}} \{\|x - z\|\}$ を用いた場合，式 (4) は

$$x^{k+1} = \underset{x \in Q}{\operatorname{argmin}} \left\{ \langle x, F'(x^k) \rangle + \frac{1}{2t_k} \|x - x^k\|^2 \right\} \tag{5}$$

と表される．ここでさらに部分問題 (5) の目的関数第 2 項 $\frac{1}{2}\|x - x^k\|^2$ を距離のような関数 $D(x, x^k)$ で置き換えることで，以下のように一般化することができる．

$$x^{k+1} \in \underset{x \in Q}{\operatorname{argmin}} \left\{ \langle x, F'(x^k) \rangle + \frac{1}{t_k} D(x, x^k) \right\}$$

本報告書では関数 D として，次の Bregman 関数 B_ψ を用いる [5].

$$B_\psi(x, y) = \psi(x) - \psi(y) - \langle x - y, \nabla \psi(y) \rangle \tag{6}$$

ここで関数 ψ は $\operatorname{int} Q$ 上で連続的微分可能な強凸関数（強凸パラメータ $\sigma > 0$）である．特に $\psi(x) = \frac{1}{2}\|x\|^2$ のとき，Bregman 関数は $B_\psi(x, y) = \frac{1}{2}\|x - y\|^2$ となる．

劣勾配射影法において，$F(x)$ が微分可能で，かつ $\nabla F(x)$ が係数 L のリプシッツ連続であるとき，ステップ幅が $\frac{1}{t_k} > L$ を満たせば大域的収束することが知られている．そうでない場合でも

$\sum_k t_k = \infty, t_k \to 0$ を満たすように t_k を選ぶと収束することが保証されている [2]．しかしそのような場合には収束が著しく遅くなる．

凸計画問題 (3) に対するもう 1 つの解法として，近接点法が挙げられる [4, 9, 10]．近接点法とは，距離のような関数 $D(x, x^k)$ を目的関数に加えることで正則化された部分問題を逐次的に解く手法である．Bregman 関数 (6) を用いた近接点法は，次のような点列 $\{x^k\}$ を生成する．

$$x^{k+1} \in \operatorname*{argmin}_{x \in Q} \left\{ F(x) + \frac{1}{t_k} B_\psi(x, x^k) \right\} \tag{7}$$

このとき t_k が下に有界であれば点列 $\{x^k\}$ は問題 (3) の最小解に収束し，特に $t_k \to \infty$ となるように選ぶと超一次収束することが知られている [9]．しかし，この部分問題は一般に非線形の問題であるため，この問題を正確に解くには時間がかかる場合がある．

近年，勾配法と近接点法を組み合わせた近接勾配法が注目を集めている [3, 11]．以下では，問題 (1) のように $F(x) = f(x) + g(x)$ となる問題を考える．ここで関数 $f : Q \to \mathbb{R}$ は微分可能な凸関数，関数 $g : Q \to \mathbb{R}$ は特殊な構造をもつ凸関数とする．このとき，微分可能な f に対して勾配法を，微分不可能な点を持つ g に対しては近接点法を用いることで，各反復は次のように表される．

$$x^{k+1} = \operatorname*{argmin}_{x \in Q} \left\{ \langle \nabla f(x^k), x \rangle + \frac{1}{t_k} B_\psi(x, x^k) + g(x) \right\} \tag{8}$$

近接勾配法について，$\nabla f(x)$ が係数 L のリプシッツ連続であるとき，次のような定理が成り立つことが知られている [11]．

定理 1 $\{x^k\}$ を近接勾配法によって生成される点列とする．任意の $x \in Q$ と k に対して $B_\psi(x, x^k) \geq \frac{1}{2}\|x - x^k\|^2$ が成り立つとする．このとき，任意の $x \in Q$ に対して

$$F(x^k) \leq F(x) + \frac{1}{k} L B_\psi(x, x^0) \quad (k \geq 1) \tag{9}$$

が成立する．

これは F が微分可能なときの劣勾配射影法と同じ性質であり，g が微分不可能であっても成り立つ結果である．しかし部分問題 (8) は，一般には解くことが難しい．$g = \sum_{i=1}^n |x_i|$, $Q = \mathbb{R}^n$ の場合であれば，$\psi(x) = \frac{1}{2}\|x\|^2$ として，部分問題 (8) は

$$\begin{aligned} x^{k+1} &= T_{t_k}(x^k - t_k \nabla f(x^k)) \\ T_t(x)_i &= (|x_i| - t)_+ \operatorname{sgn}(x_i) \end{aligned}$$

と表されることが知られている [6]．しかしながら $Q = \mathbb{R}^n$ の場合でなければこのように表すことができない．次節では，$Q = \{x \in \mathbb{R}^n \mid \sum_{i=1}^n x_i = 1, x_i \geq 0 \, (i = 1, \ldots, n)\}$ の場合に対して部分問題 (8) の有効な解法を提案する．

3　単体制約と L_1 正則化項をもつ凸計画問題への近接勾配法の実装

3.1　単体制約と L_1 正則化項をもつ凸計画問題に対する近接勾配法

この節では次のような問題を考える．

$$\begin{aligned} \min \quad & f(x) + \sum_{i=1}^{n} |x_i - c_i| \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_i = 1 \\ & x_i \geq 0 \quad (i = 1, \ldots, n) \end{aligned} \tag{10}$$

ここで集合 $Q = \{x \in \mathbb{R}^n \mid \sum_{i=1}^{n} x_i = 1,\, x_i \geq 0\,(i = 1, \ldots, n)\}$ とする．関数 $f : Q \to \mathbb{R}$ は微分可能な凸関数，$c \in \mathbb{R}^n$ は定数ベクトルであり，関数 $g = \sum_{i=1}^{n} |x_i - c_i|$ とする．問題 (10) に対して近接勾配法 (8) を適用したとき，部分問題は

$$\begin{aligned} \min \quad & \langle \nabla f(x^k), x \rangle + \frac{1}{t_k} B_\psi(x, x^k) + \sum_{i=1}^{n} |x_i - c_i| \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_i = 1 \\ & x_i \geq 0 \quad (i = 1, \ldots, n) \end{aligned} \tag{11}$$

と書ける．Bregman 関数を定義する関数 $\psi : Q \to \mathbb{R}$ および Bregman 関数 $B_\psi : Q \times \operatorname{int} Q \to \mathbb{R}$ を

$$\psi(x) = \sum_{i=1}^{n} x_i \log x_i \tag{12}$$

$$B_\psi(x, y) = \sum_{i=1}^{n} \left\{ x_i \log \frac{x_i}{y_i} + y_i - x_i \right\} \tag{13}$$

とする．問題 (10) を解く近接勾配法は次のようになる．

Step 0：　初期点 $x^0 \in \operatorname{int} Q$，初期ステップ幅 $t_0 \in (0, \infty)$，正の定数 $\gamma < 1$ を与える．$k = 0$ とする．

Step 1：　停止条件を満たせば停止する．そうでない場合，次のような $\bar{x}$ を計算する．

$$\bar{x} = \operatorname*{argmin}_{x \in Q} \left\{ \langle \nabla f(x^k), x \rangle + \frac{1}{t_k} B_\psi(x, x^k) + \sum_{i=1}^{n} |x_i - c_i| \right\}$$

Step 2：　$f(\bar{x}) + g(\bar{x}) > f(x^k) + g(x^k)$ ならば $x^{k+1} = x^k$, $t_{k+1} = \gamma t_k$, $k = k + 1$ として Step 1 へ．そうでなければ $x^{k+1} = \bar{x}$, $t_{k+1} = t_k$, $k = k + 1$ として Step 1 へ．

このアルゴリズムに対して次の定理が成立する．

定理 2 関数 B_ψ を式 (13) によって定義したとき，近接勾配法によって生成される点列 $\{x^k\}$ は収束する．

証明 B_ψ の性質より，$x^k \in \operatorname{int} Q$ である．よって定理 1 より，任意の $x \in Q$ と $y \in \operatorname{int} Q$ に対して $B_\psi(x,y) \geq \frac{1}{2}\|x-y\|^2$ であることを示すことができれば，大域的収束が成立する．$B_\psi(x,y) = \sum_{i=1}^n \{x_i \log \frac{x_i}{y_i} + y_i - x_i\}$ と書けることより，以下では

$$x_i \log \frac{x_i}{y_i} + y_i - x_i \geq \frac{1}{2}(x_i - y_i)^2 \tag{14}$$

を示す．

$x_i = 0$ のとき：$y_i \in (0,1)$ から $y_i \geq \frac{1}{2}y_i^2$ が成り立つ．つまり式 (14) が成立する．

$x_i \neq 0$ のとき：関数 h を $h(t) = t\log t$ とする．h について $t \in (0,1]$ のとき

$$\nabla^2 h(t) = \frac{1}{t} \geq 1 \tag{15}$$

となる．よって Taylor の定理より

$$h(x_i) - h(y_i) = \nabla h(y_i)(x_i - y_i) + \frac{1}{2}\nabla^2 h(y_i + \tau(x_i - y_i))(x_i - y_i)^2 \tag{16}$$

となる $\tau \in (0,1]$ が存在する．ここで $y_i + \tau(x_i - y_i) \in (0,1]$ となることに注意すると，式 (15) より

$$h(x_i) - h(y_i) - \nabla h(y_i)(x_i - y_i) \geq \frac{1}{2}(x_i - y_i)^2 \tag{17}$$

が成り立つ．式 (17) の左辺は $x_i \log \frac{x_i}{y_i} + y_i - x_i$ であるから，式 (14) が成立する．

以上より，$B_\psi(x,y) \geq \frac{1}{2}\|x-y\|^2$ が成立し，定理 1 より数列 $\{x^k\}$ は収束する．□

3.2 近接勾配法の部分問題の解法

以下では，部分問題 (11) の効率の良い解き方を提案する．部分問題 (11) に対する KKT 条件 [8] は

$$\begin{gathered} t_k \nabla f(x^k) + \log \frac{x_i}{x_i^k} + t_k\eta_i - t_k\lambda - t_k\mu_i = 0 \quad (i = 1,\dots,n) \\ \eta \in \partial g(x) \\ \sum_{i=1}^n x_i = 1 \\ \mu_i \geq 0,\ x_i \geq 0,\ x_i\mu_i = 0 \quad (i = 1,\dots,n) \end{gathered}$$

と表せる．log の性質より $x_i > 0,\ (i = 1,\dots,n)$ となるから $\mu_i = 0,\ (i = 1,\dots,n)$ が成立する．さらに劣勾配の定義より，部分問題 (11) を解くことは

$$t_k \nabla f(x^k) + \log \frac{x_i}{x_i^k} + t_k\eta_i - t_k\lambda = 0 \quad (i = 1,\dots,n) \tag{18}$$

$$\eta_i \in \begin{cases} \{-1\} & (x_i < c_i) \\ [-1,1] & (x_i = c_i) \\ \{1\} & (x_i > c_i) \end{cases} \tag{19}$$

$$\sum_{i=1}^n x_i = 1 \tag{20}$$

を満たすような x, η, λ を求めることと等価になる．

式 (18) より $x_i\,(i=1,\ldots,n)$ は

$$x_i = x_i^k \exp(-t_k \nabla f(x^k)_i) \exp(t_k \lambda) \exp(-t_k \eta_i) \tag{21}$$

を満たす．ここで，$z(\lambda)$ を

$$z_i(\lambda) = x_i^k \exp(-t_k \nabla f(x^k)_i) \exp(t_k \lambda)$$

とすると式 (21) は次のように表される．

$$x_i = z_i(\lambda) \exp(-t_k \eta_i) \tag{22}$$

まず最初に，λ を固定したときに式 (18), (19) を満たす x, η がどのように表されるかを考える．

$z_i(\lambda)\exp(t_k) < c_i$ のとき：式 (22) と $\eta_i \in [-1,1]$ より $x_i < c_i$ が成立する．よって式 (19) より

$$\begin{aligned} \eta_i &= -1 \\ x_i &= z_i(\lambda)\exp(t_k) \end{aligned}$$

が成立する．

$z_i(\lambda)\exp(-t_k) > c_i$ のとき：同様にして $x_i > c_i$ が成立し，

$$\begin{aligned} \eta_i &= 1 \\ x_i &= z_i(\lambda)\exp(-t_k) \end{aligned}$$

が成立する．

$z_i(\lambda)\exp(t_k) \le c_i \le z_i(\lambda)\exp(t_k)$ のとき：$x_i < c_i$ を仮定する．式 (19) および式 (22) より $\eta_i = -1, x_i = z_i(\lambda)\exp(t_k)$ となる．これは $x_i \ge c_i$ となり仮定と矛盾する．$x_i > c_i$ と仮定した場合も同様に矛盾が導かれる．よって $x_i = c_i$ となり，式 (22) より

$$\eta_i = \frac{1}{t_k} \log \frac{c_i}{z_i(\lambda)}$$

が成立する．

以上より，λ を固定したときに式 (22) と式 (19) を満たす $x_i(\lambda)$ は

$$x_i(\lambda) = \begin{cases} z_i(\lambda)\exp(t_k) & (\lambda \le l_i) \\ c_i & (l_i < \lambda < u_i) \\ z_i(\lambda)\exp(-t_k) & (u_i \le \lambda) \end{cases} \tag{23}$$

で与えられる．ただし

$$l_i = \begin{cases} \frac{1}{t_k} \log \frac{c_i}{x_i^k} + \nabla f(x_k) - 1 & (c_i > 0) \\ -\infty & (c_i \le 0) \end{cases} \tag{24}$$

$$u_i = \begin{cases} \frac{1}{t_k} \log \frac{c_i}{x_i^k} + \nabla f(x_k) + 1 & (c_i > 0) \\ -\infty & (c_i \le 0) \end{cases} \tag{25}$$

である．ここで l_i, u_i は λ に対する場合分けの境界値を表している．$c_i \le 0$ のときは $x_i = z_i(\lambda)\exp(-t_k) > c_i$ となるため，常に $u_i \le \lambda$ が成立するよう $l_i = u_i = -\infty$ としている．

式 (23) を満たす $x_i(\lambda)$ は KKT 条件の式 (18) と式 (19) を満たしている．そこで式 (20)，つまり $\sum_{i=1}^n x_i(\lambda) = 1$ となるような λ を式 (23) を用いて求める．ここで $\sum_{i=1}^n x_i(\lambda)$ は λ に対して単調増加であり，$\lim_{\lambda\to-\infty}\sum_{i=1}^n x_i(\lambda) = 0$, $\lim_{\lambda\to+\infty}\sum_{i=1}^n x_i(\lambda) = +\infty$ が成り立つ．したがって λ に関して二分法などを用いることで効率よく計算することができる．

いま，添字集合 $L(\lambda), F(\lambda), U(\lambda)$ を

$$L(\lambda) = \{i|\lambda \le l_i\},\ F(\lambda) = \{i|\lambda \in (l_i, u_i)\},\ U(\lambda) = \{i|u_i \le \lambda\} \tag{26}$$

とする．関数 $\beta_i(\lambda)$ を次のように定義する．

$$\beta_i(\lambda) = \begin{cases} x_i^k \exp(-t_k \nabla f(x)_i + t_k) & i \in L(\lambda) \\ x_i^k \exp(-t_k \nabla f(x)_i - t_k) & i \in U(\lambda) \end{cases} \tag{27}$$

このとき式 (20) は，式 (23)，式 (27) より

$$\sum_{i=1}^n x_i(\lambda) = \sum_{i\in L(\lambda)\cup U(\lambda)} \beta_i(\lambda)\exp(t_k\lambda) + \sum_{i\in F(\lambda)} c_i = 1 \tag{28}$$

と表される．

ここで添字集合 $L(\lambda), F(\lambda), U(\lambda)$ を定める λ を一旦 $\hat{\lambda}$ に固定して考える．このとき式 (28) の解 λ^* は

$$\lambda^* = \frac{1}{t_k}\log\frac{1 - \sum_{i\in F(\hat{\lambda})} c_i}{\sum_{i\in L(\hat{\lambda})\cup U(\hat{\lambda})} \beta_i(\hat{\lambda})} \tag{29}$$

で与えられる．この λ^* に対して

$$L(\lambda^*) = L(\hat{\lambda}),\ F(\lambda^*) = F(\hat{\lambda}),\ U(\lambda^*) = U(\hat{\lambda}) \tag{30}$$

が成立すれば λ^* が式 (20) を満たす解となる．そこで，そのような $\hat{\lambda}$（とその結果 λ^*）を見つける方法を考える．

まず $\hat{\lambda}$ は，式 (29) において添字集合 $L(\lambda), F(\lambda), U(\lambda)$ を定めるためだけに用いられることに注意する．式 (26) より，$\hat{\lambda}_1 < \hat{\lambda}_2$ となる $\hat{\lambda}_1$ と $\hat{\lambda}_2$ に対して集合 $(L(\hat{\lambda}_1), F(\hat{\lambda}_1), U(\hat{\lambda}_1))$ と集合 $(L(\hat{\lambda}_2), F(\hat{\lambda}_2), U(\hat{\lambda}_2))$ が異なるのは，$\hat{\lambda}_1 < l_i < \hat{\lambda}_2$ または $\hat{\lambda}_1 < u_i < \hat{\lambda}_2$ となるような l_i または u_i が存在するときである．そこで $\{l_i\}, \{u_i\}$ で作られる区間に注目する．

まず，$l_1, \ldots, l_n, u_1, \ldots, u_n$ をソートした数列を $s = \{s_1, \ldots, s_{2n}\}$ とする．さらに $s_0 = -\infty, s_{2n+1} = \infty$ とする．各区間 $[s_i, s_{i+1}]\,(i = 1, \ldots, 2n-1)$ の中点を取った数列 C を次のように定義する．

$$C = \left\{s_1, \frac{s_1 + s_2}{2}, \ldots, \frac{s_{2n-1} + s_{2n}}{2}, s_{2n}\right\} \tag{31}$$

ここで $\lambda \in (s_i, s_{i+1})$ に対しては $(L(\lambda), F(\lambda), U(\lambda))$ は同じになるため，$\hat{\lambda}$ の選び方としては C の各要素 C_i について調べれば十分である．$\hat{\lambda} = C_i$ のとき，式 (29) によって得られた λ^* が $s_i \le \lambda^* \le s_{i+1},\ (i = 0, \ldots, 2n)$ を満たせば，式 (30) は成立することになる．$\sum_{i=1}^n x_i(\lambda)$ の単調増加性より，このような C に対して二分法を用いることで $O(\log(n))$ 回の反復で $\hat{\lambda}$ を求めることができる．ソートに $O(n\log n)$ かかるので，部分問題 (11) の計算量は $O(n\log n)$ となる．

まとめると，部分問題 (11) に対する提案手法は以下のようになる．

Input： 点 $x^k \in Q$，ステップ幅 $t_k \in (0,\infty)$，定数ベクトル c を入力として与える．
Output： $\bar{x}$ を出力する．

Step 1： 式 (24) および式 (25) を用いて l_i, u_i を計算する．
Step 2： $\{l_1,\ldots,l_n,u_1,\ldots,u_n\}$ をソートした数列 s を求める．$s_0=-\infty, s_{2n+1}=\infty$ とする．
さらに式 (31) によって数列 C を定める．
Step 3： $\alpha_1=1, \alpha_2=2n+1, j=n$ とする．
Step 4： $\hat{\lambda}=C_j$ とし，式 (29) により λ^* を計算する．$s_{j-1}\le\lambda^*\le s_j$ を満たせば Step 6 へ．
Step 5： $\sum_{i=1}^n x(\hat{\lambda})<1$ ならば，$\alpha_1=j$, $j=\lceil\frac{j+\alpha_2}{2}\rceil$ として Step 4 へ．$\sum_{i=1}^n x(\hat{\lambda})>1$ ならば，
$\alpha_2=j$, $j=\lfloor\frac{\alpha_1+j}{2}\rfloor$ として Step 4 へ．
Step6 $\bar{x}=x(\lambda^*)$ を計算する．

ここで α_1 は $\hat{\lambda}$ が存在し得る C の要素の下限，α_2 は $\hat{\lambda}$ が存在し得る C の要素の上限，j は二分法において現在探索している C の要素を表している．

4 数値実験

この節では，前節で提案したアルゴリズムの有用性を示すため，f が 2 次関数であるような問題 (10) に対してアルゴリズムを適用した数値実験結果を示す．なお実験は CPU が Intel(R)Core(TM)2Quad 2.83GHz, メモリが 4GB の計算機上で MATLAB7.6.0(R2008a) を用いて行った．

数値実験では以下の問題を解いた．

$$\begin{aligned}\min\quad & F(x)\equiv\alpha(\frac{1}{2}x^{\mathrm{T}}Vx-\mu^{\mathrm{T}}x)+\sum_{i=1}^n|x_i-c_i|\\ \text{s.t.}\quad & \sum_{i=1}^n x_i=1\\ & x_i\ge 0\,(i=1,\ldots,n)\end{aligned}\tag{32}$$

ここで行列 $V\in\mathbb{R}^{n\times n}$ は，各成分が $(-1,1)$ 上の一様乱数によって生成された行列 $M\in\mathbb{R}^{n\times n}$ を用いて $V=M^{\mathrm{T}}M$ とし，$\mu\in\mathbb{R}^n$ は各成分の和が 1 となるようにランダムに生成された $\hat{x}\in\mathbb{R}^n$ を用いて $\mu=V\hat{x}$ とする．また f と g の比率を調整する重み係数 α と定数ベクトル $c\in\mathbb{R}^n$ は，実験の目的ごとに定義する．

提案手法の有効性を調べるため，MATLAB の 2 次計画問題用ソルバー quadprog の結果を参考として用いる．ただし quadprog は通常の 2 次計画問題用ソルバーなので，$\sum_{i=1}^n|x_i-c_i|$ という項をそのまま取り扱うことはできない．そこでスラック変数 $y\in\mathbb{R}^n$ を用いて問題 (32) を等価に変換した，次の 2 次計画問題に対して quadprog を適用する．

$$\begin{aligned}\min_{x,y}\quad & \alpha(\frac{1}{2}x^{\mathrm{T}}Vx-\mu^{\mathrm{T}}x)+\sum_{i=1}^n y_i\\ \text{s.t.}\quad & \sum_{i=1}^n x_i=1\\ & x_i\ge 0,\ y_i\ge x_i-c_i,\ y_i\ge -x_i+c_i\,(i=1,\ldots,n)\end{aligned}\tag{33}$$

提案手法の目的は，比較的精度の良い近似解を高速に見つけることである．一方，quadprog は厳密な解を与える．そこで提案手法の終了条件は quadprog の解 x^* を用いて以下のように設定した．

$$\frac{F(x^k) - F(x^*)}{|F(x^*)|} < \frac{1}{100}\epsilon \tag{34}$$

この式の左辺は最小値 $F(x^*)$ に対する，$F(x^k)$ における相対的な誤差を表している．すなわち適当な非負の数 ϵ を定めると，誤差 $\epsilon\%$ 以下で停止させるという条件となる．

実験は次の 2 種類を行った．

実験 1： 許容誤差 ϵ と問題の次元 n を変化させ，計算時間と反復回数の計測を行う．初期点 $x^0 \in \mathbb{R}^n$ の各成分を $x_i^0 = \frac{1}{n}$ とし，提案アルゴリズムの初期設定値を $t_0 = 10, \gamma = \frac{1}{2}$ とする．また定数ベクトル $c \in \mathbb{R}^n$ は各成分 c_i が $(0, 1)$ 上の一様乱数によって生成された値を $\frac{1}{n}$ 倍したものとし，重み係数を $\alpha = 2$ とする．各次元ごとに 10 回乱数を発生させて問題を生成し，誤差ごとの提案アルゴリズムの計算時間と反復回数の平均，および quadprog の計算時間と反復回数の平均を算出した．

ただし MATLAB の quadprog ではメモリの都合上，次元 n が大きくなり過ぎると問題 (33) を取り扱えなくなる．よって実験対象とする次元 n は $n = 50, 100, 200, 400$ とした．

実験 2： 問題 (32) において，重み係数 α が小さいときは関数 g の影響が強くなり，最小解 x^* が c と一致しやすくなると考えられる．x と c の一致性と計算時間の関係を調べるため，定数ベクトル $c \in \mathbb{R}^n$ の各成分 $c_i > 0$ の総和を $\sum_{i=1}^n c_i = 1$ とし，重み係数 α を $\alpha = 1, 2, 4$ と変化させた際に，解 x^* の各成分が $x_i^* = c_i\,(i = 1, \ldots, n)$ となる数と提案アルゴリズムの計算時間を計測する．初期点 $x^0 \in \mathbb{R}^n$ の各成分を $x_i^0 = \frac{1}{n}$，提案アルゴリズムの初期設定値を $t_0 = 10, \gamma = \frac{1}{2}$ とする．次元 $n = 100$ とし，各重み係数 α に対して乱数をそれぞれ 3 回ずつ発生させて問題を生成し，停止条件の許容誤差は 0.001% として実験を行った．

4.1 数値実験結果

実験 1 において，次元 $n = 50, 100, 200, 400$ の問題に対して許容誤差を $5\%, 1\%, 0.1\%, 0.01\%$ とした際の実験結果を表 1-3 にまとめる．表 1 は次元と許容誤差ごとの平均計算時間，表 2 は平均反復回数，表 3 は各次元における 1 反復にかかる平均時間を表している．図 1 は次元 n の変化に応じた平均計算時間を，図 2 は許容誤差の変化に応じた平均計算時間を表したグラフである．図 3 は $n = 400$ のある 1 つの問題を解いた際の，反復ごとの精度の変化を表したグラフである．また各次元における計算時間および反復回数の最大・最小など詳細な結果を記載した表 5-8 を巻末に付録として添付した．

実験 2 において，重み係数 $\alpha = 1, 2, 4$ とした際の実験結果を表 4 にまとめる．この表は，個々の問題に対してかかった計算時間と反復回数，および解 x^* に対して $x_i^* = c_i$ となる成分の数を表している．

表 1　次元と許容誤差ごとの平均計算時間

次元 n	quadprog (sec)	誤差 5% (sec)	誤差 1% (sec)	誤差 0.1% (sec)	誤差 0.01% (sec)
50	0.5849	0.0025	0.0036	0.0072	0.0130
100	3.9372	0.0092	0.0120	0.0196	0.0328
200	57.3707	0.0152	0.0224	0.0396	0.0681
400	1017.1	0.0248	0.0405	0.0794	0.7828

表 2　次元と許容誤差ごとの平均反復回数

次元 n	quadprog	誤差 5%	誤差 1%	誤差 0.1%	誤差 0.01%
50	213.6	9.5	17.2	43.7	88.2
100	515.5	10.0	21.7	55.4	117.8
200	1202.8	14.0	28.4	67.3	132.9
400	2902.0	9.5	20.4	54.2	823.8

表 3　各次元における 1 反復ごとの計算時間

次元 n	提案手法 (sec)	quadprog (sec)
50	0.00015	0.00274
100	0.00030	0.00764
200	0.00053	0.04770
400	0.00119	0.35048

表 4　重み係数 α ごとの計算時間と反復回数

重み係数	問題番号	計算時間 (sec)	反復回数	$x_i^* = c_i$ の数
$\alpha = 1$	問題 1	0.0009	3	98
	問題 2	0.0007	2	100
	問題 3	0.0050	11	95
$\alpha = 2$	問題 4	0.0163	41	88
	問題 5	0.0240	60	83
	問題 6	0.0106	29	89
$\alpha = 4$	問題 7	0.0092	29	72
	問題 8	0.0204	57	74
	問題 9	0.0120	36	65

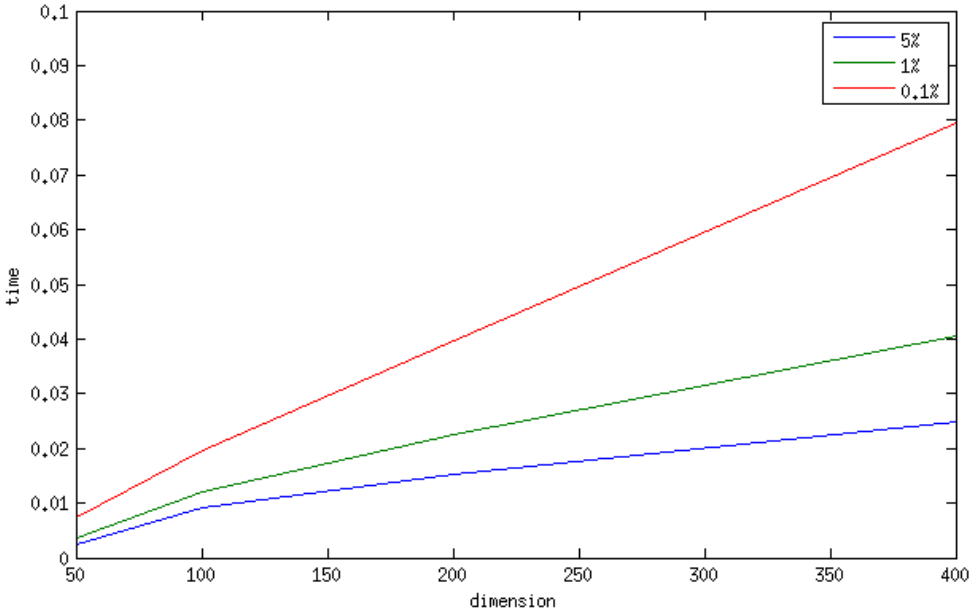

図１　次元と平均計算時間の関係

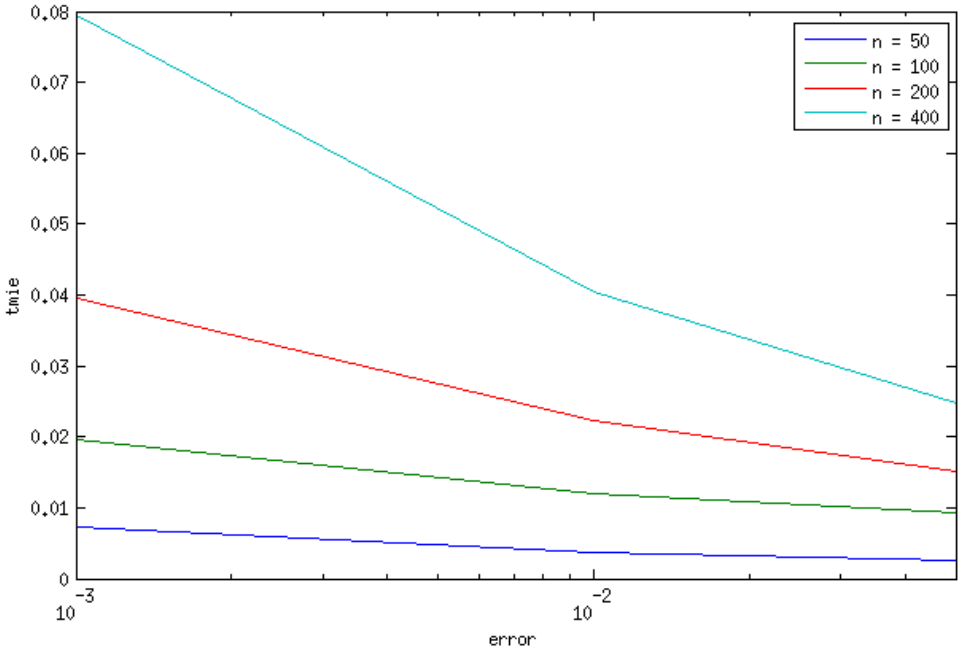

図２　許容誤差と平均計算時間の関係

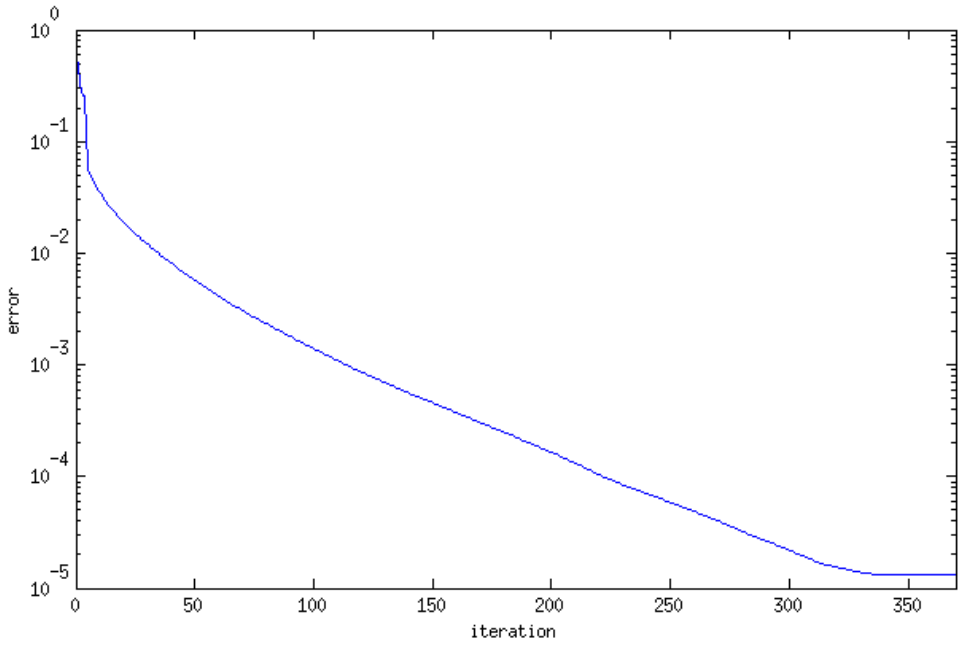

図 3　$n = 400$ における反復回数と精度の関係の例

4.2　考察

実験 1 の結果（表 1-2）から，提案アルゴリズムの反復回数および計算時間は，許容誤差を小さくするにつれて増加していることがわかる．しかしながら，0.01% 程度の誤差であれば，多くの場合 quadprog の計算時間よりも速くなる．また，提案アルゴリズムと quadprog の計算時間の差は，次元が増加するにつれて顕著になる．これは表 3 から読み取れるように，次元に対する 1 反復における計算時間の増加が，提案アルゴリズムは少ないということによる．また図 1-2 から，許容誤差を大きく取った場合には計算時間が次元の増加の影響を受け難いことがわかる．特に表 1 より，$n = 400$ の場合に 0.01% の誤差に対しても quadprog と比較して計算時間が $\frac{1}{1000}$ 程度となる場合もあることがわかる．したがって大規模な問題においてある程度の精度の近似解を素早く求めたい場合，提案アルゴリズムが効果的であると言える．

また次元の大きな問題に対して高い精度を要求して提案アルゴリズムを適用した際に，膨大な量の反復回数が必要となる場合があることが，巻末に掲載した表 8 の誤差 0.01% における最大計算時間や最大反復回数からわかる．これは図 3 のように，提案アルゴリズムの解が一定の精度に達するとそれ以降は解の精度の改善が著しく遅くなる場合があるということによる．

実験 2 における結果の表 4 から，提案アルゴリズムの計算時間は x_i^* と c_i の一致性に大きく影響を受けており，とりわけ $x^* = c$ となる場合には，反復回数と計算時間が大幅に小さくなることがわかる．

5 結論と今後の課題

本報告書では，L_1 正則化項と単体制約を持つような大規模凸計画問題に対して近接勾配法を適用し，その部分問題に対して効率的な解法を提案した．また数値実験によって，提案手法がどのような問題に対して有効なのかを検証した．

今後の課題としては，L_1 正則化項や単体制約のみならず，目的関数に区分線形な関数が加わる場合や上下限制約をもつ場合など，より一般的な場合に対する手法の開発が挙げられる．また実際の応用問題に対して提案手法を適用し，その有効性を確かめることも重要である．

謝辞

本報告書の作成にあたり，細部に至るまでご指導をいただいた山下信雄准教授に深く感謝の意を表します．また，日頃よりお世話になっている福島雅夫教授，林俊介助教ならびに福島研究室の皆様に厚く御礼申し上げます．

参考文献

[1] A. Argyriou, T. Evgeniou, and M. Pontil, Convex multi-task feature learning, *Machine Learning*, Vol. 73, pp. 243–272, 2008.

[2] A. Beck and M. Teboulle, Mirror descent and nonlinear projected subgradient methods for convex optimization, *Operations Research Letters*, Vol. 1, pp. 167–175, 2002.

[3] A. Beck and M. Teboulle, A fast shrinkage-thresholding algorithm for linear inverse problems, *SIAM J. Imaging Sciences*, Vol. 2, No. 1, pp. 183 – 202, 2009.

[4] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, 2nd edition, 1999.

[5] L. M. Bregman, The relaxation method of finding the common pont convex sets and its application to the solution of problems in convex programming, *USSR Computational Mathematics and Mathematical Physics*, Vol. 7, pp. 200–217, 1967.

[6] A. Chambolle, R. A. Devore, N. Y. Lee, and B. J. Lucier, Nonlinear wavelet image processing: Variational problems, compression, and noise removal through wavelet shrinkage, *IEEE Transactions on Image Processing*, Vol. 7, pp. 319–335, 1998.

[7] 福島雅夫, 数理計画入門, 朝倉書店, 1996.

[8] 福島雅夫, 非線形最適化の基礎, 朝倉書店, 2001.

[9] 茨木俊秀, 福島雅夫, 最適化の手法, 共立出版, 1993.

[10] R. T. Rockafellar, Monotone operators and the proximal point algorithm, *SIAM Journal on Control and Optimization*, Vol. 14, pp. 877–898, 1976.

[11] P. Tseng, Approximation accuracy, gradient methods, and error bound for structured convex optimization, *Mathematical Programming*, Vol. 1, pp. 263–295, 2010.

付録 A　数値実験結果詳細

表 5　$n = 50$ のとき

	quadprog	誤差 5%	誤差 1%	誤差 0.1%	誤差 0.01%
平均計算時間 (sec)	0.5849	0.0025	0.0036	0.0072	0.0130
最大計算時間 (sec)	0.6364	0.0054	0.0071	0.0107	0.0205
最小計算時間 (sec)	0.5476	0.0014	0.0020	0.0037	0.0053
平均反復回数	213.6	9.5	17.2	43.7	88.2
最大反復回数	239	27	39	72	152
最小反復回数	195	4	8	17	30

表 6　$n = 100$ のとき

	quadprog	誤差 5%	誤差 1%	誤差 0.1%	誤差 0.01%
平均計算時間 (sec)	3.9372	0.0092	0.0120	0.0196	0.0328
最大計算時間 (sec)	4.1057	0.0232	0.0263	0.0302	0.0548
最小計算時間 (sec)	3.8048	0.0028	0.0046	0.0095	0.0171
平均反復回数	515.5	10.0	21.7	55.4	117.8
最大反復回数	556	28	52	86	215
最小反復回数	478	4	11	33	54

表 7　$n = 200$ のとき

	quadprog	誤差 5%	誤差 1%	誤差 0.1%	誤差 0.01%
平均計算時間 (sec)	57.3707	0.0152	0.0224	0.0396	0.0681
最大計算時間 (sec)	59.4375	0.0388	0.0557	0.0876	0.1425
最小計算時間 (sec)	56.5930	0.0068	0.0103	0.0246	0.0421
平均反復回数	1202.8	14.0	28.4	67.3	132.9
最大反復回数	1289	39	73	151	277
最小反復回数	1161	5	11	31	77

表 8　$n = 400$ のとき

	quadprog	誤差 5%	誤差 1%	誤差 0.1%	誤差 0.01%
平均計算時間 (sec)	1017.1	0.0248	0.0405	0.0794	0.7828
最大計算時間 (sec)	1038.8	0.0606	0.0910	0.1414	5.3903
最小計算時間 (sec)	987.7	0.0145	0.0258	0.0529	0.1069
平均反復回数	2902.0	9.5	20.4	54.2	823.8
最大反復回数	3069	25	45	92	5898
最小反復回数	2657	5	13	39	82