数理解析研究所講究録
第 1676 巻 2010 年 195-208

絶対値計画問題に対する主双対法と逐次線形化アルゴリズム

京都大学・情報学研究科　山中翔太
京都大学・情報学研究科　福島雅夫

概要

目的関数や制約条件に変数の絶対値を含む問題を，絶対値計画問題という．また，変数の絶対値を含む方程式を絶対値方程式という．絶対値方程式は線形相補性問題と等価であることが知られている．この事実より，絶対値計画問題は均衡制約をもつ数理計画問題の特殊な場合を含んでいる．絶対値方程式の解法として，それを解くことで絶対値方程式の解が得られるような最適化問題を構成し，逐次線形化アルゴリズムを適用する方法が提案されている．

本稿では，変数の絶対値を含む不等式が絶対値方程式に変換できることを用いて，絶対値計画問題の主問題と双対問題に関する双対ギャップが 0 となるときには，絶対値計画問題を絶対値方程式に等価に変換できることを示す．また，双対ギャップが 0 であるようなテスト問題に対して，逐次線形化アルゴリズムと平滑化手法の二つの手法を適用し，数値実験により両者を比較する．その結果，逐次線形化アルゴリズムは多くの問題において，広い範囲から選んだ初期点に対して最適解に収束することが確認された．

1　序論

絶対値計画問題 (Absolute Value Programming, 以下 AVP) とは，以下のような問題である．

$$
\begin{array}{lll}
\text{(P)} & \min & c'x + d'|x| \\
 & \text{s.t.} & Ax + B|x| = b \\
 & & Hx + K|x| \geq p
\end{array}
$$

ここで，$c, d \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $p \in \mathbb{R}^k$, $A, B \in \mathbb{R}^{m\times n}$, $H, K \in \mathbb{R}^{k\times n}$ である．また，$'$ は転置を表し，$|x|$ は x の各要素の絶対値を要素とするベクトルである．この問題 (P) に対して双対問題 (D) は以下のように定義される．

$$
\begin{array}{lll}
\text{(D)} & \max & b'u + p'v \\
 & \text{s.t.} & |A'u + H'v - c| + B'u + K'v \leq d \\
 & & v \geq 0
\end{array}
$$

ここで，問題 (D) は凸計画問題であり，問題 (P) は一般に凸計画問題ではないことに注意する．

AVP の主問題 (P) と双対問題 (D) に関して，Mangasarian[5] は二つの定理を示した．一つは弱双対定理であり，(P) と (D) の任意の実行可能解における目的関数値の大小関係を与えるものである．もう一つは最適性の十分条件であり，(P) と (D) の各々の実行可能解における目的関数値が一致すれば，それらの実行可能解は (P) と (D) の最適解である，というものである．

また，AVP と密接に関係している問題に，次式のように表される絶対値方程式 (Absolute Value Equation, 以下 AVE) がある．

$$\text{(AVE)} \quad Ax + B|x| = b \tag{1}$$

ここで，$A, B \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m$ である．Mangasarian[5] と Prokopyev[9] は，この AVE が線形相補性問題 (Linear Complementarity Problem, 以下 LCP) と等価であることを示した．Mangasarian と Meyer[8] は，特に $B = -I$ の場合にも LCP と等価であることを示した．LCP とは，

$$(\mathrm{LCP}(M, q)) \quad 0 \leq z \perp Mz + q \geq 0$$

をみたすベクトル $z \in \mathbb{R}^n$ を求める問題であり，二次計画問題や双行列ゲームをはじめ多くの重要な数理計画問題を含んでいる [2]．ここで $M \in \mathbb{R}^{n\times n}$，$q \in \mathbb{R}^n$ であり，$x \perp y$ はベクトル x と y が直交することを表す．特に，$(I - M)^{-1}$ が存在するとき，LCP は以下のような形の AVE に変換できる．

$$\begin{aligned}(I + M)(I - M)^{-1}x - |x| &= -((I + M)(I - M)^{-1} + I)q \\ z &= (I - M)^{-1}(x + q)\end{aligned}$$

AVE の解法の一つとして，それを解くことで絶対値方程式の解が得られるような最適化問題を考え，逐次線形化アルゴリズム (Successive Linearization Algorithm, 以下 SLA) を用いて解く方法が Mangasarian[5, 6] によって提案されている．また，AVE(1) において $B = -I$ であるとき，ある仮定の下で適用できるアルゴリズムとして，Mangasarian[7] によって一般化ニュートン法 (Generalized Newton Method, 以下 GNM) が提案され，LCP を変換して得られた AVE に GNM を適用した数値実験結果が報告されている．

また，AVE が LCP と等価であることから，(P) は均衡制約をもつ数理計画問題 (Mathematical Program with Equilibrium Constraints, 以下 MPEC) の特殊な場合を含んでいる．

以上に述べたことを含め，AVP および AVE に関する研究は非常に少ないのが現状である．

本稿では，AVP の主問題 (P) と双対問題 (D) の制約条件に含まれる不等式を，スラック変数を導入することで AVE に等価に変換できることを示す．その結果，(P) と (D) の制約条件は全て AVE となることを用いて，双対ギャップが 0 のときに AVP が AVE に等価に変換できることを示す．次に，数値実験に必要となる双対ギャップが 0 となるようなテスト問題の作成方法を紹介し，それらの問題に対して次の二つの手法を適用した数値実験を行う．一つは (P) と (D) の両方の問題を必要とする SLA を用いる手法であり，もう一つは直接 (P) に対して適用する平滑化手法である．

本稿の構成は以下の通りである．2節では，AVPの主問題(P)と双対問題(D)に対する定理を示す．3節では，AVEとLCPの等価性とNP困難性を示し、AVPの主問題(P)の特別な場合であるMPECの例を紹介する．4節では，AVEの解を得るために解く最適化問題と，その問題を解くために提案されているアルゴリズムであるSLAを示す．5節では，双対ギャップが0となるようなテスト問題の作成方法を与え，6節ではその方法を用いて作成したAVPのテスト問題に対して，SLAと平滑化手法を比較した数値実験の結果を報告する．最後に7節において，まとめと考察を行う．

2　絶対値計画問題の諸性質

この節では，AVPに関して既存の研究[5]で知られている結果を示す．AVPの双対問題(D)は凸計画問題であるが，主問題(P)は凸計画問題ではなく，一般に双対ギャップが存在する．それらの問題に対して以下の二つの定理が示されている．

最初の定理は弱双対定理であり，最小化問題である(P)と最大化問題である(D)のそれぞれの目的関数値の関係を示している．特に，双対問題の実行可能解が得られると，その点における目的関数値は主問題の最小値の一つの下界値を与える．

定理 1. (P)と(D)の任意の実行可能解xと(u, v)に対して，

$$c'x + d'|x| \geq b'u + p'v$$

が成り立つ．

次の定理は最適性の十分条件を与える．

定理 2. $\bar{x}$と$(\bar{u}, \bar{v})$をそれぞれ(P)と(D)の実行可能解とする．このとき$\bar{x}$と$(\bar{u}, \bar{v})$において(P)と(D)の目的関数値が一致するならば，すなわち

$$c'\bar{x} + d'|\bar{x}| = b'\bar{u} + p'\bar{v}$$

が成立すれば，$\bar{x}$と$(\bar{u}, \bar{v})$はそれぞれ(P)と(D)の最適解である．

以降のAVPに関する議論は，以上の二つの定理に基づいている．

3　絶対値方程式の諸性質

この節ではAVEとその諸性質を示す．次の定理は，AVEとLCPが等価であることを示すものである[5, 9]．

定理 3. LCP(M,q) の解 z は次の AVE を解くことで得られる.

$$\begin{aligned}(I+M)(I-M)^{-1}x-|x| &= -((I+M)(I-M)^{-1}+I)q\\ z &= (I-M)^{-1}(x+q)\end{aligned}$$

また，AVE(1) の解 x は，次のように定めた $\tilde{M}$ と $\tilde{q}$ から定義される LCP$(\tilde{M},\tilde{q})$ の解 z を $z=(p,y,\theta)\in\mathbb{R}^l$, $p\in\mathbb{R}^n$, $y\in\mathbb{R}^n$, $\theta\in\mathbb{R}^{l-2n}$ と置いたときに，$x=p-y$ で与えられる[1].

$$\tilde{M}=\begin{pmatrix}-I & 2I & 0\\ A & B-A & 0\\ -A & A-B & 0\\ 0 & 0 & 0\end{pmatrix},\quad \tilde{q}=\begin{pmatrix}0\\ -b\\ b\\ 0\end{pmatrix}$$

ここで $\tilde{M}$, $\tilde{q}$ のゼロの成分は $\tilde{M}\in\mathbb{R}^{l\times l}$, $\tilde{q}\in\mathbb{R}^l$, $l=\max\{n+2m,2n\}$ となるように定められる.

定理3より AVE(1) が LCP(M,q) と等価であることから，問題 (P) は，相補性条件として LCP(M,q) を制約条件にもつ，次のような問題を含む.

$$\begin{aligned}\min\quad & c'x\\ \text{s.t.}\quad & Ax=b,\ Hx\le p\\ & x\ge 0,\ Mx+q\ge 0\\ & x'(Mx+q)=0\end{aligned}$$

このように，変分不等式や相補性条件を制約条件に含むような数理計画問題は均衡制約をもつ数理計画問題 (MPEC) と呼ばれ，工学設計や経済における市場均衡などの研究に用いられている. MPEC は，実行可能領域は凸ではなく，連結でさえない場合があるため，非常に取り扱いにくい問題である.

また，定理3と，NP 困難であるナップサック実行可能性問題 (Knapsack Feasibility Problem, 以下 KFP) [2]が

$$\hat{M}=\begin{pmatrix}-I & 0 & 0\\ e' & -n & 0\\ -e' & 0 & -n\end{pmatrix},\quad \hat{q}=\begin{pmatrix}a\\ -b\\ b\end{pmatrix}$$

によって定義される LCP$(\hat{M},\hat{q})$ と等価である [5] という事実を用いて AVE(1) の NP 困難性が示されている [6]. ここで $\hat{M}\in\mathbb{R}^{(n+2)\times(n+2)}, \hat{q}\in\mathbb{R}^{n+2}, a=(a_1,\dots,a_n)'$ であり，e はすべての要素が 1 のベクトルである. このことより大規模な AVE(1) の解を得るのは非常に困難であると考えられる.

---

[1]以降，表記の煩雑さを避けるため $(p',y',\theta')'$ を (p,y,θ) と記す.

[2]KFP とは $n+1$ 個の正の整数 $a_1,\dots,a_n,b$ が与えられたときに，$a_1x_1+\cdots a_nx_n=b$ をみたす $x\in\{0,1\}^n$ が存在するかどうかを判定する問題である [1].

4 アルゴリズム

AVE(1) の解法として，AVE(1) から導かれた凹最小化問題 (Concave Minimization Problem, 以下 CMP) [3] に対して逐次線形化アルゴリズム (SLA) を適用する方法が提案されている [5]．この節では AVE(1) から導かれる CMP と，その CMP に対するアルゴリズム SLA を示す．

4.1 絶対値方程式と最適化問題 CMP

まず，以下の線形計画問題 (Linear Program, 以下 LP) を定義する．

$$\min_{z\in Z} d'z, \quad Z=\{z \mid Hz\le h\}\neq\emptyset \tag{2}$$

ここで $d\in\mathbb{R}^l, H\in\mathbb{R}^{k\times l}, h\in\mathbb{R}^k$ である．

次に，LP(6) とそれに関連して定義される CMP の関係を示す定理を紹介する [5]．

定理 4. LP(6) に対して，次の CMP を考える．

$$\min_{z\in Z} d'z+\epsilon f(z), \quad Z=\{z \mid Hz\le h\}\neq\emptyset \tag{3}$$

ここで，$f(z)$ は $\mathbb{R}^l$ 上の凹関数とする．また，ある $\tilde{\epsilon}>0$ が存在し，任意の $\epsilon\in(0,\tilde{\epsilon}]$ に対して CMP(7) の目的関数は Z 上で有界であるとする．さらに，Z は直線を含まないものとする．このとき，LP(6) の空でない解集合 $\bar{Z}$ 上で $f(z)$ を最小化する実行可能領域の頂点 $\bar{z}$ が，任意の $\epsilon\in(0,\bar{\epsilon}]$ に対して (7) の解となるような $\bar{\epsilon}\le\tilde{\epsilon}$ が存在する．

定理 4 より，次のような CMP を解くことで，AVE(1) の解が得られることがいえる [5].

$$\begin{array}{ll} \displaystyle\min_{(x,t,s)\in\mathbb{R}^{n+n+m}} & \epsilon(-e'|x|+e't)+e's \\ \text{s.t.} & -s\le Ax+Bt-b\le s \\ & -t\le x\le t \end{array} \tag{4}$$

定理 5. もし AVE(1) が解をもつならば，CMP(8) を解いて得られた解 $\bar{x}$ は AVE(1) の解である．

証明．$z=(x,t,s)$ とし，CMP(8) の実行可能領域を Z と置いて定理 4 を適用する．すなわち，LP $\min_{z\in Z} e's$ の解集合上で $-e'|x|+e't$ を最小化する点 $\bar{z}=(\bar{x},\bar{t},\bar{s})$ に対して，$s\ge 0, t\ge |x|$ を考慮すると $\bar{s}=0, \bar{t}=|\bar{x}|$ が成り立つ．このとき $A\bar{x}+B\bar{t}-b=0$ が成立しているので，$\bar{x}$ は AVE(1) の解である． □

[3] CMP とは凸多面体上で凹関数を最小化する問題である．

4.2 逐次線形化アルゴリズム

CMP(8) を解くアルゴリズム SLA を述べる.

SLA

$z=(x,t,s)$ と置き，CMP(8) の実行可能領域を Z，目的関数を $f(z)$ と書く．適当な $z^0 \in \mathbb{R}^{n+n+m}$ を初期点とし，z^i において定義される次の最適化問題を解いて z^{i+1} を定める．ただし，ξ は $f(z)$ の z^i における劣勾配を表す.

$$z^{i+1} \in arg\ vertex\ \min_{z\in Z} \xi'(z-z^i)$$

$\xi'(z^{i+1}-z^i)=0$ となれば終了し，そうでなければ $i \leftarrow i+1$ として次の反復に進む.

ここで $arg\ vertex\ \min_{z\in Z} \xi'(z-z^i)$ とは，LP $\min_{z\in Z} \xi'(z-z^i)$ の実行可能領域の頂点であるような最適解の集合を表す.

また，アルゴリズムを適用する際に，凹関数 $-e'|x|$ の劣勾配の第 i 成分を以下のように定めて用いる.

$$-\mathrm{sign}(x_i) = \left\{ \begin{array}{ll} 1 & (x_i<0) \\ 0 & (x_i=0) \\ -1 & (x_i>0) \end{array} \right\},\quad i=1,\cdots,n.$$

CMP の性質として，最適解は少なくとも頂点に一つ存在することが知られている．この性質を利用して，SLA は現在の反復点において目的関数を線形近似し，実行可能領域の頂点を次々に求めることにより，CMP の最適解に到達しようとするアルゴリズムである．SLA は有限回で CMP の局所的最適解に到達して終了することが示されている [4, 5].

5 絶対値計画問題の絶対値方程式への変換

この章では，(P) と (D) の双対ギャップが 0 となるときには，AVP を AVE へ等価に変換することにより，SLA が適用できることを示す.

次の定理は，ある仮定の下で AVP が AVE に等価に変換できることを示している.

定理 6. AVP の主問題 (P) と双対問題 (D) が共に実行可能かつ双対ギャップが 0 なら，AVP は AVE に等価に変換される.

証明．(P) と (D) が実行可能かつ双対ギャップが 0 となる点を求めるには以下の連立方程式・不等式を解けばよい.

$$\begin{cases} Ax+B|x|=b \\ Hx+K|x|\geq p \\ |A'u+H'v-c|+B'u+K'v\leq d \\ v\geq 0 \\ c'x+d'|x|-b'u-p'v=0 \end{cases} \tag{5}$$

ここで不等式を方程式に変換するために，各々の不等式に対してスラック変数 $y \in \mathbb{R}^k$, $w \in \mathbb{R}^k$, $s \in \mathbb{R}^n$, $t \in \mathbb{R}^n$ を導入し，次のように等価な方程式に変換する．

$$\begin{aligned} Hx + K|x| \geq p &\iff Hx + K|x| - |y| = p \\ v \geq 0 &\iff v - |w| = 0 \end{aligned}$$

$$\begin{aligned}
& |A'u + H'v - c| + B'u + K'v \leq d \\
\iff & |A'u + H'v - c| \leq d - B'u - K'v \\
\iff & -(d - B'u - K'v) \leq A'u + H'v - c \leq d - B'u - K'v \\
\iff & \begin{cases} -d + B'u + K'v \leq A'u + H'v - c \\ A'u + H'v - c \leq d - B'u - K'v \end{cases} \\
\iff & \begin{cases} (B-A)'u + (K-H)'v \leq d - c \\ (A+B)'u + (H+K)'v \leq d + c \end{cases} \\
\iff & \begin{cases} (B-A)'u + (K-H)'v + |s| = d - c \\ (A+B)'u + (H+K)'v + |t| = d + c \end{cases}
\end{aligned}$$

よって (9) は以下の連立方程式と等価である．

$$\begin{cases} Ax + B|x| = b \\ Hx + K|x| - |y| = p \\ (B-A)'u + (K-H)'|w| + |s| = d - c \\ (A+B)'u + (H+K)'|w| + |t| = d + c \\ c'x + d'|x| - b'u - p'|w| = 0 \end{cases}$$

この連立方程式は次の AVE として表すことができる．

$$\begin{pmatrix} A & 0 & 0 & \\ H & 0 & 0 & \\ 0 & 0 & B'-A' & \mathbf{0} \\ 0 & 0 & A'+B' & \\ c' & 0 & -b' & \end{pmatrix} \begin{pmatrix} x \\ y \\ u \\ w \\ s \\ t \end{pmatrix} + \begin{pmatrix} B & 0 & 0 & 0 & 0 & 0 \\ K & -I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & K'-H' & I & 0 \\ 0 & 0 & 0 & H'+K' & 0 & I \\ d' & 0 & 0 & -p' & 0 & 0 \end{pmatrix} \left| \begin{matrix} x \\ y \\ u \\ w \\ s \\ t \end{matrix} \right| = \begin{pmatrix} b \\ p \\ d-c \\ d+c \\ 0 \end{pmatrix}$$

□

6 数値実験

この節では，数値実験で用いる双対ギャップが 0 であるようなテスト問題の作成方法を示す．さらにそれらのテスト問題に対して SLA による手法と平滑化手法を数値実験により比較し，最適解が得られているかどうかを確認する．本実験は CPU が Intel(R) Core(TM)2

Duo 3Ghz×2 の Fedora7 の計算機で，MATLAB と CPLEX を用いて行った．数値実験を行うためには，(P) と (D) がともに実行可能であり，さらに双対ギャップが 0 という二つの仮定を満たす問題を作成する必要がある．そこで，実験結果を示す前にそれらの仮定を満たす問題の作成方法について述べる．

6.1 問題の作成方法

(P) と (D) が共に実行可能となるような係数は，まず x, u, v および A, B, H, K, c を定め，次に残りの定数 b, d, p を決めることで得られる．

次に双対ギャップが 0 となるような問題の作成方法を示す．上に述べたように $\bar{x}$ と $(\bar{u}, \bar{v})$ をそれぞれ (P) と (D) の実行可能解となるように定める．次に，双対ギャップを最小化するような以下の問題を考える．

$$\begin{array}{ll} \displaystyle\min_{b,c,d,p} & c'\bar{x} + d'|\bar{x}| - b'\bar{u} - p'\bar{v} \\ \text{s.t.} & H\bar{x} + K|\bar{x}| \geq p, \quad A\bar{x} + B|\bar{x}| = b \\ & |A'\bar{u} + H'\bar{v} - c| + B'\bar{u} + K'\bar{v} \leq d \end{array}$$

この問題は b, c, d, p を変数とする最小化問題であり，それ以外はすべて定数とみなしている．ここで，d と p の係数が非負であることを考慮すると，$d = |A'\bar{u} + H'\bar{v} - c| + B'\bar{u} + K'\bar{v}$, $p = H\bar{x} + K|\bar{x}|$ ととれば目的関数を最小化できる．このとき目的関数について，

$$\begin{aligned} c'\bar{x} + d'|\bar{x}| - b'\bar{u} - p'\bar{v} &= c'\bar{x} + |A'\bar{u} + H'\bar{v} - c|'|\bar{x}| + \bar{u}'B|\bar{x}| + \bar{v}'K|\bar{x}| \\ &\qquad - \bar{x}'A'\bar{u} - |\bar{x}|'B'\bar{u} - \bar{x}'H'\bar{v} - |\bar{x}|'K'\bar{v} \\ &= |A'\bar{u} + H'\bar{v} - c|'|\bar{x}| - (A'\bar{u} + H'\bar{v} - c)'\bar{x} \\ &\geq 0 \end{aligned}$$

が成り立つ．よって c の各要素を，$\mathrm{sign}((A'\bar{u} + H'\bar{v} - c)_i) = \mathrm{sign}(\bar{x}_i)$ もしくは $c_i = (A'\bar{u} + H'\bar{v})_i$ をみたすようにとれば，双対ギャップが 0 となる．

この方法を用いて生成した (P) と (D) において $\bar{x}$ と $(\bar{u}, \bar{v})$ はそれぞれ最適解の一つである．よって，数値実験を行う際には，最適解の一つは既知であることに注意する．

6.2 不等式制約のみの場合

6.2.1 逐次線形化アルゴリズムと平滑化手法

数値実験では，以下の AVP の主問題 (P′) と双対問題 (D′) を用いる．

$$\begin{array}{lll} (\mathrm{P}') & \min & c'x + d'|x| \\ & \text{s.t.} & Hx + K|x| \geq p \end{array}$$

$$
\begin{array}{lll}
(\mathrm{D}') & \max & p'v \\
 & \text{s.t.} & |H'v - c| + K'v \le d \\
 & & v \ge 0
\end{array}
$$

ここで，$c, d \in \mathbb{R}^n$，$p \in \mathbb{R}^k$，$H, K \in \mathbb{R}^{k\times n}$ は，前節で述べた方法を用いて双対ギャップが 0 となるように定められているとする．

定理 6 の証明に示した方法を (P′) と (D′) に適用すると，次のような AVE が得られる．

$$
\begin{pmatrix} H & \\ 0 & \\ 0 & \mathbf{0} \\ 0 & \\ c' & \end{pmatrix}
\begin{pmatrix} x \\ y \\ w \\ s \\ t \end{pmatrix}
+
\begin{pmatrix} K & -I & 0 & 0 & 0 \\ 0 & 0 & K'-H' & I & 0 \\ 0 & 0 & H'+K' & 0 & I \\ d' & 0 & -p' & 0 & 0 \end{pmatrix}
\begin{vmatrix} x \\ y \\ w \\ s \\ t \end{vmatrix}
=
\begin{pmatrix} p \\ d-c \\ d+c \\ 0 \end{pmatrix}
$$

この AVE から導かれる CMP(8) に対して SLA を適用する．ただし，CMP(8) の目的関数において $\epsilon = 10^{-3}$ とする．

平滑化手法の数値実験では以下のアルゴリズムを採用した．数値実験を行う上で，最大反復回数は $imax = 20$ と定めた．

平滑化

Step 0: $\nu_1 = 1$ とおく．

Step 1: $|x|$ を $\tilde{x} \equiv (\sqrt{x_1^2 + \nu_i}, \ldots, \sqrt{x_n^2 + \nu_i})'$ と平滑化して得られる近似問題 (P′) を解く．

Step 2: Step1 で得られた解において (P′) が実行可能かつ双対ギャップが 10^{-3} より小さい，もしくは $i = imax$ なら終了．そうでなければ Step3 へ

Step 3: $\nu_{i+1} = \dfrac{\nu_i}{10}$，$i \leftarrow i + 1$ として Step1 へ

このように平滑化することで (P) は通常の非線形最適化問題となり，MATLAB のソルバ fmincon を用いて直接解くことができる．

6.2.2　実験結果

双対ギャップが 0 となるような (P′) と (D′) を，$(k, n) = (50, 100)$ と $(k, n) = (100, 50)$ の二つの場合に対して，それぞれランダムに 10 問生成し，SLA と平滑化手法を適用した．ここで，k と n は作成した問題における行列 $K, H \in \mathbb{R}^{k\times n}$ のサイズである．各問題において，行列とベクトルの要素はすべて $(-1, 1)$ 上の一様乱数に従って生成した．初期点は各問題において次のように三つ生成した．

ケース 1：$(-0.1, 0.1)$ 上の一様乱数に従って生成した点
ケース 2：作成した問題の既知の最適解に，各要素を $(0.1, 1)$ の一様乱数にランダムに負の符号を付けて生成したベクトルを加えた点

ケース3：作成した問題の既知の最適解に，各要素を (1, 10) の一様乱数にランダムに負の符号を付けて生成したベクトルを加えた点

ケース1，ケース2とケース3の結果をそれぞれ表1，表2と表3に示した．

SLA について，time は計算時間の一問あたりの平均で単位は秒，iter は反復回数の一問あたりの平均，dif は作成した問題の既知の最適解と得られた解との差のベクトルの無限大ノルムが 10^{-5} よりも大きかった問題数，gap は得られた解において双対ギャップが 10^{-3} より大きかった問題数，pf と df は得られた解がそれぞれ主問題と双対問題の実行可能解でなかった問題数である．平滑化手法において，time は平滑化アルゴリズムの最終反復における計算時間の一問あたりの平均，fail は $i = imax$ となってアルゴリズムが終了した問題数である．

表1：ケース1の結果

	SLA						平滑化	
(k, n)	time	iter	dif	gap	pf	df	time	fail
(50,100)	0.38	2.0	10	0	0	0	0.83	0
(100,50)	0.26	2.0	0	0	0	0	0.43	6

ケース1. $(k, n) = (50, 100)$ の場合，SLA に関して dif=10 であることから，すべての問題に対して既知の最適解とは異なった点が得られ，gap=0 であることからそれらの点はすべて最適解であり，最適解が複数存在することが分かる．平滑化手法については，すべての問題において既知の最適解と同じ解が得られた．$(k, n) = (100, 50)$ の場合，SLA に関しては，すべての問題において既知の最適解と同じ解が得られ，平滑化手法では10問中4問において既知の最適解と同じ解が得られた．

表2：ケース2の結果

	SLA						平滑化	
(k, n)	time	iter	dif	gap	pf	df	time	fail
(50,100)	0.37	2.0	10	0	0	0	4.66	0
(100,50)	0.27	2.0	0	0	0	0	7.03	10

ケース2. $(k, n) = (50, 100)$ の場合，SLA に関して dif=10 と gap=0 より，すべての問題に対して既知の最適解とは異なった最適解が得られた．平滑化手法については，すべての問題において既知の最適解と同じ解が得られた．$(k, n) = (100, 50)$ の場合，SLA に関しては，すべての問題において既知の最適解と同じ解が得られた．平滑化手法ではすべての問題に対して反復が最大反復回数 $imax = 20$ に到達し，最適解は得られなかった．

表3：ケース3の結果

	SLA						平滑化	
(k, n)	time	iter	dif	gap	pf	df	time	fail
(50,100)	0.37	2.0	10	0	0	0	8.34	1
(100,50)	0.27	2.0	0	0	0	0	7.68	10

ケース 3. $(k,n)=(50,100)$ の場合，SLA に関して dif=10 と gap=0 より，すべての問題に対して既知の最適解とは異なった最適解が得られた．平滑化手法については，すべての問題において既知の最適解と同じ解が得られた．$(k,n)=(100,50)$ の場合，SLA に関しては，すべての問題において既知の最適解と同じ解が得られた．平滑化手法ではすべての問題に対して反復が最大反復回数 $imax=20$ に到達し，最適解は得られなかった．

表 1，表 2 と表 3 より，k と n の大小関係にかかわらず SLA の方が平滑化手法より短い計算時間で解を得ていることが分かる．また，平滑化手法では初期点を解の近くにとるほど解を得るまでの計算時間が短くなり，解ける問題数が増えている．一方，SLA の計算時間や反復回数は，初期点の取り方による差があまりない．

6.3 等式制約を含む場合

次に，等式制約と不等式制約を含む問題 (P) と (D) に対して数値実験を行う．問題の係数 c,d,b,p,A,B,H,K は，6.1 節で述べた方法で双対ギャップが 0 となるように定める．定理 6 の証明で示した方法で AVP を AVE に変換し，その AVE から導かれる CMP(8) に SLA を適用した．平滑化手法に対する実験方法は 6.2.1 節と同様である．

6.3.1 実験結果

双対ギャップが 0 となるような (P) と (D) を $(k,m,n)=(100,100,30)$, $(100,20,50)$, $(20,100,50)$, $(30,30,100)$ のそれぞれに対してランダムに 10 問ずつ生成し，SLA と平滑化手法を適用した．ただし，k,m,n は作成した問題における行列 $A,B\in\mathbb{R}^{m\times n}$, $K,H\in\mathbb{R}^{k\times n}$ のサイズである．各問題において，行列とベクトルの要素はすべて $(-1,1)$ 上の一様乱数に従って生成した．初期点は各問題において次のように三つ生成した．

ケース 1： $(-0.1,0.1)$ 上の一様乱数に従って生成した点
ケース 2： 作成した問題の既知の最適解に，各要素を $(0.1,1)$ の一様乱数にランダムに負の符号を付けて生成したベクトルを加えた点
ケース 3： 作成した問題の既知の最適解に，各要素を $(1,10)$ の一様乱数にランダムに負の符号を付けて生成したベクトルを加えた点

ケース 1，ケース 2 とケース 3 の結果をそれぞれ表 4，表 5 と表 6 に示した．表の項目は 6.2.2 節と同様である．

表 4：ケース 1 の結果

	SLA						平滑化	
(k,m,n)	time	iter	dif	gap	pf	df	time	fail
(100,100,30)	0.41	2.0	0	0	0	0	0.015	10
(100,20,50)	0.33	2.0	0	0	0	0	0.15	2
(20,100,50)	0.39	2.0	0	0	0	0	0.019	10
(30,30,100)	0.44	2.0	10	0	0	0	0.18	0

ケース 1. SLA については，$(k, m, n) = (100, 100, 30), (100, 20, 50), (20, 100, 50)$ の場合にはすべての問題において既知の最適解と同じ解が得られ，$(k, m, n) = (30, 30, 100)$ の場合には dif=10 と gap=0 より，すべての問題に対して既知の最適解とは異なった最適解が得られた．平滑化手法では，$(k, m, n) = (30, 30, 100)$ の場合にはすべての問題で，$(k, m, n) = (100, 20, 50)$ の場合には 8 問で最適解が得られ，それ以外の場合ではすべての問題に対して反復が最大反復回数 $imax = 20$ に到達し，最適解は得られなかった．

表 5：ケース 2 の結果

	SLA						平滑化	
(k, m, n)	time	iter	dif	gap	pf	df	time	fail
(100.100.30)	0.40	2.0	0	0	0	0	0.015	10
(100,20,50)	0.35	2.0	0	0	0	0	2.73	9
(20.100.50)	0.38	2.0	0	0	0	0	0.019	10
(30.30,100)	0.51	2.3	10	0	0	0	0.66	0

ケース 2. SLA については，$(k, m, n) = (100, 100, 30), (100, 20, 50), (20, 100, 50)$ の場合にはすべての問題において既知の最適解と同じ解が得られ，$(k, m, n) = (30, 30, 100)$ の場合には dif=10 と gap=0 より，すべての問題に対して既知の最適解とは異なった最適解が得られた．平滑化手法では，$(k, m, n) = (30, 30, 100)$ の場合にはすべての問題で，$(k, m, n) = (100, 20, 50)$ の場合には 1 問で最適解が得られ，それ以外の場合ではすべての問題に対して反復が最大反復回数 $imax = 20$ に到達し，最適解は得られなかった．

表 6：ケース 3 の結果

	SLA						平滑化	
(k, m, n)	time	iter	dif	gap	pf	df	time	fail
(100,100.30)	0.40	2.0	0	0	0	0	0.015	10
(100.20.50)	0.34	2.0	0	0	0	0	0.15	10
(20,100.50)	0.37	2.0	0	0	0	0	0.019	10
(30.30.100)	0.76	3.5	10	0	0	0	0.18	1

ケース 3. SLA については，$(k, m, n) = (100, 100, 30), (100, 20, 50), (20, 100, 50)$ の場合にはすべての問題において既知の最適解と同じ解が得られ，$(k, m, n) = (30, 30, 100)$ の場合には dif=10 と gap=0 より，すべての問題に対して既知の最適解とは異なった最適解が得られた．平滑化手法では，$(k, m, n) = (30, 30, 100)$ の場合には 9 問で最適解が得られ，それ以外の場合ではすべての問題に対して反復が最大反復回数 $imax = 20$ に到達し，最適解は得られなかった．

表 4，表 5 と表 6 より，SLA では，ある程度広い範囲から選んだ初期点に対しても，$(k, m, n) = (100, 100, 30), (100, 20, 50), (20, 100, 50)$ の場合には既知の最適解と同じ解が，$(k, m, n) = (30, 30, 100)$ の場合には既知の最適解とは異なった最適解が得られた．また，

SLA の反復回数と計算時間は，$(k,m,n) = (100,100,30),(100,20,50),(20,100,50)$ の場合には初期点による差はあまり無く，$(k,m,n) = (30,30,100)$ の場合には初期点を既知の最適解から離れた点にとるほど，両者とも増加している．

平滑化手法は，$(k,m,n) = (100,20,50),(30,30,100)$ の場合に初期点が既知の最適解に近いほど，多くの問題で最適解が得られている．$(k,m,n) = (100,100,30),(20,100,50)$ の場合で最適解が得られていないことから，変数の次元 n よりも等式制約条件の数 m が多いときには，最適解が得られにくいと推測される．また，平滑化手法は初期点を既知の最適解の近くに定めた方が，計算時間が短い．

等式制約を含む場合と含まない場合の実験結果より，SLA では，変数の次元 n と制約条件の数 k,m によって既知の最適解とは別の解が得られる場合もあるが，いずれの場合に対しても最適解が得られることが確認された．平滑化手法では，変数の次元 n より制約条件の数 k,n が少ない場合で，初期点を解の近くにとると最適解が得やすいことが分かった．以上の結果より，等式制約を含む場合と含まない場合の違いは見られなかった．

7 結論

本稿では，絶対値計画問題の主問題と双対問題に関する双対ギャップが 0 となるときに，絶対値計画問題を絶対値方程式に等価に変換することができることを示した．また，双対ギャップが 0 であるようなテスト問題に対して逐次線形化アルゴリズムと平滑化の二つの手法を適用し，数値実験により比較した．その結果，逐次線形化アルゴリズムは多くの問題において，広い範囲から選んだ初期点に対して最適解に収束することが確認された．

参考文献

[1] S. J. Chung, *NP-completeness of the linear complementarity problem.* Journal of Optimization Theory and Applications, Vol. 60, pp. 393-399, 1989.

[2] R. W. Cottle, J.-S. Pang and R. E. Stone, *The Linear Complementarity Problem.* Academic Press, New York, 1992.

[3] O. L. Mangasarian, *The linear complementarity problem as a separable bilinear program.* Journal of Global Optimization, Vol. 6, pp. 153-161, 1995.

[4] O. L. Mangasarian, *Solution of general linear complementarity problems via nondifferentiable concave minimization.* Acta Mathematica Vietnamica, Vol. 22, pp. 199-205, 1997.

[5] O. L. Mangasarian, *Absolute value programming.* Computational Optimization and Applications, Vol. 36, pp. 43-53, 2007.

[6] O. L. Mangasarian, *Absolute value equation solution via concave minimization.* Optimization Letters, Vol. 1, pp. 3-8, 2007.

[7] O. L. Mangasarian, *A generalized Newton method for absolute value equations*. Optimization Letters, Vol. 3, pp. 101-108, 2009.

[8] O. L. Mangasarian and R. R. Meyer, *Absolute value equations*. Linear Algebra and its Applications, Vol. 419, pp. 359-367, 2006.

[9] O. A. Prokopyev, *On equivalent reformulations for absolute value equations*. Computational Optimization and Applications, Vol. 44, pp. 363-372, 2009.