

混合相補性条件を制約に持つ 数理計画問題に対する分枝限定法

京都大学情報学研究科 田島 潤 (TAJIMA Jun)
 京都大学情報学研究科 山下 信雄 (YAMASHITA Nobuo)
 京都大学情報学研究科 福島 雅夫 (FUKUSHIMA Masao)

1 序論

本論文では、以下の均衡制約つき数理計画問題 (MPEC)[5] を考える。

$$\begin{aligned} & \text{minimize} && f(x, y) \\ & \text{subject to} && (x, y) \in X \\ & && z = g(x, y) \\ & && x \geq 0, z \geq 0, x^T z = 0 \end{aligned} \tag{1}$$

ここで、 $x \in \mathbb{R}^m$, $y \in \mathbb{R}^n$, $z \in \mathbb{R}^m$ であり、 $f: \mathbb{R}^{m+n} \rightarrow \mathbb{R}$, $g: \mathbb{R}^{m+n} \rightarrow \mathbb{R}^m$, $X \subset \mathbb{R}^{m+n}$ である。また、制約式に含まれる条件

$$x \geq 0, z \geq 0, x^T z = 0$$

を相補性条件という。

MPEC は 2 レベル数理計画問題や種々の均衡問題などを含む広いクラスの問題である [5]。このため、工学や社会科学のさまざまな分野で扱われる問題には MPEC として定式化できるものが多い。MPEC に対してはこれまでにいくつかのアルゴリズムが提案されている [3][4][6]。しかしながらこれらの手法では、生成された点列が局所的最適解に収束するためにはある種の厳しい仮定を満たす必要があり、また仮に解が求まったとしてもそれが大域的最適解である保証はない。さらに、それらのアルゴリズムの中には実際に計算機上に実装した場合、数値的に不安定になり解が求めるのが困難な場合もある。

一方、相補性条件は各添字 $i = 1, \dots, m$ に対して、 $x_i = 0, z_i \geq 0$ となるかまたは $x_i \geq 0, z_i = 0$ となることと等価である。そのため、 $J \cup L = \{1, \dots, m\}$, $J \cap L = \emptyset$ となるような 2^m 個の J, L の組み合わせすべてに対して数理計画問題

$$\begin{aligned} & \text{minimize} && f(x, y) \\ & \text{subject to} && (x, y) \in X \\ & && z = g(x, y) \\ & && x_i = 0, z_i \geq 0 \quad (i \in J) \\ & && x_i \geq 0, z_i = 0 \quad (i \in L) \end{aligned} \tag{2}$$

を解くことにより、(1) の大域的最適解を求めることができる。とくに、 f が凸関数、 g がアフィン関数で X が凸集合である場合には、(2) は凸計画問題となり、その大域最適解を容易に求めることができる。Bard と Moores[2] は、(1) において f, g がともにアフィン関数であり、集合 X が有限の線形制約式によって表される問題、つまり以下のように定式化できる問題に対する分枝限定法を提案した。

$$\begin{aligned} & \text{minimize} && c^T x + d^T y \\ & \text{subject to} && Ax + By = p \\ & && z = Mx + Ny + q \\ & && x \geq 0, z \geq 0, x^T z = 0 \end{aligned} \tag{3}$$

ここで、 A, B, M, N はそれぞれ r 行 m 列、 r 行 n 列、 m 行 m 列、 m 行 n 列の行列、 c, q は m 次元ベクトル、 d は n 次元ベクトル、 p は r 次元ベクトルである。このとき、(3) に対する問題 (2) は

線形計画問題となる。[2]で提案された分枝限定法は、単純な列挙法に比べてはるかに計算速度が速い。しかし、その分枝限定法でも相補性条件の次元 m の増加に伴い、計算時間は次元 m に対して指数的に増加する傾向にある。そのため、問題が特殊な構造を持つ場合は、その構造を利用して組合わせの数を減らす工夫をする必要がある。

一方、相補性条件を一般化した概念として、混合相補性条件がある。ここで、ベクトル x, z が混合相補性条件を満たすとは、与えられた m 次元の定数ベクトル l, u に対して、

$$l \leq x \leq u \quad \begin{cases} x_i = l_i & \implies z_i \geq 0 \\ l_i < x_i < u_i & \implies z_i = 0 \quad (i = 1, \dots, m) \\ x_i = u_i & \implies z_i \leq 0 \end{cases} \quad (4)$$

を満たすことである。 l と u の各要素はそれぞれ $-\infty, +\infty$ をとることも許されている。特に、 $l := 0, u := +\infty$ とすると、(4) は通常の相補性条件に帰着され、 $l := -\infty, u := +\infty$ のときは、(4) は等式条件 $z = 0$ となる。

現実の数理計画問題においては変数に対して上下限制約 $l \leq x \leq u$ が課されていることが多い。混合相補性条件は、変数に上下限制約がある数理計画問題のカルーシュ・キューン・タッカー条件に現われることが知られている。例えば、変数に上下限制約がある次の非線形計画問題を考えてみよう。

$$\begin{aligned} & \text{minimize} && f(x) \\ & \text{subject to} && l \leq x \leq u \end{aligned}$$

この上下限制約 $l \leq x \leq u$ を2つの不等式制約 $l - x \leq 0, x - u \leq 0$ と考えると、この問題のカルーシュ・キューン・タッカー条件は、

$$\begin{aligned} & \nabla f(x^*) - \lambda^* + \mu^* = 0 \\ & l - x^* \leq 0, x^* - u \leq 0 \\ & \begin{cases} x_i^* = l_i & \implies \lambda_i^* \geq 0 \\ x_i^* > l_i & \implies \lambda_i^* = 0 \end{cases} \quad (i = 1, \dots, m) \\ & \begin{cases} x_i^* = u_i & \implies \mu_i^* \geq 0 \\ x_i^* < u_i & \implies \mu_i^* = 0 \end{cases} \quad (i = 1, \dots, m) \end{aligned}$$

となる。ただし、 λ^*, μ^* はラグランジュ乗数である。一方、混合相補性条件を用いてカルーシュ・キューン・タッカー条件を書き下すと、

$$\begin{aligned} & \nabla f(x^*) - \nu^* = 0 \\ & l \leq x \leq u \\ & \begin{cases} x_i^* = l_i & \implies \nu_i^* \geq 0 \\ l_i < x_i^* < u_i & \implies \nu_i^* = 0 \quad (i = 1, \dots, m) \\ x_i^* = u_i & \implies \nu_i^* \leq 0 \end{cases} \end{aligned}$$

となる。このように、混合相補性条件を用いることにより、変数の上下限制約を2つの不等式制約とみなした場合と比べてラグランジュ乗数の次元を半分にすることができる。このため、変数に対する上下限制約を下位レベルの問題としてもつMPECを考えた場合、計算効率の観点からは混合相補性条件を用いるほうが有利である。特に、組み合わせ的な解法を用いるときには、組み合わせに関わる変数の数が半分になることの意義は大きい。このようなことから、本論文では混合相補性条件を制約にもつ問題に対しては、混合相補性条件をそのまま扱うことができるアルゴリズムを構築することを考える。

本論文では、以下の混合相補性条件を制約を持つ数理計画問題 (mathematical program with

mixed complementarity constraints) を考える.

$$\begin{aligned}
 & \text{minimize} && c^T x + d^T y \\
 & \text{subject to} && z = Mx + Ny + q \\
 & && l \leq x \leq u \\
 \text{MPMCC:} & && \begin{cases} x_i = l_i & \implies z_i \geq 0 \\ l_i < x_i < u_i & \implies z_i = 0 \\ x_i = u_i & \implies z_i \leq 0 \end{cases} \quad (5)
 \end{aligned}$$

ここで, M, N はそれぞれ m 行 m 列, m 行 n 列の行列であり, c, q, l, u は m 次元ベクトル, d は n 次元ベクトルである.

混合相補性条件 (4) は

$$x_i = l_i, z_i \geq 0 \quad \text{or} \quad l_i < x_i < u_i, z_i = 0 \quad \text{or} \quad x_i = u_i, z_i \leq 0 \quad (i = 1, \dots, m) \quad (6)$$

と等価であるので, 次節で説明するように, MPMCC(5) は 3^m 個の線形計画問題を解くことで大域的最適解を理論的には求めることができる. 本論文では, [2] で提案された手法を拡張して MPMCC(5) に対する分枝限定法を構築する.

本論文は次のように構成される. 第2節では MPMCC(5) に対するアルゴリズムを提案し, 第3節では数値実験の結果を報告する. 最後に, 第4節で結論を述べる.

なお, 本論文を通して $\mathcal{M} := \{1, \dots, m\}$ とし, $I \subset \mathcal{M}$ に対してその補集合を $\bar{I} := \mathcal{M} \setminus I$ と表す.

2 アルゴリズム

2.1 分枝限定法

ここでは, MPMCC(5) に対する分枝限定法の枠組みについて説明する. 混合相補性条件の全ての組み合わせを数え上げるために, 図1で示された分枝図を考える.

この分枝図では, 最上部のノードが MPMCC(5) の制約条件から混合相補性条件を取り除いた緩和問題

$$\begin{aligned}
 & \text{minimize} && c^T x + d^T y \\
 & \text{subject to} && z = Mx + Ny + q \\
 & && l \leq x \leq u
 \end{aligned}$$

に対応し, そこから各添字に対して制約 $x_i = l_i, z_i \geq 0$ または $z_i = 0$ または $x_i = u_i, z_i \leq 0$ を付加していく様子を表わしている. また, 最下部のノードでは混合相補性条件 (6) が成り立っている状態であるので, これらの最下部のノードに対応する問題の実行可能解は MPMCC(5) の実行可能解となる.

互いに素な \mathcal{M} の部分集合 J, K, L に対して, $x_i = l_i, z_i \geq 0$ ($i \in J$), $z_i = 0$ ($i \in K$), $x_i = u_i, z_i \leq 0$ ($i \in L$) と固定されているノードを (J, K, L) で表わすことにする. ノード (J, K, L) に対応する部分問題を

$$R(J, K, L): \begin{aligned} & \text{minimize} && c^T x \\ & \text{subject to} && (x, y, z) \in S(J, K, L) \end{aligned}$$

と表わす. ただし, $R(J, K, L)$ の実行可能領域 $S(J, K, L)$ は

$$S(J, K, L) := \left\{ (x, y, z) \left| \begin{array}{ll} z = Mx + Ny + q, & l \leq x \leq u \\ x_i = l_i, z_i \geq 0 & (i \in J) \\ z_i = 0 & (i \in K) \\ x_i = u_i, z_i \leq 0 & (i \in L) \\ z_i \geq 0 & (i \in \{i | u_i = +\infty\}) \\ z_i \leq 0 & (i \in \{i | l_i = -\infty\}) \end{array} \right. \right\} \quad (7)$$

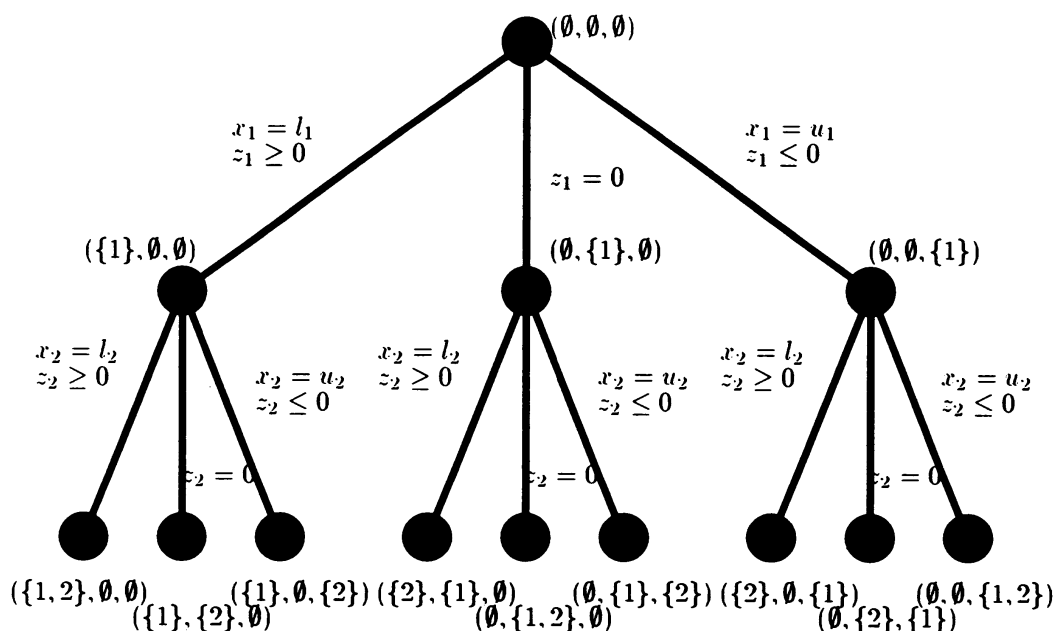


図 1: 混合相補性条件の表現 (2 変数の場合)

で定義される集合である。

このとき、MPMCC(5)の実行可能領域を T とすると、 $J \cup K \cup L = M, J \cap K = \emptyset, K \cap L = \emptyset, L \cap J = \emptyset$ であるようなすべての J, K, L に対して $S(J, K, L) \subseteq T$ であり、さらに

$$T = \bigcup_{\substack{J \cup K \cup L = M \\ J \cap K = \emptyset, K \cap L = \emptyset, L \cap J = \emptyset}} S(J, K, L)$$

が成り立つ。よって、これら 3^m 個の問題を解くことにより理論的には最適解を求めることができる。しかし、問題の次元 m が大きくなると、このようにすべての場合を数えあげていく方法は計算時間の観点から実質上不可能となる。そこで、分枝図の中で最適解が得られる可能性のない部分木をできるだけ省いて、探索する範囲を絞り込むことによって計算を高速化することが重要になる。分枝限定法では、最適解が含まれる可能性があるかどうかの判定方法として部分問題の下界値の判定を行う。いま、ノード (J, K, L) について考えているとする。 (J, K, L) から生成されるノード (分枝図ではノード (J, K, L) の子孫のノードに相当) に対応する問題の制約条件は、 $R(J, K, L)$ の制約条件にさらにいくつかの条件を付加したものになっている。そのため $R(J, K, L)$ の最適値 $\beta(J, K, L)$ と、 $R(J, K, L)$ から生成される任意の部分問題 $R(J', K', L')$ の最適値 $\beta(J', K', L')$ の間には、

$$\beta(J, K, L) \leq \beta(J', K', L')$$

という関係が成立する。つまり、 $R(J, K, L)$ の最適値はノード (J, K, L) の子孫のノードに対応する部分問題の下界値となっている。よって、MPMCC(5)のある実行可能解 $(\bar{x}, \bar{y}, \bar{z})$ が何らかの手段で求まっているとき

$$c^T \bar{x} \leq \beta(J, K, L)$$

が成立したとすると、 $R(J, K, L)$ から生成される部分問題の中に $(\bar{x}, \bar{y}, \bar{z})$ よりも良い解は存在しない。したがって、このときノード (J, K, L) の子孫ノードを探索する必要はない。

また、もし集合 $S(J, K, L)$ が空であれば、 (J, K, L) から生成されるノードに対応する問題の実行可能領域もまた空である。よって、このときもノード (J, K, L) の子孫ノードを探索する必要は

以上の議論より、次のような MPMCC(5) に対する分枝限定法の枠組みが得られる。

MPMCC(5) に対する分枝限定法

Step 0: 問題 $R(\emptyset, \emptyset, \emptyset)$ の実行可能解領域が空ならば、MPMCC(5) には実行可能解が存在しないので終了する。活性ノード集合を $\tau := \{(\emptyset, \emptyset, \emptyset)\}$ 、MPMCC(5) の暫定最適値を $\gamma := \infty$ とする。問題 $R(\emptyset, \emptyset, \emptyset)$ を解きその最適解を $(\bar{x}(\emptyset, \emptyset, \emptyset), \bar{y}(\emptyset, \emptyset, \emptyset), \bar{z}(\emptyset, \emptyset, \emptyset))$ 、最適値を $\beta(\emptyset, \emptyset, \emptyset)$ とする。もし、 $\bar{x}(\emptyset, \emptyset, \emptyset), \bar{z}(\emptyset, \emptyset, \emptyset)$ が (4) を満足していれば、MPMCC(5) の最適解が得られているので終了する。そうでなければ Step 1 へ。

Step 1: $\tau = \emptyset$ ならば、暫定最適解を MPMCC(5) の最適解として採用しアルゴリズムを終了する。そうでないときは、1つのノード $(J, K, L) \in \tau$ を選ぶ。 $\beta(J, K, L) \geq \gamma$ ならば (J, K, L) を終端し、Step 1 のはじめに戻る。そうでなければ、 $\tau := \tau \setminus \{(J, K, L)\}$ として、Step 2 へ。

Step 2: 1つの添字 $\alpha \in \bar{J} \cap \bar{K} \cap \bar{L}$ を選ぶ。ノード $(J_1, K_1, L_1) := (J \cup \{\alpha\}, K, L)$, $(J_2, K_2, L_2) := (J, K \cup \{\alpha\}, L)$, $(J_3, K_3, L_3) := (J, K, L \cup \{\alpha\})$ を生成する。各ノード (J_i, K_i, L_i) ($i = 1, 2, 3$) に対する部分問題 $R(J_i, K_i, L_i)$ の最適値を計算し、 $\beta(J_i, K_i, L_i)$ とする。(実行可能領域が空のときは $\beta(J_i, K_i, L_i) = \infty$ とおく) また、問題 $R(J_i, K_i, L_i)$ の最適値を求める過程で現在の暫定最適値 γ よりも良い MPMCC(5) の実行可能解が得られたときは、暫定最適値を更新する。Step 3 へ。

Step 3: 最適値が暫定最適値以上である部分問題 $R(J_i, K_i, L_i)$ を終端し、それ以外の部分問題を活性ノード集合に付け加える。すなわち、 $\tau := \tau \cup \{(J_i, K_i, L_i) \mid \beta(J_i, K_i, L_i) < \gamma, i = 1, 2, 3\}$ とする。Step 1 へ。

このアルゴリズムが終了したとき、MPMCC(5) の最適解が得られているか、もしくは実行可能解が存在しないか有界でないかが分かる。ここでは、MPMCC(5) に対する分枝限定法の大枠について説明したが、このアルゴリズムの計算速度は Step 1 のノードの選択法や、Step 2 の添字の選択法、下界値の計算法などに大きく依存する。次節では、下界値の計算法について説明する。

2.2 部分問題の下界値計算

本節では、ノード (J, K, L) から新たにノード $(J_1, K_1, L_1) := (J \cup \{\alpha\}, K, L)$, $(J_2, K_2, L_2) := (J, K \cup \{\alpha\}, L)$, $(J_3, K_3, L_3) := (J, K, L \cup \{\alpha\})$ を生成して、ノード (J_i, K_i, L_i) ($i = 1, 2, 3$) に対する部分問題 $R(J_i, K_i, L_i)$ の最適値を求める、すなわち、前節で提案した分枝限定法の下界値計算の詳細な手続きについて議論する。

部分問題 $R(J_i, K_i, L_i)$ は線形計画問題であるので、内点法やシンプレックス法といったアルゴリズムを用いて解くことができるが、ここではシンプレックス法を採用する。シンプレックス法を用いる理由は、問題 $R(J, K, L)$ とその子問題 $R(J_i, K_i, L_i)$ の類似性より問題 $R(J, K, L)$ の最適基底解と問題 $R(J_i, K_i, L_i)$ の最適基底解が近いと予想されるので、問題 $R(J, K, L)$ の最適解から問題 $R(J_i, K_i, L_i)$ の初期基底解を生成し、その初期基底解を用いることにより、少ないピボットにより問題 $R(J_i, K_i, L_i)$ の解を得ることが期待できるからである。

まず、 $R(J_i, K_i, L_i)$ の初期実行可能基底解を $R(J, K, L)$ の最適基底解から求める方法を説明する。いま、 $R(J, K, L)$ の最適解 $(\bar{x}, \bar{y}, \bar{z})$ が求まっているとする。そのとき、それぞれの (J_i, K_i, L_i) ($i = 1, 2, 3$) に対して、 $R(J_i, K_i, L_i)$ の初期実行可能解は次のように求められる。

部分問題 $R(J \cup \{\alpha\}, K, L)$ の初期実行可能基底解の求め方

ここでは部分問題 $R(J \cup \{\alpha\}, K, L)$ の初期実行可能基底解を求めることを考える。 $(\bar{x}, \bar{y}, \bar{z}) \in S(J, K, L)$ であるので、この制約を満たした上でさらに次の制約

$$x_\alpha = l_\alpha, z_\alpha \geq 0 \quad (8)$$

を満足する基底解が得られればよい。それには、次の問題を考える。

$$\begin{aligned} & \text{minimize} && x_\alpha + \max\{-z_\alpha, 0\} \\ & \text{subject to} && (x, y, z) \in S(J, K, L) \end{aligned} \quad (9)$$

$x_\alpha \geq l_\alpha$ であるから、この問題の最適値が l_α ならば、その最適解は (8) を満足する。これは、(9) の最適解が $R(J \cup \{\alpha\}, K, L)$ の実行可能解であることを意味している。なお、問題 (9) の最適解は新しい変数 t を導入し、等価な線形計画問題

$$\begin{aligned} & \text{minimize} && x_\alpha + t \\ & \text{subject to} && (x, y, z) \in S(J, K, L) \\ & && t \geq -z_\alpha, t \geq 0 \end{aligned}$$

を解くことによって求める。この線形計画問題にシンプレックス法を適用する際の初期実行可能基底解は、 $(\bar{x}, \bar{y}, \bar{z})$ を用いて

$$(x, y, z, t) = (\bar{x}, \bar{y}, \bar{z}, \max\{-\bar{z}_\alpha, 0\})$$

と定めることができる。さらに、シンプレックス法によって得られた問題 (9) の最適解は、 $R(J \cup \{\alpha\}, K, L)$ の実行可能基底解となる。よって、この最適解を $R(J \cup \{\alpha\}, K, L)$ の初期実行可能基底解として用いることができる。

なお、(9) の最適値が l_α より大きいときは、集合 $S(J, K, L)$ には (8) を満たす領域が存在しないことを意味しているので、 $S(J \cup \{\alpha\}, K, L)$ は空であり、 $R(J \cup \{\alpha\}, K, L)$ には実行可能解が存在しない。

部分問題 $R(J, K \cup \{\alpha\}, L)$ の初期実行可能基底解の求め方

次に、部分問題 $R(J, K \cup \{\alpha\}, L)$ の初期実行可能基底解を求めることを考える。 $(\bar{x}, \bar{y}, \bar{z}) \in S(J, K, L)$ であるので、この制約を満たした上でさらに次の制約

$$z_\alpha = 0 \quad (10)$$

を満足する基底解が得られればよい。このとき、 $z_\alpha \geq 0$ ならば次の問題

$$\begin{aligned} & \text{minimize} && z_\alpha \\ & \text{subject to} && (x, y, z) \in S(J, K, L) \\ & && \bar{z}_\alpha \geq 0 \end{aligned} \quad (11)$$

を、 $z_\alpha < 0$ ならば

$$\begin{aligned} & \text{minimize} && -z_\alpha \\ & \text{subject to} && (x, y, z) \in S(J, K, L) \\ & && \bar{z}_\alpha \leq 0 \end{aligned} \quad (12)$$

を初期実行可能解 $(\bar{x}, \bar{y}, \bar{z})$ として解く。(11) または (12) の最適値が 0 ならば、その最適解は (10) を満足する。これは、(11) または (12) の最適解が $R(J, K \cup \{\alpha\}, L)$ の実行可能基底解であることを意味している。

一方、(11) または (12) の最適値が 0 より大きいときは、集合 $S(J, K, L)$ には (10) を満たす領域が存在しないことを意味しているので、 $S(J, K \cup \{\alpha\}, L)$ は空である。

部分問題 $R(J, K, L \cup \{\alpha\})$ の初期実行可能基底解の求め方

最後に、部分問題 $R(J, K, L \cup \{\alpha\})$ の初期実行可能基底解を求めることを考える。このときは、部分問題 $R(J \cup \{\alpha\}, K, L)$ のときと同様の考え方をすればよく、次の線形計画問題

$$\begin{aligned} & \text{minimize} && -x_\alpha + t \\ & \text{subject to} && (x, y, z) \in S(J, K, L) \\ & && t \geq -z_\alpha, t \geq 0 \end{aligned}$$

に初期実行可能解を

$$(x, y, z, t) = (\bar{x}, \bar{y}, \bar{z}, \min\{-\bar{z}_\alpha, 0\})$$

としてシンプレックス法を適用すればよい。この問題の最適値が $-u_\alpha$ ならば、この問題の最適解は、 $R(J, K, L \cup \{\alpha\})$ の実行可能基底解となり、最適値が $-u_\alpha$ より大きいときには、 $R(J, K, L \cup \{\alpha\})$ には実行可能解が存在しない。

3 数値実験

ここでは、MPMCC(5) に対して本論文で提案したアルゴリズムを適用する場合と、混合相補性条件を通常の相補性条件に変換した問題に従来のアルゴリズムを適用する場合を実際に数値実験によって比較する。

数値実験では次の2レベル線形計画問題を解いた。

$$\begin{aligned} & \text{minimize} && \sum_{i=1}^k 2\sigma_i + \sum_{i=1}^k \tau_i \\ & \text{subject to} && \sum_{i=1}^k \sigma_i \leq \frac{3}{2}k \\ & && \tau_i \geq 0 \quad (i = 1, \dots, k) \\ & && \sigma_i \in \arg \min \left\{ -\lambda \left| \begin{array}{l} -2\lambda + \tau_i \leq 2 \\ \lambda + \tau_i \leq \rho_i \\ 0 \leq \lambda \leq 2 \end{array} \right. \right\} \quad (i = 1, \dots, k) \end{aligned} \quad (13)$$

ただし、 ρ_i は $2 \leq \rho_i \leq 8$ を満たすパラメータであり、 k は問題の規模を表すパラメータである。この問題の下位レベルの問題をカルーシュ・キューン・タッカー条件に置き換えることによって以下の問題を得る。

$$\begin{aligned} & \text{minimize} && \sum_{i=1}^k 2\sigma_i + \sum_{i=1}^k \tau_i \\ & \text{subject to} && \sum_{i=1}^k \sigma_i \leq \frac{3}{2}k \\ & && \tau_i \geq 0 \quad (i = 1, \dots, k) \\ & && -2\xi_{4i-3} + \xi_{4i-2} - \xi_{4i-1} + \xi_{4i} = 1 \quad (i = 1, \dots, k) \\ & && z_{4i-3} = 2\sigma_i - \tau_i + 2 \quad (i = 1, \dots, k) \\ & && z_{4i-2} = -\sigma_i - \tau_i + \rho_i \quad (i = 1, \dots, k) \\ & && z_{4i-1} = \sigma_i \quad (i = 1, \dots, k) \\ & && z_{4i} = -\sigma_i + 2 \quad (i = 1, \dots, k) \\ & && \xi_i \geq 0, \quad z_i \geq 0, \quad \xi_i z_i = 0 \quad (i = 1, \dots, 4k) \end{aligned} \quad (14)$$

ここで、 ξ_i ($i = 1, \dots, 4k$) は下位レベルの問題のラグランジュ乗数である。一方、これを混合相

表 1: 従来の手法 [2] と提案した手法の比較

k	従来の手法 [2]			提案手法		
	Dim	Time(s)	Iteration	Dim	Time(s)	Iteration
6	24	1	142	19	1	238
8	32	7	478	25	6	1182
10	40	51	2208	31	45	5800
12	48	222	7262	37	216	22896

補性条件をそのまま用いて定式化すると以下の問題が得られる.

$$\begin{aligned}
& \text{minimize} && \sum_{i=1}^k 2\sigma_i + \sum_{i=1}^k \tau_i \\
& \text{subject to} && \sum_{i=1}^k \sigma_i \leq \frac{3}{2}k \\
& && \tau_i \geq 0 \quad (i = 1, \dots, k) \\
& && -2\xi_{2i-1} + \xi_{2i} - \nu_i = 1 \quad (i = 1, \dots, k) \\
& && z_{2i-1} = 2\sigma_i - \tau_i + 2 \quad (i = 1, \dots, k) \\
& && z_{2i} = -\sigma_i - \tau_i + \rho_i \quad (i = 1, \dots, k) \\
& && \xi_i \geq 0, \quad z_i \geq 0, \quad \xi_i z_i = 0 \quad (i = 1, \dots, 2k) \\
& && \begin{cases} \sigma_i = 0 & \implies \nu_i \geq 0 \\ 0 < \sigma_i < 2 & \implies \nu_i = 0 \\ \sigma_i = 2 & \implies \nu_i \leq 0 \end{cases}
\end{aligned} \tag{15}$$

本実験では, 問題 (14) に対して [2] で提案された分枝限定法を適用した場合と, 問題 (15) に対して本論文で提案したアルゴリズムを適用した場合の比較を行った. なお, 問題の規模を表すパラメータ k は 6, 8, 10, 12 の 4 通りを考えた.

次に, 分枝限定法におけるノードの選択法と分枝操作を行う際の添字の選択法について説明する.

まず, [2] で提案された分枝限定法の場合, ノードの選択については $\bar{x}(J, K, L)^T \bar{z}(J, K, L)$ が最小となるものを選び, 添字の選択については $\bar{x}_i(J, K, L) \bar{z}_i(J, K, L)$ が最大となる添字を選択した.

次に, 本論文で提案したアルゴリズムに対するノードと添字の選択法について述べる. ノードの選択については,

$$\phi(x_i, z_i) := \begin{cases} \min\{(\bar{x}_i(J, K, L) - l_i + \max\{-z_i(J, K, L), 0\})|z_i(J, K, L)|, \\ (u_i - \bar{x}_i(J, K, L) + \max\{z_i(J, K, L), 0\})|z_i(J, K, L)|\} & (\text{if } l_i > -\infty, u_i < +\infty) \\ (\bar{x}_i(J, K, L) - l_i + \max\{-z_i(J, K, L), 0\})|z_i(J, K, L)| & (\text{if } l_i > -\infty, u_i = +\infty) \\ (u_i - \bar{x}_i(J, K, L) + \max\{z_i(J, K, L), 0\})|z_i(J, K, L)| & (\text{if } l_i = -\infty, u_i < +\infty) \\ 0 & (\text{if } l_i = -\infty, u_i = +\infty) \end{cases}$$

とおいたとき, $\sum_{i=1}^k \phi(x_i, z_i)$ が最小となるものを選んだ. また, 添字については, $\phi(x_i, z_i)$ が最大となる添字を選択した. このノードと添字の選択法は, 文献 [2] において相補性条件に対して提案された手法を混合相補性条件に対して拡張したものである.

表 1 に数値実験の結果を示す. 表中の Dim は問題に含まれる相補性条件と混合相補性条件の数を表している. Time はアルゴリズムが終了するまでに要した時間を秒単位で示し, Iteration はアルゴリズムが終了するまでの分枝限定法の反復数である.

表 1 から分かるように, 従来の手法に比べて提案手法は多くの反復を要するが, 問題の次元が小さくなることにより, ノード 1 つあたりの計算時間は少ないため, 全体の計算時間はわずかながら減少している.

分枝限定法では計算速度がノードを探索する順番や分枝する条件の順番などに大きく依存する。従来の手法では効率の良い選択法が提案されているのに対し、今回提案した手法では決定的な選択法がまだ見つかっていないので、提案したアルゴリズムにはまだ改善の余地が大きい。

4 結論

本論文では、混合相補性条件を制約にもつ数理計画問題に対する分枝限定法を提案し、そのアルゴリズムに対する数値実験の結果を報告した。その結果、ノードや添字の選択法にはまだ改善の余地が大きいにもかかわらず従来の手法と同等の性能を持つことが確かめられた。今後の課題としては、より多くの問題に対して数値実験を行うことによって、提案したアルゴリズムを多角的に評価することがまず挙げられる。また、探索するノードや分枝する条件の効果的な決定法についてもより工夫する必要がある。

参考文献

- [1] Al-Khayyal, F. A., An implicit enumeration procedure for the general linear complementarity problem, *Mathematical Programming Study* 31 (1987) 1-20.
- [2] Bard, J. F. and Moores, J. T., A branch and bound algorithm for the bilevel programming problem, *SIAM J. Sci. Stat. Comput.* Vol 11(1990) 281-292.
- [3] Facchinei, F., Jiang, H. and Qi, L., A smoothing method for mathematical programs with equilibrium constraints, *Math. Programming*, 85 (1999), 107-134.
- [4] Fukushima, M. and Pang, J.-S., Convergence of a smoothing continuation method for mathematical programs with complementarity constraints, in *Ill-posed Variational Problems and Regularization Techniques, Lecture Notes in Economics and Mathematical Systems*, Vol. 477, M. Thera and R. Tichatschke, eds., Springer Verlag, Berlin/Heidelberg, 1999.
- [5] Luo, Z.-Q., Pang, J.-S. and Ralph, D., *Mathematical Programs with Equilibrium Constraints*, Cambridge University Press, Cambridge, 1996.
- [6] Scholtes, S., Convergence properties of a regularization scheme for mathematical programs with complementarity constraints, Working paper, Judge Institute of Management Studies, University of Cambridge, Cambridge, UK, 1999.