2001年度日本オペレーションズ・リサーチ学会
春季研究発表会

均衡制約つき数理計画問題に対する分枝限定法について

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

均衡制約つき数理計画問題 (MPEC) は以下のように定式化される問題である.

$$\begin{array}{ll} \text{minimize} & d^Tx + f^Ty \\ \text{subject to} & Ax + By = c,\ y \geq 0 \\ & z = Mx + Ny + q \\ & x \geq 0,\ z \geq 0 \\ & x_iz_i = 0\ \ (i = 1, \cdots, m) \end{array} \tag{1}$$

ここで, $(x, y, z) \in \Re^m \times \Re^n \times \Re^m$ が決定変数であり, $A \in \Re^{r\times m}, B \in \Re^{r\times n}, M \in \Re^{m\times m}, N \in \Re^{m\times n}, c \in \Re^r, d \in \Re^m, f \in \Re^n, q \in \Re^m$ である. このMPECは2レベル線形計画問題などを含む広いクラスの問題である. 一方, MPECは相補性制約 $x_iz_i = 0\ (i = 1, \cdots, m)$ をもつため, 既存の数理計画の手法で解くことが困難な問題である.

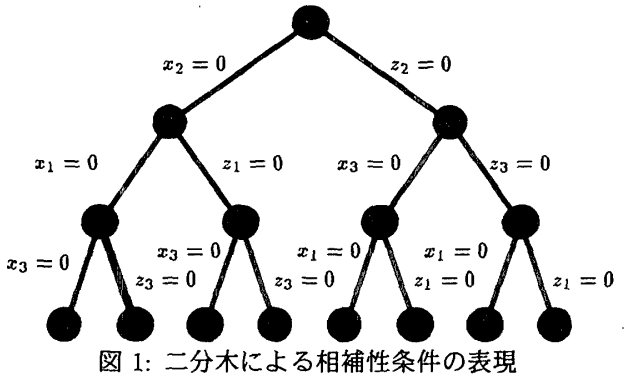

図 1: 二分木による相補性条件の表現

相補性条件はすべての $i = 1, \cdots, m$ に対して $x_i = 0$ または $z_i = 0$ となることと等価である. そこで, 各 i に対して $x_i = 0$ とするか $z_i = 0$ とするかについて二分木を考えることができる (図1). この二分木において, 各ノードはいくつかの i に対し $x_i = 0$ もしくは $z_i = 0$ と固定された状態に対応する. ここで, J は $x_i = 0$ と固定されている添字の集合を, L は $z_i = 0$ と固定されている添字の集合をそれぞれ表すものとする. そして, 各ノードを (J, L) と記すことにする. このとき, 二分木の葉の部分では $J \cup L = \{1, \cdots, m\}$ であり, MPECにおいて $x_iz_i = 0$ のかわりに $x_j = 0\ (j \in J), z_l = 0\ (l \in L)$ とした問題は線形計画問題となる. そのため, 2^m 個の線形計画問題を解くことでMPECの解を理論的には求めることができる.

Bard と Moores[1] は, MPECに対する分枝限定法を提案した. このアルゴリズムでは, 二分木の各ノードにおいて次の線形計画問題 $PP(J, L, c_x, c_z)$ を解いて, 実行可能領域の存在のチェックと下界値の計算を行なっている.

$$PP(J, L, c_x, c_z) \quad \begin{array}{ll} \text{minimize} & c_x^Tx + c_z^Tz \\ \text{subject to} & (x, y, z) \in S(J, L) \end{array}$$

ここで, 制約領域 $S(J, L)$ は

$$S(J, L) := \left\{ (x, y, z) \;\middle|\; \begin{array}{l} Ax + By = c \\ z = Mx + Ny + q \\ x_J = 0, z_L = 0 \\ x \geq 0, y \geq 0, z \geq 0 \end{array} \right\}$$

であり, (1) の制約において $x_iz_i = 0\ (i = 1, \cdots, m)$ のを $x_j = 0\ \ (j \in J)$, $z_l = 0\ \ (l \in L)$ で置き換えた領域を表している.

具体的には, それぞれ $PP(J, L, e_\alpha, 0)$ と $PP(J, L, 0, e_\alpha)$ を解いて, 最適値が0であるかどうかを調べることによって $S(J \cup \{\alpha\}, L)$ と $S(J, L \cup \{\alpha\})$ の実行可能性をチェックする. ただし, $\alpha \in J \cap L$ であり, e_α は α 番目の要素のみが1で他の要素はすべて0であるようなベクトルである.

しかし, $PP(J, L, e_\alpha, 0)$ と $PP(J, L, 0, e_\alpha)$ を直接扱うと, 実際にこれらの問題の最適解や最適値を求めなければならない. 一方, これらの問題の双対問題を扱うことにすれば, 目的関数値が0より大きくなる双対問題の実行可能解が得られた時点で $PP(J, L, e_\alpha, 0)$ や $PP(J, L, 0, e_\alpha)$ の最適値が0より大きくなることがわかる. そのため, 無駄なノードの計算をしないですむことが期待できる.

本発表では, 実行可能領域が存在するかチェックするために, $PP(J, L, e_\alpha, 0)$ と $PP(J, L, 0, e_\alpha)$ の双対問題を用いることを提案する.

2　双対問題を用いた実行可能性判定

$PP(J, L, c_x, c_z)$ の双対問題は,

$$DP(J, L, c_x, c_z) \quad \begin{array}{ll} \text{maximize} & c^Tu - q^Tv \\ \text{subject to} & A^Tu + M^Tv + s = c_x \\ & B^Tu + N^Tv + w = c_z \\ & s_{\bar{J}} \geq 0, v_{\bar{L}} \geq 0, w \geq 0 \end{array}$$

となる. さらに, 次の問題

$$DP2(J, L) \quad \begin{array}{ll} \text{maximize} & c^Tu - q^Tv \\ \text{subject to} & (u, v, s, w) \in D(J, L) \end{array}$$

を導入する．ただし，

$$D(J,L) := \left\{ (u,v,s,w) \;\middle|\; \begin{array}{l} A^T u + M^T v + s = 0 \\ B^T u + N^T v + w = 0 \\ s_{\bar{J}} \geq 0, v_{\bar{L}} \geq 0, w \geq 0 \end{array} \right\}$$

であり，$\bar{J} := \{1,\cdots,m\} \setminus J, \bar{L} := \{1,\cdots,m\} \setminus L$ である．

ここで，次の定理が成立することを示すことができる．

定理 1 $DP2(J \cup \{\alpha\}, L)$ の最適値が 0 であることと，$PP(J,L,e_\alpha,0)$ の最適値が 0 であることは等価である．同様に，$PP(J,L,0,e_\alpha)$ の最適値が 0 であることと $DP2(J,L \cup \{\alpha\})$ の最適値が 0 であることも等価である．

さらに，$DP2(J,L)$ に関して次のことを示すことができる．

定理 2 シンプレックス法によって問題 $DP2(J,L)$ の最適解 (u^*,v^*,s^*,w^*) が得られているとする．このとき以下が成立する．

(a) s^*_α が基底変数であれば (u^*,v^*,s^*,w^*) は $DP2(J \cup \{\alpha\}, L)$ の最適解でもあり，$DP2(J \cup \{\alpha\}, L)$ の最適値は 0 である．

(b) s_α に対応する相対コスト係数が 0 ならば，(u^*,v^*,s^*,w^*) は $DP2(J \cup \{\alpha\}, L)$ の最適解でもあり，$DP2(J \cup \{\alpha\}, L)$ の最適値は 0 である．

定理 1 を用いてシンプレックス法に基づいた次のアルゴリズムを提案する．このアルゴリズムにより得られる集合 J^*, L^* はそれぞれ次のようなものである．

J^* : $DP2(J \cup \{\alpha\}, L)$ の最適値が 0 より大きくなる α の集合

L^* : $DP2(J, L \cup \{\alpha\})$ の最適値が 0 より大きくなる α の集合

アルゴリズム (実行可能性の判定)

Step 0: $J^* := \emptyset, L^* := \emptyset$ とする．

Step 1: $\alpha \in \bar{J} \cap \bar{L}$ に対して s_α が基底変数であるような α の集合を $\bar{J}^*$，$\alpha \in \bar{J} \cap \bar{L}$ に対して v_α が基底変数であるような α の集合を $\bar{L}^*$ とする．Step 2 へ．

Step 2: $\alpha \in (\bar{J} \cap \bar{L}) \setminus \bar{J}^*$ に対して s_α に対する相対コスト係数が 0 となっていたら，$\bar{J}^* := \bar{J}^* \cup \{\alpha\}$ とする．同様に，$\alpha \in (\bar{J} \cap \bar{L}) \setminus \bar{L}^*$ に対して t_α に対する相対コスト係数が 0 となっていたら，$\bar{L}^* := \bar{L}^* \cup \{\alpha\}$ とする．Step 3 へ．

Step 3: $\alpha \in (\bar{J} \cap \bar{L}) \setminus \bar{J}^*$ に対して $DP2(J \cup \{\alpha\}, L)$ を解き，最適値が 0 より大きくなれば $J^* := J^* \cup \{\alpha\}$ とし，最適値が 0 ならば $\bar{J}^* := \bar{J}^* \cup \{\alpha\}$ とする．

同様に，$\alpha \in (\bar{J} \cap \bar{L}) \setminus \bar{L}^*$ に対して $DP2(J, L \cup \{\alpha\})$ を解き，最適値が 0 より大きくなれば $L^* := L^* \cup \{\alpha\}$ とし，最適値が 0 ならば $\bar{L}^* := \bar{L}^* \cup \{\alpha\}$ とする．

ここで，$\alpha \in J^*$ に対する $S(J \cup \{\alpha\}, L)$ と $\alpha \in L^*$ に対する $S(J, L \cup \{\alpha\})$ には実行可能領域が存在しないため，そのノードはカットすることができる．$PP(J,L,e_\alpha,0)$ や $PP(J,L,0,e_\alpha)$ を解くことでは，ある α に対する実行可能性しかチェックできなかった．一方，ここで提案したアルゴリズムを用いることによって $\bar{J} \cap \bar{L}$ に含まれるすべての添字に対する実行可能性を一度にチェックできる．このことにより，分枝限定法の高速化が期待できる．

3 提案する分枝限定法の概要

前節までの議論を用いれば，問題 (1) に対して次のような手続きを取り入れた分枝限定法が構成できる．

(a) 下界値の計算: ノード (J,L) の部分問題の下界値を，次の線形計画問題

$$R(J,L) \quad \begin{array}{ll} \text{minimize} & d^T x + f^T y \\ \text{subject to} & (x,y,z) \in S(J,L) \end{array}$$

を解くことによって求める．

(b) 実行可能性の判定: ノード (J,L) の部分問題に対して，次の手続きを適用する．

(b1): ある $i \in \bar{J} \cap \bar{L}$ に対して $i \in J^* \cap L^*$ が成り立てば，ノード (J,L) の部分問題は実行可能でないのでこのノードを終端する．そうでなければ，(b2) へ．

(b2): $(J^* \cap \bar{L}^*) \cup (\bar{J}^* \cap L^*) \neq \emptyset$ ならば，$J := J \setminus L^*, L := L \setminus J^*$ とおく．そうでなければ，$\alpha \in \bar{J} \cap \bar{L}$ を選び，新たなノード $(J \cup \{\alpha\}, L), (J, L \cup \{\alpha\})$ を生成する．

本稿では，部分問題の実行可能性の判定において双対問題を用いた分枝限定法の改良案を提案した．当日にはさらにいくつかのアイデアを導入したアルゴリズムの詳細とその数値実験の結果を発表する．

参考文献

[1] J. F. Bard and J. T. Moores, A branch and bound algorithm for the bilevel programming problem, SIAM J. Sci. Stat. Comput. Vol 11(1990) 281-292.