

MC-nets における利得分配：上界保証付き ε -コアを求めめるアルゴリズム

平山 勝敏 赤木 純 沖本 天太

本稿では、MC-nets のルール集合で記述された提携形ゲームの利得分配問題に対して、代表的な解概念の一つである ε -コアに属する利得ベクトルを求めるアルゴリズムを 2 つ提案する。1 つ目のアルゴリズム eCore1 は、MC-nets ルール集合と最大不満の最小値に対する上界保証値を入力とし、要求された ε -コアに属する利得ベクトルとその最大不満の値を返す、いわゆる事前保証付きアルゴリズムである。一方、もう一つのアルゴリズムである eCore2 は、MC-nets ルール集合と任意の提携集合を入力とし、 ε -コアに属する利得ベクトルとそれが達成する上界保証値を返す、いわゆる事後保証付きアルゴリズムである。本稿では、それぞれのアルゴリズムの理論的性質を示す。

1 準備

提携形ゲーム (coalitional game) は、エージェントの集合 $A = \{1, \dots, n\}$ 、および、エージェントの可能な部分集合 (提携 (coalition)) に対する利得を計算する特性関数 (characteristic function) $v: 2^A \rightarrow \mathbb{R}$ で構成される。

提携形ゲームにおける代表的な問題の 1 つに提携構造形成問題 (coalition structure generation problem) がある。提携構造 (coalition structure) とは、エージェント集合 A に対する分割であり、提携構造形成問題では、最大総利得 p_{\max} を達成する A の分割 CS^* を求める。この問題は NP 困難な問題に属する。

提携形ゲームにおけるもう 1 つの問題は利得分配問題 (payoff division problem) である。これは、最適な提携構造 CS^* を形成することにより獲得した総利

得 p_{\max} を各エージェントにどのように分配するかを決める問題である。以降、分配の結果、各エージェント i は x_i の利得を得るものとし、それを利得ベクトル $x = (x_1, \dots, x_n)$ で表す。利得分配問題に対してはいくつかの解概念が提案されている。そのうち、不満という概念に基づくコア (core)、 ε -コア (ε -core)、最小コア (least core)、および、仁 (nucleolus)、さらには、貢献度という概念に基づくシャプレイ値 (Shapley value) が良く知られた代表的な解概念である。本稿では最小コアとその近似解に相当する ε -コアを扱う。

1.1 ε -コアと最小コア

ある利得ベクトル $x = (x_1, \dots, x_n)$ に対し、 $v(S) > \sum_{i \in S} x_i$ を満たす部分提携 $S \subset A$ を阻止提携 (blocking coalition) という。阻止提携 S が存在すれば、 S のメンバーは利得ベクトル x に対して不満をもち、最適な提携構造 CS^* から離脱して独自に部分提携 S を作る動機をもつ。コアとは、阻止提携が存在しない利得ベクトルの集合 C であり、形式的には、以下の 2 つの条件式を満たす利得ベクトル x

Payoff Division in MC-nets: Algorithms for ε -core with Upper Bound Guarantee.

This is an unrefereed paper. Copyrights belong to the Authors.

Katsutoshi Hirayama, Tenda Okimoto, 神戸大学大学院海事科学研究科, Graduate School of Maritime Sciences, Kobe University.

Jun Akagi, 神戸大学海事科学部 (現在, 株式会社テクノアソシエ), Faculty of Maritime Sciences, Kobe University.

の集合である。

$$v(S) - \sum_{i \in S} x_i \leq 0, \forall S \subset A, \quad (1)$$

$$\sum_{i \in A} x_i = p_{\max}. \quad (2)$$

式 (1) の左辺は、部分提携 S の利得ベクトル x に対する不満 (excess) を表しており、式 (1) は不満の値が正となるような部分提携が存在しないことを意味する。ここで、式 (1) を

$$v(S) - \sum_{i \in S} x_i \leq \varepsilon, \forall S \subset A, \quad (3)$$

に置き換えることにより、コアを一般化した ε -コアが得られる。すなわち、 ε -コアとは、任意の部分提携において不満の値が ε 以下である利得ベクトルの集合 C^ε である。

一般に $\varepsilon' < \varepsilon$ のとき、 $C^{\varepsilon'} \subset C^\varepsilon$ が成立する。すなわち、 ε の値を小さくしていくと、 ε -コアは次第に小さくなり、やがて空となる。最小コアとは、すべての非空な ε -コアの共通部分 C^{ε^*} であり、以下の線形計画問題の最適解の集合である。

$$\begin{aligned} \min. \quad & \varepsilon \\ \text{s.t.} \quad & v(S) - \sum_{i \in S} x_i \leq \varepsilon, \forall S \subset A, \\ & \sum_{i \in A} x_i = p_{\max}. \end{aligned} \quad (4)$$

すなわち、最小コアとは、可能なすべての部分提携における不満の最大値 (最大不満) を最小化する利得ベクトルの集合である。また、(4) の線形計画問題の最適値を ε^* とすれば、 ε^* -コア (C^{ε^*}) が最小コアとなる。

1.2 MC-nets と価格問題

提携形ゲームでは、従来、特性関数はいわゆるブラックボックス関数とされ、エージェント間の可能な提携を与えれば値を返すものとされていた。近年、特性関数の簡略記述法として SCG [1] や MC-nets [4] 等が提案されている。

MC-nets では、提携が満たすべきルールの集合 R によって特性関数を表現する。各ルール $r \in R$ は、 $(P_r, N_r) \rightarrow v_r$ という形式で記述され、 P_r は存在しなければならないエージェントの集合、 N_r は存在してはならないエージェントの集合であり、 $P_r \cap N_r = \emptyset$ である。また、 $v_r \in \mathfrak{R}$ は、ルール r の条件部が満

たされた場合の利得である。ある提携 S について、 $P_r \subseteq S$ かつ $N_r \cap S = \emptyset$ のとき、ルール r は提携 S に適用可能であるという。 S に適用可能なルールの全体集合を R_S とするとき、任意の提携 S の利得は $v(S) = \sum_{r \in R_S} v_r$ で与えられる。

MC-nets の利点は、任意の特性関数を表現でき、また、もう一つの簡略記述法である SCG に比べて、個々の提携の特性関数値の計算、および、利得分配問題に対するシャプレイ値の計算が高速に行えることにある [4]。一方、欠点は、利得分配問題に対するコア関係の計算に弱く、例えば MC-nets を入力としたコア非空性判定問題は coNP 完全であることが示されている [2][6]。

MC-nets におけるコアのアルゴリズムに関する研究は非常に少ない。MC-nets を最初に提案した論文である [4] では、コアのメンバーシップおよび非空性を判定するアルゴリズムのアイデアのみが示されている。一方、我々は、MC-nets を入力とし、制約生成法 (列生成法) を用いてコア非空性判定問題を解くアルゴリズム [5]、および、最小コアに属する利得ベクトルを求めるアルゴリズム [3] を提案し、それぞれ実装および評価を行った。

文献 [3] のアルゴリズムの背景となった理論的成果は、線形計画問題 (4) に対して、利得ベクトル x と ε の任意の値の実行可能性を MC-nets のルールの情報のみを用いて判定する仕組みを考案した点である。

利得ベクトルの値 $\hat{x} = (\hat{x}_1, \dots, \hat{x}_n)$ (ただし、 $\sum_{i=1}^n \hat{x}_i = p_{\max}$ とする) と ε の値 $\hat{\varepsilon}$ が、(4) の実行可能解か否かを判定する際、素朴な方法では $2^n - 1$ 個の制約式を全てチェックする必要があるが、考案した方法では、以下の整数計画問題 (価格問題) の最適値 z^* を求め、 $z^* \leq \hat{\varepsilon}$ ならば実行可能と判定する。

$$\begin{aligned} \max. \quad & \sum_{r \in R} v_r \beta_r - \sum_{i \in A} \hat{x}_i \alpha_i \\ \text{s.t.} \quad & \sum_{i \in P_r} \alpha_i + \sum_{i \in N_r} (1 - \alpha_i) \geq |P_r \cup N_r| \beta_r, \forall r \in R^+, \\ & \sum_{i \in P_r} (1 - \alpha_i) + \sum_{i \in N_r} \alpha_i \geq 1 - \beta_r, \forall r \in R^-, \\ & \sum_{i \in A} \alpha_i \leq n - 1, \\ & \sum_{i \in A} \alpha_i \geq 1, \\ & \alpha_i, \beta_r \in \{0, 1\}, \forall i \in A, \forall r \in R. \end{aligned} \quad (5)$$

ここで、価格問題 (5) における各変数の意味は次の通りである。

- A : エージェントの集合 ($|A| = n$)
- R : MC-nets ルールの集合
- R^+ : 利得が正である MC-nets ルールの集合
- R^- : 利得が負である MC-nets ルールの集合
- P_r : MC-nets ルール r の条件部で指定された存在しなければならないエージェントの集合
- N_r : MC-nets ルール r の条件部で指定された存在してはならないエージェントの集合
- α_i : エージェント i ($i \in A$) に対応した 0-1 決定変数
- β_r : MC-nets ルール r ($r \in R$) に対応した 0-1 決定変数

2 提案アルゴリズム

2.1 最小コアを求める既存アルゴリズム

制約生成法では、制約の数が潜在的に膨大となる大規模な線形計画問題に対し、一部の制約からなる問題 (制限されたマスター問題) からスタートし、必要な制約を順次追加しながら暫定的な解を繰り返し求め、求めた暫定解がある条件 (主実行可能性) を満たすときそれを最適解として返す。なお、主実行可能性の判定、および、それが成立しない場合に追加する制約の決定において、前述の価格問題が重要な役割を果たす。

文献 [3] では、次のようなアルゴリズム LeastCore を提案した。

Algorithm LeastCore(R)

Input: MC-nets のルール集合 R
Output: 最小コアに属する利得ベクトル x^* , 最大不満の最小値 ε^*

Step 1. 提携集合 \mathcal{T} の初期値をすべての可能な単独提携の集合とする。

Step 2. 次の線形計画問題 (制限されたマスター問題) の最適解 $(\hat{x}, \hat{\varepsilon}) = (\hat{x}_1, \dots, \hat{x}_n, \hat{\varepsilon})$ を求める。

$$\begin{aligned} \min. \quad & \varepsilon \\ \text{s.t.} \quad & v(S) - \sum_{i \in S} x_i \leq \varepsilon, \forall S \in \mathcal{T}, \quad (6) \\ & \sum_{i \in A} x_i = p_{max}, \end{aligned}$$

Step 3. $(\hat{x}_1, \dots, \hat{x}_n)$ を用いて価格問題 (5) を解き、その最適値 z^* と最適解の一部である $\alpha^* = (\alpha_1^*, \dots, \alpha_n^*)$ を求める。

Step 4. $z^* \leq \hat{\varepsilon}$ であれば、 \hat{x} が主実行可能性を満たすため、 $x^* \leftarrow \hat{x}$ および $\varepsilon^* \leftarrow \hat{\varepsilon}$ とし、それぞれを出力して終了する。そうでなければ提携 $S^* = \{i \mid \alpha_i^* = 1, \forall i \in A\}$ を構成し、提携集合 \mathcal{T} に提携 S^* を追加する。Step 2 に戻る。

2.2 近似アルゴリズム eCore1

本研究では、新しい近似アルゴリズムを得るために、文献 [3] のアルゴリズムの終了条件を一般化する。まず、以下の補題 1 および 2 を証明なしで与える。

補題 1 制限されたマスター問題 (6) の最適値 $\hat{\varepsilon}$ は、最小コアを求める線形計画問題 (4) の最適値 (最大不満の最小値) ε^* の下界を与える。

補題 2 価格問題 (5) の最適値 z^* は、最小コアを求める線形計画問題 (4) の最適値 (最大不満の最小値) ε^* の上界を与える。

以上の 2 つの補題を利用して次のような近似アルゴリズム eCore1 が構成できる。

Algorithm eCore1($R, bound$)

Input: MC-nets のルール集合 R , 上界保証値 $bound$ (≥ 0)
Output: ε -コアに属する利得ベクトル x , 最大不満の値 ε (ただし $\varepsilon - \varepsilon^* \leq bound$)

Step 1. 提携集合 \mathcal{T} の初期値をすべての可能な単独提携の集合とする。

Step 2. 制限されたマスター問題 (6) の最適解 $(\hat{x}, \hat{\varepsilon}) = (\hat{x}_1, \dots, \hat{x}_n, \hat{\varepsilon})$ を求める。

Step 3. $(\hat{x}_1, \dots, \hat{x}_n)$ を用いて価格問題 (5) を解き、その最適値 z^* と最適解の一部である $\alpha^* = (\alpha_1^*, \dots, \alpha_n^*)$ を求める。

Step 4. $z^* - \hat{\varepsilon} \leq bound$ が成立すれば、 $x \leftarrow \hat{x}$ および $\varepsilon \leftarrow z^*$ とし、それぞれを出力して終了する。そうでなければ提携 $S^* = \{i \mid \alpha_i^* = 1, \forall i \in A\}$ を構成し、提携集合 \mathcal{T} に提携 S^* を追加する。Step 2 に戻る。

このアルゴリズムについて、以下が成立する。

定理 1 (正当性) アルゴリズム eCore1 は z^* -コアに属する利得ベクトルを返す。その最大不満の値 z^* とその最小値 ε^* との差は $bound$ 以下である。

証明: 補題 1 と 2 が成り立つことと、同アルゴリズムの Step 4 の終了条件から自明に導ける。 \square

定理 2 (停止生) アルゴリズム eCore1 は有限時間内

で停止する.

証明：可能な提携の数は高々 $2^n - 1$ 個であるため，Step 2 から 4 の繰り返しの過程において，提携 S^* として毎回異なる提携が選択されて提携集合 \mathcal{T} に追加されるならば，制限されたマスター問題 (6) はいずれも最小コアを求める線形計画問題 (4) に一致し，その際， $z^* = \hat{\varepsilon}$ となるため停止する．従って，Step 4 において，すでに \mathcal{T} の要素である提携が S^* として選択されることはないという事実 ($S^* \notin \mathcal{T}$) を示せば充分である． S^* は利得ベクトル \hat{x} に対して最大の不満をもつ提携であり，その不満の値 $v(S^*) - \sum_{i \in S^*} \hat{x}_i$ が価格問題 (5) の最適値 z^* と一致する．今， S^* が構成されたため， $z^* - \hat{\varepsilon} > bound$ ，すなわち，

$$v(S^*) - \sum_{i \in S^*} \hat{x}_i > \hat{\varepsilon} + bound$$

が成立している．一方， $S^* \in \mathcal{T}$ だと仮定すると， \hat{x} と $\hat{\varepsilon}$ に対して制限されたマスター問題 (6) の制約条件を当然満たすため，

$$v(S^*) - \sum_{i \in S^*} \hat{x}_i \leq \hat{\varepsilon}$$

が成立するはずである．よって矛盾が発生し， $S^* \notin \mathcal{T}$ と結論できる． \square

なお，アルゴリズム eCore1 において $bound$ の値を 0 に設定すれば，最小コアを求めるアルゴリズム LeastCore と同一になる．すなわち，eCore1 は，従来アルゴリズムである LeastCore を特殊ケースとして含む．

2.3 近似アルゴリズム eCore2

アルゴリズム eCore1 は，制限されたマスター問題 (6) と価格問題 (5) を繰り返し解くことにより，最大不満の値 ε がその最小値 ε^* から $bound$ 以内であることが保証された ε -コア (に属する利得ベクトル) を求める．一方で，価格問題 (5) は整数計画モデルで記述された組合せ最適化問題であり，特にエージェント数 n が大きくなると，これを解くための 1 回あたりの計算コストが増大するとともに，繰り返しの回数も増大すると考えられる．

そこで，任意の提携集合 \mathcal{T} に対する制限されたマスター問題 (6) を 1 回だけ解いて得た利得ベクトル \hat{x} を返す単純な近似アルゴリズム eCore2 を新たに導入

する．

Algorithm eCore2(R, \mathcal{T})

Input: MC-nets のルール集合 R ，提携集合 \mathcal{T}
Output: 利得ベクトル x ，最大不満の値 ε とその最小値 ε^* との差の上界 $bound$

Step 1. 制限されたマスター問題 (6) の最適解 $(\hat{x}, \hat{\varepsilon}) = (\hat{x}_1, \dots, \hat{x}_n, \hat{\varepsilon})$ を求める．

Step 2. $(\hat{x}_1, \dots, \hat{x}_n)$ を用いて価格問題 (5) を構成し，各決定変数 α_i ($\forall i \in A$) と β_r ($\forall r \in R$) の取り得る値を 0 以上 1 以下の実数値に緩和した線形緩和問題を

$$\begin{aligned} \max. & \sum_{r \in R} v_r \beta_r - \sum_{i \in A} \hat{x}_i \alpha_i \\ \text{s.t.} & \sum_{i \in P_r} \alpha_i + \sum_{i \in N_r} (1 - \alpha_i) \geq |P_r \cup N_r| \beta_r, \forall r \in R^+, \\ & \sum_{i \in P_r} (1 - \alpha_i) + \sum_{i \in N_r} \alpha_i \geq 1 - \beta_r, \forall r \in R^-, \\ & \sum_{i \in A} \alpha_i \leq n - 1, \\ & \sum_{i \in A} \alpha_i \geq 1, \\ & 0 \leq \alpha_i \leq 1, \forall i \in A, \\ & 0 \leq \beta_r \leq 1, \forall r \in R. \end{aligned} \quad (7)$$

を作って，その最適値 l^* を求める．

Step 3. $x \leftarrow \hat{x}$ および $bound \leftarrow l^* - \hat{\varepsilon}$ とし，それぞれを出力して終了する．

定理 3 アルゴリズム eCore2 が返す利得ベクトルについて，その最大不満の値 ε と最小値 ε^* の差は $l^* - \hat{\varepsilon}$ 以下である．

証明：問題 (7) は価格問題 (5) の線形緩和ゆえ $l^* \geq z^*(= \varepsilon)$ ，補題 2 より $z^* \geq \varepsilon^*$ ，補題 1 より $\varepsilon^* \geq \hat{\varepsilon}$ ，よって $\varepsilon - \varepsilon^* \leq l^* - \hat{\varepsilon}$ が言える． \square

このアルゴリズムの特徴は，利得ベクトル \hat{x} とともに， \hat{x} に対する最大不満の値とその最小値 ε^* との差の上界を付帯情報として返す点である．また，Step 2 において，価格問題 (5) そのものでなく，その線形緩和問題を解く点も特筆すべきである．これにより，eCore2 は線形計画問題を 2 回解いて終了するため，計算速度の大幅な向上が期待できる．

3 おわりに

本稿では，MC-nets のルール集合で記述された提携形ゲームの利得分配問題に対して，代表的な解概念の一つである ε -コアに属する利得ベクトルを求めるアルゴリズムを提案した．今後の課題は，両アルゴリ

ズムの実際の性能を評価する実験を行うことである。

参考文献

- [1] Conitzer, V., Sandholm, T.: Complexity of constructing solutions in the core based on synergies among coalitions. *Artificial Intelligence*, 170, pp.607–619, 2006.
- [2] Greco, G., Malizia, E., Palopoli, L., Scarcello, F.: On the complexity of core, kernel, and bargaining set. *Artificial Intelligence*, 175(12–13), pp.1877–1910, 2011.
- [3] Hirayama, K., Hanada, K., Ueda, S., Yokoo, M., Iwasaki, A.: Computing a payoff division in the least core for MC-nets coalitional games *PRIMA-2014*, pp.319–332, 2014.
- [4] Yeung, S., Shoham, Y.: Marginal contribution nets: A compact representation scheme for coalitional games. *Proc. of the 6th ACM conference on Electronic Commerce (EC-2005)*. pp. 193–202, 2005.
- [5] 神谷 竜平, 花田 研太, 平山 勝敏: 列生成法を用いた提携形ゲームのコア非空性判定アルゴリズム, 2013 年度人工知能学会全国大会 (第 27 回) (JSAI-2013) 講演論文集, 2013.
- [6] Malizia, E., Palopoli, L., Scarcello, F.: Infeasibility certificates and the complexity of the core in coalitional games. *IJCAI-2007*. pp.1402–1407, 2007.