2D4-4

# 外部性が存在する提携ゲームのための提携構造形成アルゴリズムの提案

Coalitional Structure Generation Algorithm for Coalition Games with Externalities

野本 一貴 Kazuki Nomoto　伊原 尚正 Takamasa Ihara　櫻井 祐子 Yuko Sakurai　横尾 真 Makoto Yokoo

九州大学 システム情報科学府
Graduate School of Information Science and Electrical Engineering, Kyushu University

Coalitional games with externalities have been attracting considerable attention from the AI research community. Traditionally, the input of a coalitional game with externalities is a black-box function called a partition function. Previous works have found that many problems in coalitional games with externalities tend to be computationally intractable in this black-box function representation. As such a representative problem, Coalition Structure Generation (CSG) problem is for partitioning a set of agents into coalitions so that the sum of the rewards of all coalitions is maximized. Recently, a novel concise representation scheme called the Partition Decision Trees (PDT) have been proposed. In this paper, we develop two methods for solving a CSG problem reasonably well by utilizing a PDT. First, we propose an efficient algorithm by applying a branch-and-bound algorithm. Next, we provide an IP formalization and solve it by using a general-purpose mixed integer programming package CPLEX.

## 1. 序論

協力ゲーム理論は，利己的に行動するエージェント間で拘束力のある合意が可能な場合，すなわち提携を形成可能な場合のエージェントの振る舞いに関する理論である．利己的なエージェントが協力関係を結び，提携を形成する場合に解くべき問題の 1 つに，提携構造形成問題 (CSG, Coalition Structure Generation) がある．これは，全員で協力することが最適ではない場合に，提携のもたらす効用 (嬉しさ) の総和が最大となるようにエージェントの集合を分割する問題である．

従来の協力ゲームの研究では，提携のもたらす効用は特性関数と呼ばれるブラックボックスの関数 (オラクル) によって与えられることを仮定していた．特性関数の記述量は，エージェント数を $n$ とすると $\Theta(2^n)$ と非常に大きく，多数のエージェントが存在する提携ゲームでは，現実的な時間で提携構造形成問題の解を発見することは困難である．通常，特性関数の構造には何らかの特徴が存在するため，その特徴を利用して簡潔に記述することが可能である．近年，マルチエージェントシステムの研究者らを中心に，特性関数の特徴を利用した特性関数の簡潔記述法の提案が行われている [Conitzer 06, Ieong 05, Ueda 11]．

現実に即した提携ゲームでは，外部性が存在する場合がある．外部性とは，自身が含まれる提携の外部で形成される提携によって，自身が含まれる提携のもたらす効用が変化する性質である．例えば，競合企業 $a$，$b$，$c$ が存在するとき，$b$ と $c$ が協力関係を結ぶと，$b$ と $c$ が単独で行動する場合と比較して $a$ の売り上げが低下するような状況が考えられる．外部性の存在する提携構造形成問題において，提携のもたらす効用は提携構造と提携を引数とする分割関数と呼ばれる関数によって与えられる．分割関数の記述量は $O(n^n)$ であるため，特性関数と比較して記述量が非常に大きい．したがって，分割関数を簡略に記述するための研究が存在している [Michalak 10, Skibski 15]．

Michalak らは，外部性の存在するゲームを簡略に記述する組み込み MC-nets (Embedded MC-nets) と呼ばれる手法を提案した [Michalak 10]．また，Skibski らは分割関数を分割決定木 (PDT, Partition Decision Trees) と呼ばれる，複数の多分木によってグラフで簡潔に表現する記述法の提案を行った [Skibski 15]．多分木において，内部ノードはエージェントを示し，各枝にはエージェントが含まれる提携の番号が付与されている．根ノードから葉ノードまでの経路は分割を示し，葉ノードはその分割がもたらす効用を示す．

外部性が存在する場合の提携構造形成問題に対して，一村らは，組み込み MC-nets によってゲームが記述されている場合，最適な提携構造を求めるアルゴリズムを提案した [一村 11]．ここでは，提携構造形成問題を混合整数計画問題として定式化しているが，問題が複雑になるにしたがって制約条件が増加するため，現実的な時間内で求解可能な問題の規模に限界があった．

そこで，本論文では，分割決定木を用いた記述法による提携構造形成問題に対するアルゴリズムの提案を行う．初めに，分枝限定法を利用した提携構造形成アルゴリズムの提案を行う．次に，提案するアルゴリズムと比較する対象として，分割決定木による提携構造形成問題を整数計画問題として定式化する．ここでは，1 つの多分木内では高々1 つの分割しか選択されないこと，異なる多分木間で同時に選択できない分割が存在することを制約条件として与えることで定式化を行っている．また，分枝限定法では，求解に必要な計算時間は枝刈り手法によって大きく影響を受ける．1 つの多分木内では，高々1 つの分割しか選択されないこと，異なる多分木間では加法性が満たされることを利用した枝刈りを行うことで，効率的に求解可能であることを計算機実験によって示す．

## 2. モデル

エージェントの全体集合を $A$ としたとき，協力関係にあるエージェントの集合 $S \subseteq A$ を提携と呼ぶ．外部性の存在しない協力ゲームでは，提携のもたらす効用は特性関数によって与えられる．

**定義 1 (特性関数)** 特性関数 $v : 2^A \to \mathbb{R}$ は，任意の提携 $S \subseteq A$ に含まれるエージェントが協力してもたらす効用を与える

連絡先: 野本一貴，九州大学大学院システム情報科学府，812-0395 福岡県福岡市西区元岡 744 番地，(092)802-3576，nomoto@agent.inf.kyushu-u.ac.jp

写像である．

提携構造形成問題では，エージェントの全体集合をいくつかの提携に分割する．エージェントの全体集合の分割方法 $CS = \{S_1, S_2, \ldots\}$ を提携構造と呼び，以下を満たす．

$$\forall i, j\ (i \neq j),\ S_i \cap S_j = \emptyset,\ \bigcup_{S_i \in CS} S_i = A.$$

また，エージェントの部分集合の分割方法を分割と呼ぶ．提携構造 $CS$ のもたらす効用 $V(CS)$ は，$CS$ に含まれるすべての提携がもたらす効用の総和，すなわち $V(CS) = \sum_{S_i \in CS} v(S_i)$ となる．提携構造形成問題における解とは，エージェントの全体集合 $A$ から形成されうる提携構造の全体集合を $\Psi(A)$ とするとき，以下を満たす最適な提携構造 $CS^*$ を発見することである．

$$\forall CS \in \Psi(A), V(CS^*) \geq V(CS).$$

提携構造 $CS$ との組によって表現される提携 $S \in CS$ を $(S, CS)$ と表し，全体集合は $M := \{(S, CS) \mid CS \in \Psi(A), S \in CS\}$ で表される．外部性の存在する協力ゲームでは，提携のもたらす効用は分割関数によって与えられる．

**定義 2 (分割関数)** 分割関数 $\omega : M \rightarrow \mathbb{R}$ は，任意の提携構造 $CS$ の下での任意の提携 $S \in CS$ がもたらす効用を与える写像である．

**例 1** エージェントの全体集合を $A = \{a, b, c, d\}$ とする．分割関数によって，各提携構造に含まれる各提携のもたらす効用は以下のように与えられる．

| | |
|---|---|
| $\{(\{a,b,c,d\},15)\}$, | $\{(\{a\},5)(\{b\},2)(\{c\},3)(\{d\},2)\}$, |
| $\{(\{d\},2)(\{a,b,c\},12)\}$, | $\{(\{a\},9)(\{b\},2)(\{c,d\},5)\}$, |
| $\{(\{c\},2)(\{a,b,d\},12)\}$, | $\{(\{c\},2)(\{d\},2)(\{a,b\},10)\}$, |
| $\{(\{b\},0)(\{a,c,d\},5)\}$, | $\{(\{a\},5)(\{c\},3)(\{b,d\},4)\}$, |
| $\{(\{a\},7)(\{b,c,d\},14)\}$, | $\{(\{b\},0)(\{d\},2)(\{a,c\},2)\}$, |
| $\{(\{a,d\},5)(\{b,c\},12)\}$, | $\{(\{a\},3)(\{d\},2)(\{b,c\},12)\}$, |
| $\{(\{a,c\},2)(\{b,d\},2)\}$, | $\{(\{b\},2)(\{c\},3)(\{a,d\},7)\}$, |
| $\{(\{a,b\},14)(\{c,d\},4)\}$. | |

本来，分割関数は $w(\{a\}, \{\{a\}, \{b,c,d\}\}) = 7$ や $w(\{b,c,d\}, \{\{a\}, \{b,c,d\}\}) = 14$ のように表記するが，ここでは $\{(\{a\},7)(\{b,c,d\},14)\}$ のように略記している．提携構造 $\{\{a\}, \{b,c,d\}\}$ における提携 $\{a\}$ のもたらす効用は，タプル $(\{a\},7)$ で表されており，その値は 7 である．また，提携構造 $\{\{a\}, \{b,c,d\}\}$ が最適な提携構造であり，その効用は $7 + 14 = 21$ である．

例 1 のゲームには，外部性が存在する．例えば，提携 $\{a\}$ に注目すると，提携構造が $\{\{a\},\{b\},\{c\},\{d\}\}$ のとき効用は 5 であるが，$\{\{a\},\{b\},\{c,d\}\}$ のときは 9 と増加している．これは，外部に提携 $\{c,d\}$ が存在することが提携 $\{a\}$ に正の外部性を及ぼしていることを意味している．

## 3. 分割決定木

Skibski らは，分割関数を簡潔に記述する手法として，分割決定木 (PDT, Partition Decision Trees) を提案した [Skibski 15]．これは，分割関数を複数の多分木として簡潔に表現する手法である．本論文では，分割決定木による提携構造形成問題を扱う．

**定義 3 (分割決定木)** 分割決定木では，提携ゲームを複数の多分木として表現する．多分木において，内部ノードはエージェントを表すラベルが付与され，各枝にはその枝が出ているノードのエージェントが含まれる提携を表すラベルが付与されている．葉ノードは，根ノードからその葉ノードまでの経路上に存在するエージェントらの分割がもたらす効用を表す．各多分木は，タプルを用いて $T = (V, E, x, f_V, f_E)$ で表される．ここで，$V$ はノードの集合，$E \subseteq V \times V$ は枝の集合を表し，$x$ は根ノードを表す．また，$f_V : V \rightarrow A \cup \mathbb{R}^{\mathbb{N}}$ はノードのラベルに関する関数，$f_E : E \rightarrow \{1, 2, \ldots, |A|\}$ は枝のラベルに関する関数である．ただし，内部ノード $v$ に対しては $f_V(v) \in A$，葉ノード $v$ に対しては $f_V(v) \in \mathbb{R}^{\mathbb{N}}$ であるとする．

多分木 $T$ が与えられたとき，根ノードから葉ノードまでの経路の集合を $\Pi(T)$ で表す．また，根ノードから葉ノードまでの経路を $\pi = (v_1, v_2, \ldots, v_k) \in \Pi(T)$ で表す．経路 $\pi$ は，以下の 3 つの性質を満たす．

1. 経路 $\pi$ 上に存在する各内部ノードは，異なるエージェントを示すラベルが付与される．
   $|\{f_V(v_1), f_V(v_2), \ldots, f_V(v_{k-1})\}| = k - 1$
2. 経路 $\pi$ 上に存在する枝のラベルの集合は，1 から始まる連続する自然数の集合で表される．
   $\forall i (1 \leq i < k), f_E(v_i, v_{i+1}) \leq \max_{1 \leq j < i} f_E(v_j, v_{j+1}) + 1$
3. 経路 $\pi$ 上に存在する葉ノードのラベルのサイズは，経路 $\pi$ の表す分割に含まれる提携数に等しい．
   $|f_V(v_k)| = \max_{1 \leq j < k} f_E(v_j, v_{j+1})$

各経路 $\pi$ がそれぞれ異なる分割を表すために，同じノードから出ている各枝には異なるラベルが付与される．

**定義 4 (分割決定木の充足性)** 提携構造 $CS$ が経路 $\pi = (v_1, v_1, \ldots, v_k) \in \Pi(T)$ に対応することを，$CS \sim \pi$ で表す．各経路 $\pi$ がそれぞれ異なる分割を表すため，提携構造 $CS$ に対応する経路 $\pi$ は，各多分木に高々 *1* つ存在する．もし，$CS \sim \pi$ であれば，提携構造 $CS$ から，経路 $\pi$ 上に存在する枝のラベルの集合に 0 を加えた集合への写像 $g_\pi^{CS} : CS \rightarrow \{1, 2, \ldots, \max_{1 \leq j < k} f_E(v_j, v_{j+1})\} \cup \{0\}$ が存在する． ここで，0 は，経路 $\pi$ 上に存在しないエージェントらによって形成される提携を表す．

$g_\pi^{CS}$ は，提携構造 $CS$ に含まれる提携 $S$ を入力として，経路 $\pi$ が表す分割内で対応する提携を表すラベルを返す．提携構造 $CS$ に含まれる提携 $S$ のもたらす効用 $\omega_\pi^{CS}(S)$ は，$g_\pi^{CS}(S)$ を用いて以下の式で表される．

$$\omega_\pi^{CS}(S) = \begin{cases} f_V(v_k)[g_\pi^{CS}(S)] & \text{if } g_\pi^{CS}(S) > 0, \\ 0 & \text{otherwise.} \end{cases}$$

多分木間には，加法性が存在する．加法性とは，各多分木から 1 つまで経路 $\pi$ を選択し，それらが表す分割を 1 つの提携構造に統合することで任意の提携構造が構成される性質である．多分木の集合 $\mathbb{T} = \{T_1, T_2, \ldots\}$ で表現される提携ゲーム $y^{\mathbb{T}}$ における, 提携構造 $CS$ に含まれる提携 $S$ のもたらす効用 $y^{\mathbb{T}}(S, CS)$ は以下の式で表される．

$$y^{\mathbb{T}}(S, CS) := \sum_{T \in \mathbb{T}} y^T(S, CS) = \sum_{T \in \mathbb{T}} \sum_{\pi \in \Pi(T), CS \sim \pi} \omega_\pi^{CS}(S).$$

多分木 $T$ のサイズは，各ノードに付与されたラベルのサイズの総和で定義される．

**例 2** 例 *1* の分割関数は，分割決定木を用いて図 *1* のように簡潔に記述できる．例えば，*Tree 1* において，左端の経路 $\pi$ は，エージェント $a$ と $b$ が同じ提携に含まれる場合，その提携がもたらす効用に 10 が加えられることを示す．また，左から *2* 番目の経路 $\pi$ は，エージェント $b$ と $c$ が同じ提携に含まれ，エージェント $a$ が異なる提携に含まれる場合，$a$ が含まれる提携のもたらす効用に 3，$b$ と $c$ が含まれる提携のもたらす効用に 10 が加えられることを示す．

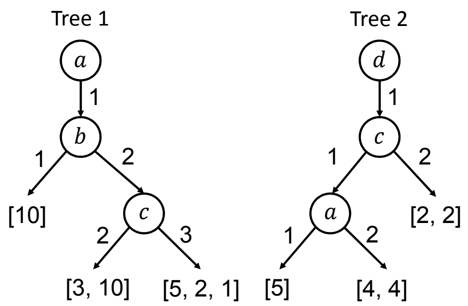


図 1: 分割決定木による記述例

## 4. 分枝限定法を用いたアルゴリズム

本章では，分枝限定法に基づいた提携構造形成アルゴリズムを提案する．具体的には，多分木間の加法性，すなわち各多分木から 1 つまで経路 $\pi$ を選択し，それらが表す分割を 1 つの提携構造に統合することで最適な提携構造が構成される性質を利用し，複数の分割の統合可能性を判断しつつ最適な提携構造を探索する提携構造形成アルゴリズムを提案する．

提案する提携構造形成アルゴリズムでは，各多分木の各経路 $\pi$ が表す分割を内部ノードとした探索木を深さ優先探索で探索する．探索木の根ノードから葉ノードまでの経路上には，各多分木の経路 $\pi$ が表す分割が存在する．ただし，最適な提携構造は必ずしも各多分木の経路 $\pi$ に対応することはなく，最適な提携構造に対応する経路 $\pi$ が存在しない多分木が存在する場合がある．そのような場合を表現するために探索木に空ノードを加える．また，各多分木における経路 $\pi$ が表す分割の探索順序は，分割がもたらす効用の降順になるようにソートしておく．葉ノードは，根ノードからその葉ノードまでの経路上に存在する各分割がもたらす効用の総和を表す．

**例 3** 例 *2* の提携ゲームから生成される探索木を図 *2* に示す．図 *2* では，特に *1* つ目の探索木を示している．左端の根ノードから葉ノードまでの経路は，*Tree 1* から分割 $\{\{a\},\{b,c\}\}$ を，*Tree 2* から分割 $\{\{c,d\},\{a\}\}$ を選択することを表す．また，葉ノードは，選択した *2* つの分割に対応する提携構造が存在すれば，その提携構造のもたらす効用は 21 であることを表す．

探索木の深さ優先探索においてノードを探索するごとに，根ノードから現在探索しているノードまでの経路上に存在する分割の統合可能性を判断する．統合不可能と判断されれば，その経路上の分割に対応する提携構造は存在しないことがわかる．したがって，その経路を含む根ノードから葉ノードまでの経路上のノードは探索しないという枝刈りを行う．統合可能であると判断されれば，探索を続行する．また，ある探索木の探索が終了し次の探索木の探索を開始するときに，次の探索木で得

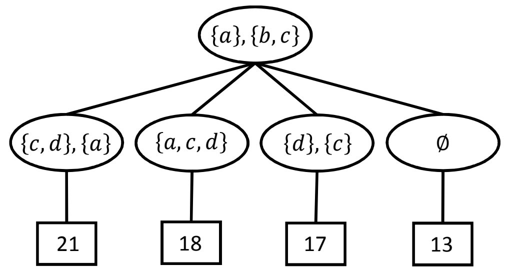


図 2: 探索木

られる最大の効用が既に得られている効用を上回らなければ，次の探索木の探索を行わないという枝刈りを行う．これらの 2 つの枝刈りを行いつつ，経路上の分割が統合可能，かつ葉ノードの表す効用が最大である経路を発見する．

複数の分割が 1 つの提携構造に統合可能であるかを判断する方法としては，まずは各エージェントの単独提携で構成される提携構造 $G$ を生成する．次に，各分割において同じ提携に含まれるエージェントの $G$ での提携を統合する．最後に，各分割において異なる提携に含まれるエージェント間で $G$ での提携が異なっていることを確認し，異なっていなければ統合不可能であると判断する．ただし，同じ提携に含まれるエージェントの $G$ での提携は統合しているため，各提携の 1 番目のエージェントについてのみ確認すればよい．最後まで確認できれば，統合可能であると判断する．

**例 4** 例 *2* の探索木の左端の経路上の分割 $\{\{a\},\{b,c\}\}$ と $\{\{c,d\},\{a\}\}$ の統合可能性を判断する．まずは，$G : \{\{a\},\{b\},\{c\},\{d\}\}$ を生成する．次に，分割 $\{\{a\},\{b,c\}\}$ の提携 $\{b,c\}$ から，$b$ と $c$ の提携を統合して，$G : \{\{a\},\{b,c\},\{d\}\}$ となる．同様に，分割 $\{\{c,d\},\{a\}\}$ の提携 $\{c,d\}$ から，$c$ と $d$ の提携を統合して，$G : \{\{a\},\{b,c,d\}\}$ となる．最後に，分割 $\{\{a\},\{b,c\}\}$ で $a$ と $b$ が別の提携に含まれているため，$G$ でエージェント $a$ と $b$ が別の提携に含まれていることを確認する．同様に，分割 $\{\{c,d\},\{a\}\}$ で $c$ と $a$ が別の提携に含まれているため，$G$ で $c$ と $a$ が別の提携に含まれていることを確認する．ここで，それぞれ別の提携に含まれていることが確認できるため，*2* つの分割は統合可能であると判断される．

## 5. 整数計画問題による定式化

本章では，分割決定木による提携構造形成問題を整数計画問題によって定式化する．定式化において，最適な提携構造に対応する経路 $\pi$ は，各多分木に高々1 つ存在するという性質を利用する．具体的には，最適な提携構造に対応する経路 $\pi$ は同じ多分木に 1 つまで存在する制約条件を与える．また，多分木間の加法性，すなわち各多分木から 1 つまで経路 $\pi$ を選択し，それらが表す分割を 1 つの提携構造に統合することで最適な提携構造が構成される性質を利用する．ただし，選択した複数の経路 $\pi$ が表す分割は，必ずしも 1 つの提携構造に統合可能ではない．そこで，表す分割が 1 つの提携構造に統合不可能な経路 $\pi$ の集合 $D$ を作成し，集合 $D$ に含まれる経路 $\pi$ は同時に選択しない制約条件を与える．通常，そのような集合 $D$ は複数存在し，集合 $D$ を全て作成する必要がある．集合 $D$ の作成，すなわち複数の分割の統合可能性の判断には第 4 章で提案したアルゴリズムを用い，作成した集合 $D$ の集合を $\mathbb{D}$ で表す．これらの 2 つの制約のなかで，提携構造のもたらす効用の総和を最大にする経路 $\pi$ の集合を発見する．

**定義 5 (整数計画問題による定式化)** 最適な提携構造を構成する経路 $\pi$ の集合を発見する問題は，以下のような整数計画問題として表現できる．

$$
\begin{aligned}
\max \quad & \textstyle\sum_{T\in\mathbb{T}}\sum_{\pi\in\Pi(T)} v_\pi \cdot x_\pi \\
\text{s.t.} \quad & \forall T \in \mathbb{T}, \\
& \quad \textstyle\sum_{\pi\in\Pi(T)} x_\pi \le 1, \text{ --- } (i) \\
& \forall D \in \mathbb{D}, \\
& \quad \forall \pi \in D, \text{ where } D \text{ is the set of paths} \\
& \quad\quad \text{not satisfied by optimal partition simultaneously,} \\
& \quad\quad \textstyle\sum_{\pi\in D} x_\pi \le |D|-1, \text{ --- } (ii) \\
& \forall T \in \mathbb{T}, \\
& \quad \forall \pi \in \Pi(T), \\
& \quad\quad x_\pi \in \{0,1\}.
\end{aligned}
$$

経路 $\pi$ の表す提携構造のもたらす効用を $v_\pi$ とする．経路 $\pi$ が最適な提携構造を構成するとき，$x_\pi = 1$ となる．制約 (i) は，最適な提携構造を構成する経路 $\pi$ は，各多分木に高々1 つ存在することを保証している．制約 (ii) は，最適な提携構造は，表す分割が 1 つの提携構造に統合不可能な経路 $\pi$ の集合 $D$ によって構成されないことを保証している．すなわち，集合 $D$ に含まれる経路 $\pi$ は同時に選択しないことを示している．この表現では，変数の数は各多分木の経路 $\pi$ の数の総和に等しい．また，制約の数は $|\mathbb{T}| + |\mathbb{D}|$ である．

## 6. 計算機実験

本章では，提案したアルゴリズムの性能を計算機実験により評価する．比較対象として，第 5 章で定式化した整数計画問題による手法を用いる．本実験では，エージェント数を 200，多分木数を 10，各多分木の葉ノード数を $5, 10, \ldots, 30$ とし，各多分木に含まれるエージェント数を $[26, 35]$ で一様ランダム，含まれるエージェント数が $s$ の提携がもたらす効用を $[1, 2s]$ で一様ランダムとした．また，探索木での多分木の探索順序を変更することで，探索の深さが浅い内に枝刈りを行うことができる場合がある．したがって，多分木の探索順序がランダムと各分割の平均提携数 (降順) の 2 通りで実験を行った．各問題設定に対し 100 インスタンスを生成した．

葉ノードの総数 (横軸) と平均計算時間 (縦軸) の関係を図 3 に示す．整数計画問題による手法では，メモリ不足となり求解不可能であった．これは，分枝限定法による手法では，探索に必要な統合可能性の情報のみを必要とする一方で，整数計画問題による手法では，表す分割が統合不可能な経路 $\pi$ の集合についての情報を全て必要とするためであると考えられる．また，各分割の平均提携数 (降順) で多分木の探索順序を決定した提案アルゴリズムが効率的に探索できている．これは，各分割の提携数が大きくなることで，それらの分割の統合が難しくなり，枝刈りを効率的に行うことができるためであると考えられる．

## 7. 結論

本論文では，提携間に外部性が存在する場合の提携構造形成問題を扱った．初めに，分割関数の簡潔記述法である分割決定木を説明し，効率的に最適な提携構造を求める，分枝限定法を用いた提携構造形成アルゴリズムを提案した．次に，提案したアルゴリズムの比較対象として，分割決定木による提携構造形成問題を整数計画問題として定式化した．最後に，計算機実験により，提案したアルゴリズムの有用性を示した．

今後の方針としては，本論文では負の効用を含む分割決定木による提携構造形成問題を扱っていなかったため，そのような提携構造形成問題を効率的に解くアルゴリズムの提案が挙げられる．

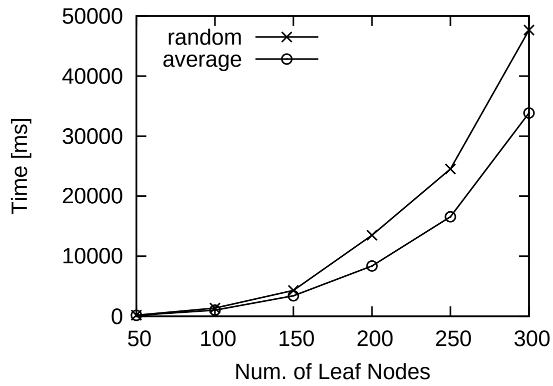


図 3: 各探索順序での平均計算時間

## 謝辞

本研究は JSPS 科研費 24220003，15K12101 の助成を受けました．ここに深く感謝いたします．

## 参考文献


[Conitzer 06] Conitzer, V. and Sandholm, T.: Complexity of constructing solutions in the core based on synergies among coalitions, *Artificial Intelligence*, Vol. 170, No. 6, pp. 607–619 (2006)

[Ieong 05] Ieong, S. and Shoham, Y.: Marginal contribution nets: a compact representation scheme for coalitional games, in *Proceedings of the Sixth ACM Conference on Electronic Commerce (ACM EC)*, pp. 193–202 (2005)

[Michalak 10] Michalak, T., Marciniak, D., Szamotulski, M., Rahwan, T., Wooldridge, M., McBurney, P., and R.Jennings, N.: A Logic-Based Representation for Coalitional Games with Externalities, in *Proceedings of the Ninth International joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS)*, pp. 125–132 (2010)

[Skibski 15] Skibski, O., Michalak, T. P., Sakurai, Y., Wooldridge, M., and Yokoo, M.: A Graphical Representation for Games in Partition Function Form, in *Proceedings of the Twenty-Ninth Conference on Artificial Intelligence (AAAI)*, pp. 1036–1042 (2015)

[Ueda 11] Ueda, S., Kitaki, M., Iwasaki, A., and Yokoo, M.: Concise characteristic function representations in coalitional games based on agent types, in *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence (IJCAI)*, pp. 393–399 (2011)

[一村 11] 一村 良, 長谷川 隆人, 上田 俊, 岩崎 敦, 横尾 真：MC-nets を用いた提携構造形成アルゴリズムの拡張: 負の利得と外部性の導入, 電子情報通信学会論文誌 D, Vol. 94, No. 11, pp. 1707–1715 (2011)