情報処理学会論文誌 Vol. 50 No. 12 3211–3221 (Dec. 2009)

開環境での協力ゲームにおける解の簡略記述法

大田直樹[†1] 岩崎敦[†2] 横尾真[†2]
ヴィンセント コニッツァー[†3] トゥオマス サンドホルム[†4]

必要に応じて協力関係/提携を構築することは，自律エージェントの持つべき重要な能力である．協力ゲーム理論では，協力によって得られた利得の適切な分配方法（協力ゲームの解概念）が提案されている．一方，既存の協力ゲームの解概念は，インターネット等の匿名の開環境での適用での問題点が指摘されており，新しい解概念である匿名操作不可能コアが提案されている．しかしながら，匿名操作不可能コアの記述量はエージェント/スキルの数に対して指数的となるという問題点が存在する．本論文ではこの課題に対応するため，入力である特性関数を，シナジー提携集合（*SCG*）を用いて簡略に記述することにより，出力である利得関数を簡略に記述する方法を提案する．匿名操作不可能コアはこの簡略表記利得関数により記述可能である．さらに，複数存在する場合があるという匿名操作不可能コアの問題点を解決した，新しい解概念である匿名操作不可能仁を提案する．匿名操作不可能仁も，匿名操作不可能コアと同じく簡略表記利得関数で記述可能である．

A Compact Representation Scheme for Coalitional Games in Open Anonymous Environments

Naoki Ohta,[†1] Atsushi Iwasaki,[†2] Makoto Yokoo,[†2]
Vincent Conitzer[†3] and Tuomas Sandholm[†1]

Recent research has revealed traditional solution concepts in coalitional game theory are vulnerable to various manipulations in open anonymous environments such as the Internet. To address this, a solution concept called the anonymity-proof core was developed. However, representing the outcome function of the anonymity-proof core requires space exponential in the number of agents/skills. We propose a compact representation of the outcome function, given that the characteristic function is represented using a compact language. This compact representation scheme can successfully express the anonymity-proof core. Furthermore, we develop the anonymity-proof nucleolus, that is compactly represented, always exists, and is uniquely determined.

1. はじめに

必要に応じて協力関係/提携を構築することは，自動化された利己的な主体（エージェント）が，他のエージェントと交渉を行う際に必要とされる重要な能力である．安定した提携を構築するためには，協力によって得られた利得をエージェント間でどのように分配するかを決定する必要がある．協力ゲーム理論では，解概念と呼ばれる，様々な望ましい性質を満足する利得の分配方法（シャプレイ値，コア，最小コア，仁等）が提案された．これらの解概念はマルチエージェントシステムの研究において利用されている[1),2),8),9),11)]．

さらに近年のインターネットの普及により，複数の企業，組織が動的，迅速に提携を構成することが可能/必要となったことから，協力ゲーム理論の適用分野は今後さらに拡大していくことが予想される．たとえば，複数の小さなベンチャー企業が提携し，仮想的な組織（バーチャルカンパニー）を動的に構成することにより，大規模かつ多様な顧客の要望に応えて利益を拡大することが可能である．このような場合に，複数の小さな企業間で，提携によって得られる利益をどのように分配するかが課題となる．また，グリッドコンピューティングを用いて大規模な計算を行う場合にも，資源の提供者に対して適切な報酬を分配することができれば，より多くの資源が利用可能になることが期待できる．

しかしながら，著者らは，既存の解概念をインターネットのような匿名の開環境で用いる際の問題点を指摘している[10),12)]．匿名の開環境では，単一のエージェントが複数のエージェントとして振る舞ったり，逆に複数のエージェントが共謀し，1 人のエージェントとして振る舞うことが可能である．さらに，エージェントは自分の持つ能力を過小申告することも可能である．文献 10), 12) では，利得の分配方法として既存の解概念を用いる場合，エージェントはこれらの操作を実行することにより，自らの利得を増加させることが可能であることを示している．

匿名の開環境においてエージェントが行う操作に対して頑健性を持つ解概念として，匿名

†1 九州大学大学院システム情報科学府
the Department of Intelligent Systems, Graduate School of ISEE, Kyushu University
†2 九州大学大学院システム情報科学研究院
the Department of Intelligent Systems, Faculty of ISEE, Kyushu University
†3 デューク大学
Department of Computer Science, Duke University
†4 カーネギーメロン大学
Computer Science Department, Carnegie Mellon University

操作不可能コアが提案されている[10),12)]．しかしながら，匿名操作不可能コアには 1 つの重要な問題点が存在する．それは，解となる利得関数の記述量がエージェント/スキルの数に対して指数的になることである．これは参加スキルの数に対して線形オーダでおさえられる，既存の解概念の解の記述量に比べてきわめて大きい．

本論文では，入力である特性関数をシナジー提携集合（synergy coalition group，SCG）[1)]を用いて簡略に表記することにより，出力である利得関数を簡略に記述する方法を提案する．シナジー提携集合とは，その提携が組まれることにより新たな価値が生み出されるような提携の集合である．本論文で提案する簡略表記利得関数は，シナジー提携集合から導かれる汎シナジー提携集合（generalized synergy coalition group，$GSCG$）を用いて，$GSCG$ の要素に対してのみ利得を記述することにより記述量の削減を実現する．この簡略表記利得関数を用いることで，匿名操作不可能コアに対応する利得関数を表現可能となる．さらに，匿名操作不可能コアは，解となる利得関数が複数存在する場合がある．そのため我々は，解が複数存在することなく，また簡略表記可能な新しい解概念である匿名操作不可能仁を提案する．

さて，協力ゲームに関する簡略表記法には，いくつかの研究事例がある．文献 4) は，ゲームのプレイヤがグラフ中のノードとして表現され，ノード間のリンクに重みが定義されている場合に，ある提携の利得が，その提携中のノードで構成されるサブグラフ中のリンクの重みの合計で与えられるゲームを対象としている．さらに，文献 3) では，ゲームのプレイヤが，フローネットワークのアークとして表現され，ある 2 点のノード s，t が定義されている場合に，ある提携の利得が，その提携中のアークで構成されるサブネットワーク中の s から t までの最大フローで与えられるゲームを対象としており，仁の探索に必要な計算量を考察している．また，文献 6) では，最小コスト被覆木ゲームにおいて，コアの存在のチェックに必要な計算量の考察を行っている．これらの研究は特定のゲームの表現に特化しているのに対して，我々の研究は，文献 5) と同様，任意の特性関数を表現可能な自然な簡略表記法を対象としている．しかしながら，文献 5) では，匿名の開環境での様々な操作を考慮していない．

本論文の構成は以下のとおりである．まず，2 章で，本論文が対象とする協力ゲームのモデルを示す．次に 3 章で，文献 10)，12) で導入された解概念である匿名操作不可能コアについて説明する．さらに，4 章で，本論文で提案する利得関数の簡略表記法を示し，この簡略表記利得関数を用いて，任意の匿名操作不可能コアに対応する利得関数を表現可能であることを示す．また，5 章で，簡略表記ができる新しい解概念である匿名操作不可能仁を提案する．最後に，6 章で，簡略表記利得関数を用いることによる記述の削減量を実験により評価し，通常の仁と匿名操作不可能仁，匿名操作不可能コアと匿名操作不可能仁の比較等に関して議論する．

2. モ デ ル

本文に入る前に本論文で頻出する数学記号について簡単に説明する．$s \in S$ とは s が集合 S の要素であることを示す．$S' \subset S$ や $S' \subseteq S$ は集合 S' が集合 S の部分集合であることを示す．ただし，本論文では一般には $S' \subseteq S$ を用い，S' と S が等しいと問題が生じる場合にのみ，$S' \subset S$ を用いる．つまり実際には S' と S が一致することがないとしても，定義や証明をする際に問題がない場合，$S' \subseteq S$ で表現する．

既存の協力ゲームの解概念は，特性関数を用いて利得の配分を決定している．特性関数 w は，任意の提携（ゲームに参加しうるエージェントの集合 N の部分集合）X を引数としており，$w(X)$ は X が協力しあうことにより得る利得の量を示す．

しかしながら，匿名の開環境でエージェントが実行可能な操作を定式化するためには，エージェントの集合に関して定義された通常の特性関数が与える情報のみでは不十分である．そこで文献 10)，12) では，エージェントの持つ能力に関するより詳細な記述として，エージェントが所有するスキルという概念を導入している．さらに，匿名の開環境でエージェントが行う操作を明確に定義するために，特性関数はスキルの集合に対して定義している．

定義 1（スキルとエージェント） それぞれのエージェントが持つ，分割不可能な技能をスキルと呼ぶ．各エージェントは 1 つ，もしくは複数のスキルを持つ．また各スキルはそれぞれ固有の名称を持ち，同じ機能/能力を持つスキルも区別することが可能であるとする．

定義 2（スキルの特性関数） スキルの特性関数 $v : 2^T \to \Re$（T はスキルの全体集合）は，任意のスキルの集合 S を引数とする関数である．$v(S)$ は，S を持つエージェントが協力した場合に得られる利得を示す．

スキルの特性関数 v は，エージェントの特性関数 w よりも詳細な情報を含んでいる．たとえばエージェント i が持つスキルの集合を S_i とすると，任意の提携 X について，$S_X = \bigcup_{i \in X} S_i$ とおいたとき，$w(X) = v(S_X)$ が成立する．このため，v から w を構成可能であるが，逆は不可能である．

本論文では，特性関数は優加法性を満足することを仮定する．

定義 3（優加法性） 任意のスキル集合 S_1，S_2 に関して，$v(S_1 \cup S_2) \geq v(S_1) + v(S_2)$ を満足する特性関数 v は優加法性を満たす．

すなわち優加法性は，2 つのスキルの集合を合わせた場合の利得は，少なくとも個々のスキルの集合の利得の和と同じであることを意味している．

ここからは，匿名の開環境でエージェントが行う操作について述べる．文献 10)，12) では，匿名の開環境におけるエージェントが実行可能な不正行為として，以下の 3 つを示している．

架空名義：スキルの集合 S_i を所持するエージェント i は，複数の名義を用いて複数のエージェントのように振る舞うことができる．このとき，i が振る舞う複数のエージェントの持つスキル（S_i^1，$\cdots$，S_i^q，$\cdots$）は以下の条件を満たす．

$$\forall q,\ \forall r \neq q,\ S_i^q \cap S_i^r = \emptyset,$$
$$\bigcup_q S_i^q \subseteq S_i.$$

共謀：任意のエージェントの集団 X は，1 人のエージェントとして振る舞うことができる．このとき，この 1 人のエージェントの持つスキルの集合は，共謀するエージェントが持つスキルの和集合の部分集合となる．

スキルの隠蔽：スキルの集合 S_i を所持するエージェント i は，一部のスキルを隠蔽し，S_i' しか持たないように振る舞うことができる．

以下のようにエージェントへの利得の配分を決定することにより，架空名義および共謀の影響を排除することが可能である[10),12)]．

- メカニズムデザイナと呼ばれる特別なエージェントが存在することを仮定する．このエージェントはこの協力ゲームに参加しうるスキルの集合 T と T の任意の部分集合を引数とする特性関数 v を知っている．
- 任意のエージェント i はゲームに参加する際，所持するスキルの集合をメカニズムデザイナに申告する．
- メカニズムデザイナはそれぞれのスキルに与える利得の配分を決定する．その際，どのエージェントがどのスキルを持っているかは考慮しない．
- 各エージェントに与える配分を決定する．この配分はエージェントが持つスキルの配分の総和とする．

この方法を用いれば，架空名義の利用や共謀を行ってもエージェントに与えられる利得は変化しないため，架空名義の利用や共謀は無意味となる．よって，本論文では以下，スキルの隠蔽の誘因を与えない解概念について議論を進める[10),12)]．

3. 匿名操作不可能コア

本章では，文献 10)，12) で導入された解概念である匿名操作不可能コアについて説明する．匿名操作不可能コアは前章で導入した 3 種類の操作に対して頑健性を持つ．

前章で述べたようにメカニズムデザイナは，事前に参加しうるスキルの集合 T とその任意の部分集合に対して利得の配分を決定する必要がある．本論文では，利得の配分方法は，以下に定義される利得関数 π により決定されることを仮定する．

定義 4（利得関数） 利得関数 $\pi(s, S)$（$S \subseteq T$，$s \in S$）とは，ゲームに参加したスキルの集合が S であった場合，スキル s が受け取る利得を示す．

この利得関数の定義は，文献 10)，12) での定義よりも単純化されているが，文献 10)，12) で示しているように，匿名操作不可能な利得関数を対象とする際には，この単純な利得関数を用いても一般性は失われない．

定義 5（匿名操作不可能な利得関数） 匿名操作不可能な利得関数 π は，以下の条件を満たす利得関数である．

- π はパレート効率性を満足する．すなわち，$\forall S, S \subseteq T$，$\sum_{s \in S} \pi(s, S) = v(S)$ が成立する．
- π はスキルの隠蔽の誘因を与えない．すなわち，$\forall S, \forall S', \forall S''$, where $S' \subseteq S \subseteq T$，$S'' \subseteq T$，$S'' \cap S = \emptyset$，$\sum_{s \in S'} \pi(s, S' \cup S'') \leq \sum_{s \in S} \pi(s, S \cup S'')$ が成立する．

定義 5 の条件のうち，下の式はスキルの集合 S を所持するエージェントが，任意のスキルの集合 S'' とともにゲームに参加する場合，自分の所持スキルを正直に S と申告した方が，S' と過少申告するよりも大きい利得を得ることを示す．

匿名操作不可能な利得関数のうち，任意のスキルの集合が独自に提携を組んで独立する誘因を生じさせない利得関数の集合を匿名操作不可能コアという．

定義 6（匿名操作不可能コア） 匿名操作不可能コアとは，以下の条件を満たす匿名操作不可能な利得関数 π の集合である．

- $\forall S, \forall S' \subseteq S$，$\sum_{s \in S'} \pi(s, S) \geq v(S')$

この条件の意味は以下のとおりである．任意のスキルの集合 S' に関して，$\sum_{s \in S'} \pi(s, S) < v(S')$ が成立するならば，これらのスキルの集合を持つエージェントは，ゲームから独立して独自に提携を組んだ方が得られる利得が大きくなる．このような提携をブロッキング提携と呼ぶ．匿名操作不可能コアは，ブロッキング提携を持たない利得関数の集合である．

以下に例を示す．

例 1　参加し得るスキルの集合が $T = \{a, b, c, d, e\}$ であり，スキルの特性関数 v が以下のように定義されているとする．

- $v(\{a, b, c, d, e\}) = 2$,
- $v(\{a, b, d, e\}) = v(\{b, c, d, e\}) = 2$,
- $v(\{a, b, c, d\}) = v(\{a, b, c, e\}) = v(\{a, c, d, e\}) = 1$,
- $v(\{a, b, c\}) = v(\{a, b, d\}) = v(\{a, b, e\}) = v(\{a, d, e\}) = v(\{b, c, d\}) = v(\{b, c, e\}) = v(\{b, d, e\}) = v(\{c, d, e\}) = 1$,
- $v(\{a, b\}) = v(\{b, c\}) = v(\{d, e\}) = 1$,
- その他の部分集合 S に対して，$v(S) = 0$.

このゲームの匿名操作不可能コア π は以下の条件を満たす利得関数の集合である．

- $\pi(b, \{a, b, c, d, e\}) = \pi(b, \{b, c, d, e\}) = \pi(b, \{a, b, d, e\}) = \pi(b, \{a, b, c, e\}) = \pi(b, \{a, b, c, d\}) = \pi(b, \{a, b, c\}) = \pi(b, \{a, b, d\}) = \pi(b, \{a, b, e\}) = \pi(b, \{b, c, d\}) = \pi(b, \{b, c, e\}) = \pi(b, \{a, b\}) = \pi(b, \{b, c\}) = 1$
- $\pi(d, \{a, b, c, d, e\}) = \pi(d, \{a, b, d, e\}) = \pi(d, \{a, c, d, e\}) = \pi(d, \{b, c, d, e\}) = \pi(d, \{a, d, e\}) = \pi(d, \{b, d, e\}) = \pi(d, \{c, d, e\}) = \pi(d, \{d, e\}) = p$
- $\pi(e, \{a, b, c, d, e\}) = \pi(e, \{a, b, d, e\}) = \pi(e, \{a, c, d, e\}) = \pi(e, \{b, c, d, e\}) = \pi(e, \{a, d, e\}) = \pi(e, \{b, d, e\}) = \pi(e, \{c, d, e\}) = \pi(e, \{d, e\}) = q$
- 上に含まれないスキル s とスキルの集合 $S \subseteq \{a, b, c, d, e\}$ の組 (s, S) について，$\pi(s, S) = 0$.

例 1 に示すとおり，匿名操作不可能コアとなる利得関数を定義する場合，利得関数の第 2 引数は，任意のスキルの集合 $S \subseteq T$ となっている．たとえば，実際に参加するスキルが $\{a, b, c, d\}$ とする．このとき，匿名操作不可能コアは，あるエージェントが何らかのスキルを隠蔽しても利得が増加しないことを保障する必要がある．そこで我々は，匿名操作不可能な解概念において，すべての起こりうる参加スキルの組合せについて利得関数をあらかじめ定義する．

しかしながら伝統的な解概念を用いる場合，配分の表記量はゲームに参加しうるスキルの総数 $|S| = n$ に対し，n となる．それに対し，匿名操作不可能コアとなる利得関数については，ある 1 つの第 1 引数に対する第 2 引数の数は第 1 引数のスキルを含む提携の総数 2^{n-1} 個になり，また第 1 引数の数は n 個なので，表記量は $n * 2^{n-1}$ となる．たとえば例 1 の利得関数をすべて書き出す場合，表記量は 80 にのぼる．そのため匿名操作不可能コアには利得関数の表記量がスキルの数に対して指数的になるという問題点がある（本論文の例 1 では返り値が 0 になるところをまとめることで表記量を 29 に抑えているがまだ不十分である）．

4. 利得関数の簡略表記

本章では，前章で示した利得関数の表記量の問題を解決するために，利得関数の簡略表記法を提案する．さらにこの簡略表記法を用いて任意の匿名操作不可能コアに対応する利得関数を簡略に表現できることを示す．本論文では，特性関数は文献 1) で提案されたシナジー提携集合（SCG）を用いて簡略に記述されていることを仮定する．シナジー提携集合は，以下のように定義される．

定義 7（シナジー提携集合）　シナジー提携集合 SCG とは，提携が組まれることで何らかの価値を生み出すスキルの提携の集合であり，以下の式を満たすスキル集合 S の集合である．

- $\bigcup_l S_l = S$ および $\forall q, \forall r \neq q$，$S_q \cap S_r = \emptyset$ を満たす任意のスキルの提携の集合 $SS = \{S_1, \ldots, S_q, \ldots\}$ について $v(S) > \sum_{S_q \in SS} v(S_q)$ が成立する．

たとえば例 1 の協力ゲームにおける SCG は $\{\{a, b\}, \{b, c\}, \{d, e\}\}$ となる．

任意のスキル集合 S に対する特性関数の値 $v(S)$ は，以下のように，そのゲームの SCG の要素に対する特性関数の値より導くことができる．

- $v(S) = \max\{\sum_{1 \leq r \leq j} v(S_r)\}$,
  （ただし $\bigcup_{1 \leq r \leq j} S_r \subseteq S$，かつすべての S_r は共通部分を持たない SCG の要素）．

すなわち，$v(S)$ は，S について互いに素かつ和集合が S の部分集合となる SCG の組合せに関して，特性関数の値の和の最大値となる．

SCG を用いることで特性関数は，簡略に表記できるが，匿名操作不可能コアになるような利得関数を簡略に表記することはできない．たとえば，SCG が $\{a, b\}$ だけの場合，参加スキルが $\{a, b, c\}$ のときの配分を $\{a, b\}$ のときの配分を参考に決定できる．しかし，$\{a, b\}$, $\{b, c\}$ が SCG に含まれ $\{a, b, c\}$ が SCG に含まれない場合，参加スキルが $\{a, b, c\}$ のときの利得の配分を，$\{a, b\}$ が参加したときの配分を参考にすべきか，$\{b, c\}$ のときを参考にすべきか決定できない．事実，SCG だけで匿名操作不可能コアになるような利得関数を簡略に表記できないような例が存在する．この問題を解決するため，本論文では，この SCG を用いて，以下のように汎シナジー提携集合（$GSCG$）と呼ばれる概念を定義する．

定義 8（汎シナジー提携集合）　汎シナジー提携集合 $GSCG$ は以下の条件を満たすスキルの提携の集合である．

- $\forall S$, if $S \in SCG$，then $S \in GSCG$,

- $\forall S_1 \in GSCG$，$\forall S_2 \in GSCG$，if $S_1 \cap S_2 \neq \emptyset$，then $S_1 \cup S_2 \in GSCG$.

具体的には，$GSCG$ は，SCG の要素，および SCG の，互いに素でない複数の要素の和集合からなる．たとえば，例 1 の協力ゲームにおける $GSCG$ は SCG に $\{a, b, c\}$ を加えたものとなる．

次に，$GSCG$ を用いた利得関数の簡略表記法について述べる．まず，スキル集合の $GSCG$ への射影を定義する．

定義 9（スキル集合の $GSCG$ への射影） あるスキル集合 S の $GSCG$ への射影 P_S を以下のように定義する：

$$P_S = \{C | C \in GSCG, C \subseteq S, \text{ and } \forall C', \text{ where } C \subset C' \subseteq S, C' \notin GSCG\}$$

たとえば例 1 のゲームにおいて，スキル集合 $\{a, b, c, d, e\}$ の $GSCG$ への射影は，$\{\{a, b, c\}, \{d, e\}\}$ となる．

次に $GSCG$ への射影を用いて，利得関数を簡略に表記する方法を説明する．まず，以下のように簡略表記利得関数 π_c を定義する．

定義 10（簡略表記利得関数） 簡略表記利得関数 π_c は，第 2 引数に $S \in GSCG$ を満たす任意のスキル集合 S，第 1 引数に $s \in S$ を満たす任意のスキル s をとる関数である．$\pi_c(s, S)$ は実際にゲームに参加したスキルの集合が S であるとき，スキル s が受け取る利得である．また簡略表記利得関数 π_c は，以下の条件（パレート効率性）を満たすことを仮定する．

- $\forall S, S \in GSCG$，$\sum_{s \in S} \pi_c(s, S) = v(S)$

簡略表記利得関数の基本的なアイデアは，任意のスキル集合に対して利得関数を定義するのではなく，$GSCG$ の要素に対してのみ利得関数を定義し，その他のスキル集合に対する利得は，$GSCG$ の要素に対する利得から算出するというものである．たとえば，例 1 のゲームについて，簡略表記利得関数 $\pi_c(s, S)$ の第 2 引数 S のとりうる値は，$GSCG$ の要素である $\{a, b\}$，$\{a, c\}$，$\{d, e\}$，$\{a, b, c\}$ のみである．

以下のように，簡略表記利得関数から通常の利得関数を一意に算出できる．

定義 11（利得関数の算出方法） 任意の簡略表記利得関数 π_c より算出される利得関数 π の値は以下で与えられる．

$$\pi(s, S) = \begin{cases} \pi_c(s, P) & \text{if there exists } P \text{ where } P \ni s \text{ and } P \in P_S \\ 0 & \text{otherwise.} \end{cases}$$

ある簡略表記利得関数 π_c より，通常の利得関数が 1 つ得られる．しかしながら，通常の利得関数には，簡略表記利得関数で表現できないものも存在する．たとえば例 1 において，$\pi(d, \{a, b, c, d\}) \neq 0$ が成立する利得関数はどのような簡略表記利得関数によっても表現されない．

定義 12（簡略に表記できる利得関数） 簡略に表記できる利得関数とは，ある簡略表記利得関数から算出される利得関数である．

以下，匿名操作不可能コアに含まれる利得関数を表現する簡略表記利得関数を定義し，匿名操作不可能コアが非空の場合，そのような簡略表記利得関数が必ず存在することを示す．

まず匿名操作不可能な利得関数を表現する簡略表記利得関数を定義する．

定義 13（隠蔽操作不可能な簡略表記利得関数） 隠蔽操作不可能な簡略表記利得関数 π_c とは以下の条件を満たす簡略表記利得関数である．

$B \in GSCG$ を満たす任意の B，$S' \subseteq S \subseteq B$ を満たす任意の S，S' および，$B \setminus (S \setminus S')$ の $GSCG$ への射影，$P_{B \setminus (S \setminus S')}$ について

$$\sum_{P \in P_{B \setminus (S \setminus S')}} \sum_{s \in (S' \cap P)} \pi_c(s, P) \leq \sum_{s \in (B \cap S)} \pi_c(s, B).$$

この定義式は，参加しているスキルの集合が任意の GSCG の要素 B であるとき，任意のエージェントが一部のスキルを隠蔽しても，獲得利得が上昇しないという条件を示す．

定理 1　隠蔽操作不可能な簡略表記利得関数 π_c によって表現される利得関数 π は匿名操作不可能である．

証明　隠蔽操作不可能性を満たす任意の簡略表記利得関数 π_c によって表現される利得関数を π とする．以下，互いに素な任意のスキルの集合 S_1，S_2，S_3 について，

$$\sum_{s \in S_1} \pi(s, S_1 \cup S_3) \leq \sum_{s \in (S_1 \cup S_2)} \pi(s, S_1 \cup S_2 \cup S_3)$$

が成立することを示す．すなわち，あるエージェントがスキルの集合 S_1，S_2 を所有しており，他のエージェントがスキルの集合 S_3 を所有している場合に，エージェントがスキルの

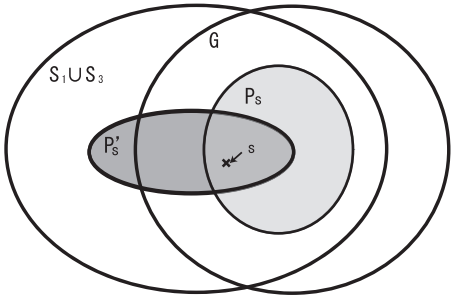

図 1　定理 1 の証明におけるスキルの集合 P，P'
Fig. 1　Groups of skills P, P' in the Proof of Theorem 1.

集合 S_2 を隠蔽しない場合の利得（上式の右辺）が，S_2 を隠蔽した場合の利得（上式の左辺）以上となることを示すことで，π が匿名操作不可能であることを示す．

以下，証明を簡単にするため $S_1 \cup S_2 \cup S_3$ の $GSCG$ への射影はただ 1 つのスキルの集合 G によって構成されると仮定する．$S_1 \cup S_2 \cup S_3$ の $GSCG$ への射影が複数のスキルの集合 $G_1, \ldots G_i \ldots G_j$ によって構成される場合については以下の証明中に補足する．

s を $s \in (S_1 \cap G)$ を満たす任意のスキルとする（$S_1 \cup S_2 \cup S_3$ の $GSCG$ への射影が複数のスキルの集合 $G_1, \ldots G_i \ldots G_j$ によって構成される場合，s は $s \in S_1 \cap \bigcup_{1 \le i \le j} G_i$ を満たす任意のスキルとする．また以下の議論で G の代わりに $G_i \ni s$ を満たすスキルの集合 G_i を用いる．任意の s について前述した条件を満たす G_i は一意に決定される）．$G \setminus (G \cap S_2)$ の $GSCG$ への射影のなかで s を含むものがたかだか 1 つ存在する．以下そのようなスキルの集合を P_s とする．ただし，そのようなスキルが存在しない場合 $P_s = \emptyset$ とする．同様に，$S_1 \cup S_3$ の $GSCG$ への射影のなかで s を含むものを P'_s とおく．ただし，そのようなスキルが存在しない場合 $P'_s = \emptyset$ とする．これらの集合の包含関係を**図 1** に示す．

まず，$P_s = P'_s$ が成立することを示す．最初に，$P'_s \subseteq G$ が成立することを示す．P'_s が G の部分集合でないと仮定する．このとき，P'_s は空集合でないため，P'_s と G は s という共通要素を持つ．$GSCG$ の定義より $P'_s \cup G$ は GSCG の要素であり，また $G \subseteq P'_s \subseteq S_1 \cup S_2 \cup S_3$ が成立する．これらの事実と $GSCG$ への射影の定義より，G は $S_1 \cup S_2 \cup S_3$ の $GSCG$ への射影には含まれず，仮定に矛盾する．よって $G = G \cup P'_s$ が成立する．

次に，$P_s \neq P'_s$ と仮定する．このとき P_s，P'_s のいずれかは $P_s \cup P'_s$ の真部分集合となる．また G の定義より $G \setminus (G \cap S_2) \subseteq S_1 \cup S_3$ が成立するため，$P_s \subseteq S_1 \cup S_3$ が成立する．さらに G の定義と，$P'_s \subseteq G$ および，$P'_s \subseteq S_1 \cup S_3$ より，$P'_s \subseteq G \setminus (G \cap S_2)$ が成立する．以上の事実より，$P_s \cup P'_s$ は $GSCG$ に含まれており，かつ $S_1 \cup S_3$ および $G \setminus (G \cap S_2)$ の部分集合である．$GSCG$ の射影の定義より，P_s が $G \setminus (G \cap S_2)$ の $GSCG$ への写像に含まれるという条件と，P'_s が $S_1 \cup S_3$ の $GSCG$ への写像に含まれるという条件が同時に満たされることはなく，仮定に矛盾する．以上より $P_s = P'_s$ が成立する．上記の議論は，P_s，P'_s が空集合の場合でも成立する．

以下，$P_s = P'_s$ が成立することを前提として，π が匿名操作不可能な特性関数であることを示す．隠蔽操作不可能な簡略表記利得関数の定義より以下の式が成立する（$S_1 \cup S_2 \cup S_3$ の $GSCG$ への射影が複数のスキルの集合 $G_1, \ldots G_i \ldots G_j$ によって構成される場合，式内で複数用いられている “$s \in (S_1 \cap G)$” の代わりに “$s \in (S_1 \cap \bigcup_{1 \le i \le j} G_i)$” を用いる．また式 “$\sum_{s \in ((S_1 \cup S_2) \cap G)} \pi_c(s, G)$” の代わりに “$\sum_{1 \le i \le j} \sum_{s \in ((S_1 \cup S_2) \cap G_i)} \pi_c(s, G_i)$” を用いる）：

$$\sum_{s \in (S_1 \cup S_2)} \pi(s, S_1 \cup S_2 \cup S_3) = \sum_{s \in ((S_1 \cup S_2) \cap G)} \pi_c(s, G) \ge \sum_{s \in (S_1 \cap G)} \pi_c(s, P_s)$$

$$\sum_{s \in S_1} \pi(s, S_1 \cup S_3) = \sum_{s \in (S_1 \cap G)} \pi_c(s, P'_s) = \sum_{s \in (S_1 \cap G)} \pi_c(s, P_s)$$

これらの式より，以下の式が得られる．

$$\sum_{s \in S_1} \pi(s, S_1 \cup S_3) \le \sum_{s \in (S_1 \cup S_2)} \pi(s, S_1 \cup S_2 \cup S_3)$$

よって隠蔽操作不可能な簡略表記利得関数 π_c により表現される利得関数 π は，匿名操作不可能な利得関数である．□

次に SCG に対するノンブロッキング条件を定義し，この条件と隠蔽操作不可能性を満たす簡略表記利得関数から導かれる利得関数が，匿名操作不可能コアの要素となることを示す．

定義 14（SCG に対するノンブロッキング条件）　ある簡略表記利得関数 π_c が SCG に対するノンブロッキング条件を満たすとは，$B \in GSCG$ を満たす任意のスキルの集合 B および $S \in SCG$，$S \subseteq B$ を満たす任意のスキルの集合 S に対して，$\sum_{s \in S} \pi_c(s, B) \ge v(S)$ を満たすことである．

定理 2　SCG に対するノンブロッキング条件を満たす隠蔽操作不可能な簡略表記利得関

数 π_c によって表現される利得関数 π は，匿名操作不可能コアに含まれる．

証明　SCG に対するノンブロッキング条件を満足する隠蔽操作不可能な簡略表記利得関数 π_c によって表現され，かつ匿名操作不可能コアに含まれない利得関数 π が存在すると仮定して矛盾を導く．仮定より，π にはブロッキング提携が存在する．すなわち，π は $\sum_{s\in S'}\pi(s,S) < v(S')$ および $S' \subseteq S$ を満足するスキルの集合の組 S，S' を持つ．ここで，以下のようにスキル集合の集合 $\{C_1,\ldots,C_q,\ldots\}$ を選択する．すなわち，すべての C_q は互いに素な SCG の要素であり，$v(S)=\sum_q v(C_q)$ が成立するものとする．このようなスキルの集合の組は定義 7 より必ず存在する．このとき，$\sum_q v(C_q)=v(S)>\sum_{s\in S'}\pi(s,S)\geq\sum_q\sum_{s\in C_q}\pi(s,S)$ が成立するため，$v(C_l)>\sum_{s\in C_l}\pi(s,S)$ が成立する C_l が最低 1 つ存在する．ここで $v(C_l)>\sum_{s\in C_l}\pi(s,S)$ が成立する C_l について，定義 8，9 より，S の $GSCG$ の射影 P_S の要素 P には，$C_q\subseteq P$ が成立するものがただ 1 つ存在するので，$\forall s\in C_l$，$\pi_c(s,P)=\pi(s,S)$ が成立する．C_l は SCG の要素なので，π_c は SCG に対するノンブロッキング条件を満たしておらず，仮定に矛盾する．以上より，π は匿名操作不可能コアに含まれる．□

例 2　例 1 のゲームの匿名操作不可能コアに含まれる利得関数 π は，以下に示す，SCG に対するノンブロッキング条件を満たす隠蔽操作不可能な簡略表記利得関数 π_c により表現可能である．

p，q を $p\geq 0$，$q\geq 0$，$p+q=1$ を満たす任意の数の組とする．

- $\pi_c(a,\{a,b,c\})=0$，$\pi_c(b,\{a,b,c\})=1$，
  $\pi_c(c,\{a,b,c\})=0$，
- $\pi_c(a,\{a,b\})=0$，$\pi_c(b,\{a,b\})=1$，
- $\pi_c(b,\{b,c\})=1$，$\pi_c(c,\{b,c\})=0$，
- $\pi_c(d,\{d,e\})=p$，$\pi_c(e,\{d,e\})=q$.

この π_c から匿名操作不可能コアに含まれる利得関数 π が導かれる．たとえば $\pi(d,\{a,b,c,d,e\})$ を求める場合，$\{a,b,c,d,e\}$ の $GSCG$ への射影は $\{\{a,b,c\},\{d,e\}\}$ となる．よって $\pi(d,\{a,b,c,d,e\})=\pi_c(d,\{d,e\})=p$ となる．またスキル b，d を持つエージェントが受け取る利得は，$\pi(b,\{a,b,c,d,e\})$ と $\pi(d,\{a,b,c,d,e\})$ の和であるので，$\pi_c(b,\{a,b,c\})+\pi_c(d,\{d,e\})=1+p$ となる．

上の例では，利得関数を簡略に表記することで，引数の組の数は 80 から 9 に減少している．この例で示されているように，簡略表記により記述量を大幅に削減することが可能である．

最後に協力ゲームの匿名操作不可能コアが非空である場合，必ず隠蔽操作不可能かつ SCG に対するノンブロッキング条件を満たす簡略表記利得関数が存在することを証明する．

定理 3　匿名操作不可能コアとなる利得関数 π が存在する場合，隠蔽操作不可能かつ SCG に対するノンブロッキング条件を満たす簡略表記利得関数 π_{ap} が存在する．

証明　π が匿名操作不可能コアに含まれる利得関数であると仮定する．ここで，$\forall C\in GSCG$，$\forall s\in C$，$\pi_c(s,C)=\pi(s,C)$ を満たす簡略表記利得関数 π_c を考える．このとき π が匿名操作不可能コアに含まれる利得関数であることから，定義 5 より，$\forall C\in GSCG$，$\forall S\subseteq C$，$\forall S'\subseteq S$，$\sum_{P\in P_{C\setminus(S\setminus S')}}\sum_{s\in(S'\cap P)}\pi_c(s,P)=\sum_{(C\setminus(S\setminus S'))\cap S}\pi(s,C\setminus(S\setminus S'))\leq\sum_{s\in(C\cap S)}\pi(s,C)=\sum_{s\in(C\cap S)}\pi_c(s,C)$ が成立する．よって，π_c は隠蔽操作不可能な簡略表記利得関数である．また定義 6 より $\forall C\in GSCG$，$\forall S\in SCG$，$S\subseteq C$，$\sum_{s\in S}\pi_c(s,C)=\sum_{s\in S}\pi(s,C)\geq v(S)$ が成立する．よって π_c は SCG に対するノンブロッキング条件を満足する．定理 2 より，π_c によって表現される利得関数を π' とすると，π' は匿名操作不可能コアに含まれる．□

5. 匿名操作不可能仁

本章では，本論文で新しく提案する解概念である匿名操作不可能仁について述べる．この解概念は，特性関数によっては，解が複数存在する場合があるという，匿名操作不可能コアが持つ問題点を解決している．

匿名操作不可能仁は，伝統的な解概念である仁[7] に基づいている．仁は「不満」という概念を用いて定義される．具体的には，ある提携がある配分に対して持つ不満は，その提携が与える特性関数の利得と，その提携がその配分から受け取る利得の差として定義される．さらに，ある配分に関して，すべての提携の不満を求め，それを降順に並べたベクトルをその配分の不満ベクトルとする．ここで，辞書式順序で最も小さい不満ベクトルを持つ配分を仁と呼ぶ．

任意の d 次元を持つベクトル $\vec{h}$，$\vec{h'}$ について，$\vec{h}$ が辞書式順序で $\vec{h'}$ より小さいとは，以下の条件を満たす d 以下の自然数 j を持つことである．

- $\vec{h}_j<\vec{h'}_j$
- j 未満の任意の自然数 j' について，$\vec{h}_{j'}=\vec{h'}_{j'}$

ただし $\vec{h}_j$ とは $\vec{h}$ の第 j 要素をさす．

仁は以下の 3 つの特徴を持つ．

- どのような特性関数に対しても，仁は存在し，かつ一意に定まる．
- コアが非空である場合，仁はコアに必ず含まれる．

- 同じ能力を持つ複数のエージェントに対して，仁が与える利得は等しい（対称性）．

匿名操作不可能仁を定義するため，不満および不満ベクトルを以下のように再定義する．

定義 15（不満） $B \in GSCG$ を満たす任意のスキルの集合 B および，$S \subseteq B$ に関して，参加スキルが B のとき，S が任意の簡略表記利得関数 π_c に持つ不満 $f(\pi_c, S, B)$ は以下の式により定義される：

$$f(\pi_c, S, B) = v(S) - \sum_{s \in S} \pi_c(s, B)$$

定義 16（簡略不満ベクトル） 簡略不満ベクトルとは，$B \in GSCG$ を満たす任意のスキルの集合 B および，$S \subseteq B$ かつ S は SCG の要素もしくは要素数が 1 のスキル集合に関して，$f(\pi_c, S, B)$ を降順に並べたものである．

定義 17（匿名操作不可能仁） 匿名操作不可能仁とは，辞書式順序で最も小さい簡略不満ベクトルを持つ隠蔽操作不可能な簡略表記利得関数である．

例 3　例 1 の協力ゲームについて考える．参加スキルが $T = \{a, b, c, d, e\}$ である場合，匿名操作不可能仁となる簡略表記利得関数は例 2 で与えられた簡略表記利得関数のうち，$p = q = 0.5$ となるもの，すなわち以下のとおりである．

- $\pi_c(a, \{a, b, c\}) = 0$，$\pi_c(b, \{a, b, c\}) = 1$，
  $\pi_c(c, \{a, b, c\}) = 0$，
- $\pi_c(a, \{a, b\}) = 0$，$\pi_c(b, \{a, b\}) = 1$，
- $\pi_c(b, \{b, c\}) = 1$，$\pi_c(c, \{b, c\}) = 0$，
- $\pi_c(d, \{d, e\}) = 0.5$，$\pi_c(e, \{d, e\}) = 0.5$.

以下，匿名操作不可能仁の主な性質を示す．まず最初に，匿名操作不可能仁が複数存在することはないことを示す．

定理 4　匿名操作不可能仁となる簡略利得関数が複数存在することはない．

証明　相異なる匿名操作不可能仁となる簡略表記利得関数 π_c'，π_c'' が存在すると仮定し矛盾を導く．仮定より，π_c'，π_c'' の簡略不満ベクトルは同一である．以下のように，新しい簡略表記利得関数 π_c を作る．すなわち π_c は $\forall s \in S$，$S \in GSCG$，$\pi_c(s, S) = (\pi_c'(s, S) + \pi_c''(s, S))/2$ を満たす関数である．ここで，π_c'，π_c'' が隠蔽操作不可能な簡略表記利得関数であるため，π_c も隠蔽操作不可能な簡略表記利得関数であるのは明らかである．さらに，文献 7) の定理 2 と同様な議論により，π_c の簡略不満ベクトルが，π_c'，π_c'' の簡略不満ベクトルより辞書式順序で小さいことが導かれる．これは π_c'，π_c'' が匿名操作不可能仁であるという仮定に矛盾する．□

定理 5　匿名操作不可能コアが非空であるとき，匿名操作不可能仁は必ず存在し，かつ匿名操作不可能コアに含まれる．

証明　匿名操作不可能仁である簡略表記利得関数を π_c とする．定理 3 より，匿名操作不可能コアが非空である場合，SCG に対するノンブロッキング条件を満たす隠蔽操作不可能な簡略表記利得関数 π_c' が存在する．このとき，π_c' はノンブロッキング条件を満たすため，簡略不満ベクトルのすべての要素は 0 以下となる．ここで π_c の簡略不満ベクトルは，π_c' の簡略不満ベクトルよりも，辞書式順序で等しい（$\pi_c = \pi_c'$ のとき）か小さいため，π_c の簡略不満ベクトルのすべての要素は 0 以下となる．したがって，π_c はノンブロッキング条件を満たすため，定理 2 より匿名操作不可能コアに含まれる．□

定理 6　2 つのスキル s，s' が対称である場合，すなわち，$\forall S$, where $s \notin S$，$s' \notin S$，$v(S \cup \{s\}) = v(S \cup \{s'\})$ が成立する場合，匿名操作不可能仁となる簡略表記利得関数 π_c は s，s' に同じ利得を与える．

証明　匿名操作不可能仁となる簡略表記利得関数 π_c が対称関係にある 2 つのスキル s，s' について，異なる利得を与えると仮定し矛盾を導く．仮定により，以下の条件のいずれかが成立する．

$$\pi_c(s, S \cup \{s\}) \neq \pi_c(s', S \cup \{s'\}),$$
$$\pi_c(s, S \cup \{s\} \cup \{s'\}) \neq \pi_c(s', S \cup \{s\} \cup \{s'\})$$

次に以下の条件を満たす簡略表記利得関数 π_c' を定義する．

$$\pi_c'(s, S \cup \{s\}) = \pi_c(s', S \cup \{s'\}),$$
$$\pi_c'(s', S \cup \{s'\}) = \pi_c(s, S \cup \{s\}),$$
$$\pi_c'(s, S \cup \{s\} \cup \{s'\}) = \pi_c(s', S \cup \{s\} \cup \{s'\}),$$
$$\pi_c'(s', S \cup \{s\} \cup \{s'\}) = \pi_c(s, S \cup \{s\} \cup \{s'\}),$$
$$\text{and } \pi_c'(\cdot, \cdot) = \pi_c(\cdot, \cdot) \text{ otherwise.}$$

このとき，π_c' は，π_c と同じ簡略不満ベクトルを持つため，π_c' も匿名操作不可能仁となる．これは定理 4 と矛盾する．□

6. 実験/考察

6.1 記述量削減の効果

本節では，簡略表記利得関数を用いた場合の表記量の削減効果を実験より示す．本実験では以下のような条件で，簡略表記利得関数を用いた場合の表記量の評価を行う．

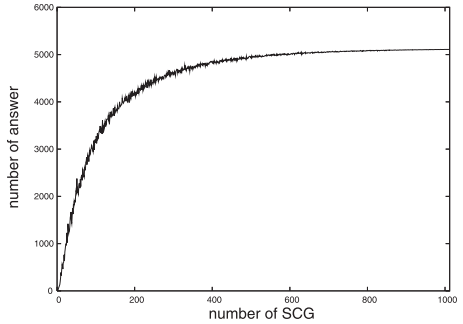

図 2　SCG の数と簡略表記利得関数の表記量の関係
Fig. 2　The relation between the number of SCG and the representation size of the compact outcome function.

- 参加しうるスキルの総数を 10 とする（この場合，提携の総数は 1,023，利得関数の表記量は 5,120 となる）．
- ゲーム内のすべての提携の中から j 個の提携をランダムに選ぶ．選ばれた提携の集合を SCG とする．

図 2 に，$1 \leq j \leq 1023$ を満たす，すべての整数 j すなわち起こりうるすべての SCG の数について計算した簡略表記利得関数の表記量を示す．図に示している表記量は 100 回の試行結果の平均である．簡略表記利得関数の表記量は，SCG の内容にのみ依存する．SCG となる提携の数が小さければ，簡略表記利得関数の表記量は，通常の利得関数の表記量よりはるかに小さいことが示されている．

また匿名操作不可能コアの算出には線形計画法を用いるが，この線形計画問題の変数の数は，利得関数の表記量と同じである．またこの線形計画問題の算出に必要な時間は，利得関数の表記量とほぼ比例している．よって，利得関数を簡略に表記することで，匿名操作不可能コアを迅速に解くことができる．

SCG となる提携の数が十分小さいというのは，一見成立しにくい条件のように見える．しかし SCG となる提携の数が多いという状況はあまり起こりえないと考えられる（スキルの大半がそれぞれ相互関係を持ち，かつ似たようなスキルは存在しないという条件が必要となるため）．そのため簡略表記利得関数は，十分に実用に耐えうるものと思われる．

6.2　仁と匿名操作不可能仁

前章で示したように，匿名操作不可能仁は従来の仁の持つ多くの性質を満足する．しかしながら，仁は全提携が持つ不満の中で，最大の不満を最小化という性質を持つが，匿名操作不可能仁はこの性質を満足しない．たとえば参加スキルが $\{a, b, c, d, e, f, g\}$ で，以下の SCG の要素に関する特性関数の値が $v(\{a, b, c\}) = v(\{a, b, d\}) = v(\{a, c, d\}) = v(\{b, c, d\}) = 1$，$v(\{e, f\}) = v(\{e, g\}) = v(\{f, g\}) = 1$ で与えられる 7 スキル協力ゲームを考える．ここで 7 つのスキルすべてが参加した場合を考える．このとき，匿名操作不可能仁は，スキル a，b，c，d にそれぞれ 1/4 ずつ，スキル e，f，g にそれぞれ 1/3 ずつの配分を与える．この場合，スキルの集合 $\{a, b, c, e, f\}$ は $2 - 3/4 - 2/3 = 7/12$ の不満を持つ．また，この不満は全提携が持つ不満の中で最大となっている．ここで e，f，g が受け取る利得のうち微少量 ϵ を a，b，c，d に渡すとする．つまり a，b，c，d が受け取る利得はそれぞれ $(1 + \epsilon)/4$，e，f，g が受け取る利得は $(1 - \epsilon)/3$ とする．このときスキルの集合 $\{a, b, c, e, f\}$ が持つ不満は，$7/12 - \epsilon/12$ に減少し，またこの不満は，全提携が持つ不満の中で最大となる．よって，匿名操作不可能仁は最大不満を最小化する解概念ではない．

一方，匿名操作不可能仁は SCG に含まれるスキル集合の提携に関しては，最大不満を最小化する．匿名操作不可能仁は，計算量/表記量の削減と，最大不満の最小化という 2 つの課題を，適切なレベルで両立している解概念と考えることができる．

6.3　匿名操作不可能コアと匿名操作不可能仁

匿名操作不可能仁は，匿名操作不可能コアが持ちえない望ましい性質を持っている．しかしながら，匿名操作不可能仁は匿名操作不可能コアに比べて，解を算出するために必要な計算量が大きいという問題を持つ．具体的には，匿名操作不可能仁を求めるためには，既存の仁の算出方法と同じように線形計画法を繰り返し適用して，簡略不満ベクトルを最小化する簡略表記利得関数を求める必要がある（仁の算出方法との相違点として利得関数が隠蔽操作不可能になるための制約条件を付ける必要がある）のに対し，匿名操作不可能コアを求めるためには，線形計画法を 1 回のみ適用すればよい．

7. おわりに

匿名の開環境においてエージェントが実行可能な様々な操作に対して頑健性を保証するために，匿名操作不可能な解概念が提案されている．しかしながら，従来の協力ゲームの解の記述量は，参加スキルの数に対して線形であるのに対して，匿名操作不可能な利得関数を表記するためには，スキルの数に対して指数的な記述量が必要とされる．

本論文では，この課題に対応するため，入力である特性関数を，シナジー提携集合（SCG）を用いて簡略に記述することにより，出力である利得関数を簡略に記述する方法を提案し

た．またこの簡略表記利得関数を用いて，任意の匿名操作不可能コアに対応する利得関数を表現することが可能であることを示した．

また実験により，入力である *SCG* のサイズが小さい場合は，簡略表記利得関数の導入により記述量が大幅に削減されることを示した．

さらに，本論文では，新しい解概念である匿名操作不可能仁を提案した．匿名操作不可能仁は簡略表記ができる，解が複数存在しえない，対称性を持つ，匿名操作不可能コアが非空である限りそれに含まれる，といった性質を持つ．

参 考 文 献

1) Conitzer, V. and Sandholm, T.: Complexity of determining nonemptiness of the core, *Proc. 18th International Joint Conference on Artificial Intelligence* (*IJCAI*), pp.613–618 (2003).
2) Conitzer, V. and Sandholm, T.: Computing Shapley values, manipulating value division schemes and checking core membership in multi-issue domain, *Proc. 19th National Conference on Artificial Intelligence* (*AAAI*), pp.219–225 (2004).
3) Deng, X., Fang, Q. and Sun, X.: Finding nucleolus of flow game, *Proc. 17th Annual ACM-SIAM Symposium on Discrete Algorithm*, pp.124–131 (2006).
4) Deng, X. and Papadimitriou, C.: On the complexity of cooperative solution concept, *Proc. 18th International Joint Conference on Artificial Intelligence* (*IJCAI*), pp.257–266 (1994).
5) Elkind, E., Goldberg, L.A., Goldberg, P.W. and Wooldridge, M.: Tractable and expressive class of Marginal Contribution Nets and its application, *Proc. 7th International Joint Conference on Autonomous Agents and Multiagent Systems* (*AAMAS*), pp.1007–1014 (2008).
6) Faigle, U., Fekete, S.P., Hochstättler, W. and Kern, W.: On the Complexity of Testing Membership in the Core of Min-cost Spanning Tree Games, *International Journal of Game Theory*, Vol.26, No.3, pp.361–366 (1997).
7) Schmeidler, D.: The nucleolus of a characteristic function game, *Journal of Applied Mathematics*, Vol.17, pp.1163–1170 (1969).
8) Shehory, O. and Kraus, S.: Methods for task allocation via agent coalition formation, *Artificial Intelligence*, Vol.101, No.1-2, pp.165–200 (1998).
9) Stuart, Jr., H.W.: The supplier-firm-buyer game and its m-sided generalization, *Mathematical Social Sciences*, Vol.34, No.1, pp.21–27 (1997).
10) Yokoo, M., Conitzer, V., Sandholm, T., Ohta, N. and Iwasaki, A.: Coalitional games in open anonymous environments, *Proc. 20th National Conference on Artificial Intelligence* (*AAAI*), pp.509–515 (2005).
11) Zlotkin, G. and Rosenschein, J.S.: Coalition, cryptography and stability: Mechanisms for coalition formation in task oriented domain, *Proc. 12th National Conference on Artificial Intelligence* (*AAAI*), pp.432–437 (1994).
12) 横尾　真，Conitzer, V.，Sandholm, T.，大田直樹，岩崎　敦：匿名の開環境下における協力ゲームについて，情報処理学会論文誌，Vol.47, No.5, pp.1451–1462 (2006).

(平成 21 年 3 月 23 日受付)
(平成 21 年 9 月 11 日採録)

大田　直樹

2005 年九州大学工学部電気情報工学科卒業．2007 年同大学院修士課程修了．同年同大学院博士後期課程に進学．協力ゲーム理論，オークション，分散制約充足問題に興味を持つ．

岩崎　　敦

2002 年神戸大学大学院自然科学研究科博士後期課程修了．同年より 2004 年まで NTT コミュニケーション科学基礎研究所に勤務．2004 年より九州大学大学院システム情報科学研究院助教．ゲーム理論，学習，オークション，実験経済学に関する研究に興味を持つ．博士（学術）．2004 年第 3 回合同エージェントワークショップ（JAWS2004）論文賞受賞．Economic Science Association 会員．

横尾　　真（正会員）

1984 年東京大学工学部電子工学科卒業．1986 年同大学院修士課程修了．同年日本電信電話（株）入社．2004 年より九州大学大学院システム情報科学研究院教授．エージェントの合意形成メカニズム，制約充足/分散制約充足等に興味を持つ．博士（工学）．1992 年，2002 年人工知能学会論文賞，1995 年情報処理学会坂井記念特別賞，2004 年 Association for Computing Machinery（ACM）Special Interest Group on Artificial Intelligence（SIGART）Autonomous Agent Research Award，2006 年日本学士院学術奨励賞受賞．日本ソフトウェア科学会，電子情報通信学会，AAAI 各会員．

ヴィンセント　コニッツァー

2001 年 Harvard University 応用数学科卒業．2003 年 Carnegie Mellon University コンピュータサイエンス学科修士課程修了．2006 年同大学院博士課程修了．同年より Duke University Assistant Professor．博士（Computer Science）ゲーム理論，メカニズムデザイン，オークション等のエージェント間の交渉 i メカニズムに興味を持つ．

トゥオマス　サンドホルム

1991 年 Helsinki University of Technology 卒業．1994 年 University of Massachusetts 修士課程修了．1996 年同大学院博士課程修了．2000 年まで Washington University 准教授．2006 年まで Carnegie Mellon University 准教授．同年より同大学教授．博士（Computer Science）．2001 年 Inaugural ACM Autonomous Agents Research Award，2003 年 Computer and Thought Award by the International Joint Conference on Artificial Intelligence 受賞．