1C3-3

特性関数の簡略記述法を用いた提携構造形成問題の近似アルゴリズムの提案及び比較

Approximation Algorithms for Coalition Structure Generation Utilizing Compact Characteristic Function Representations

長谷川 隆人 Takato Hasegawa　一村 良 Ryo Ichimura　大田 直樹 Naoki Ohta　岩崎 敦 Atsushi Iwasaki　横尾 真 Makoto Yokoo

九州大学大学院システム情報科学府
Graduate School of ISEE, Kyushu University

Coalition structure generation (CSG) involves partitioning a set of agents into coalitions so that social surplus is maximized. This problem has become a popular research topic in AI and multi-agent systems due to its computational complexity. Traditionally, the value of a coalition is assumed to be given by a black-box function called *characteristic function*. In our previous work, we proposed a new method for solving CSG assuming a characteristic function is compactly represented by a set of rules. In our previous work, we used an off-the-shelf mixed integer programming (MIP) package for finding an optimal solution. In this work, we develop a greedy approximation algorithm, which is much faster than MIP and achieves a good approximation rate. Furthermore, we develop a more sophisticated algorithm, which utilizes a linear programming relaxation of an original MIP. We show that if a problem instance has a certain structure, using the sophisticated algorithm is worthwhile.

1. はじめに

協力ゲーム理論は，複数のエージェントがどのように協力関係（提携）を形成し，提携内で得られた利得をどのように配分するかに関する理論である [岡田 96]．協力ゲーム理論はフォン・ノイマン以来の伝統ある研究分野であり，近年のインターネットの発展により，その適用分野が拡大している．

提携構造形成問題 (Coalition Structure Generation problem, CSG) に関する研究は，協力ゲーム理論の一分野であり，あるエージェントの集合を，全体の効用の総和が最大化されるように複数の提携に分割することを目的とする．この問題は，最適解を得るために多大な計算量が必要とされるため，近年，AI やマルチエージェント分野の研究者の注目を集めている．

従来の提携構造形成問題の研究では，提携に属するエージェントが協力することによって得られる利得は，特性関数と呼ばれるブラックボックスの関数によって与えられることを仮定していた．近年，提携内の利得の配分方法を与える，提携形ゲームの解概念に関する研究において，特性関数をルールの集合によって簡潔に表現する方法が提案されている [Conitzer 04, Conitzer 06, Ieong 05]．

我々は，特性関数がこのようなルールの集合によって表現されている場合に，提携構造形成問題を，いくつかの制約の下で，効用を最大にするようなルールの部分集合を発見する問題として定式化し，市販の混合整数計画法のパッケージ (CPLEX) によって，提携構造形成問題を解く手法を開発している [Ohta 09]．従来手法ではエージェント数が 30 程度の問題を解くために数時間を要していたが [Rahwan 07]，本手法により，エージェント数が 120 の問題でも，ルールの数がエージェント数と同程度の場合，20 秒程度で最適解を求めることが可能になっている．

しかしながら，本手法を用いた場合でも，提携構造形成問題を解くことは，ルール数に対して NP 困難であり，ルール数に関して最悪ケースの計算量が指数的となることは避けられない．また，混合整数計画法を用いた場合，大規模な問題に関しては，補助変数および制約の数が膨大となり，問題を主記憶に格納することが不可能となる．このため，本研究では，準最適解を高速に発見する近似アルゴリズムを提案する．さらに計算機実験によって，これらの近似アルゴリズムの最適解に対する近似率と実行時間を，様々な問題設定の下で評価する．

連絡先: 長谷川 隆人，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，hasegawa@agent.is.kyushu-u.ac.jp

2. モデル

協力ゲームでは特性関数が提携のもたらす効用を決定する．特性関数 v とは，エージェントの全体集合を A とするとき，その部分集合であるエージェントの集合（提携）S を引数とする関数であり，$v(S)$ は提携 S に属するエージェントが協力して得る効用を示す．

定義 1 (特性関数) 特性関数 $v : 2^A \to \mathbb{R}$ は，任意のエージェントの集合 S に対し，S に属するエージェントが協力した際に得る効用 $v(S)$ を与える．

特性関数は通常，ゲームの特徴に対応した性質を持ち，その代表的な性質として優加法性が存在する．優加法性とは，$S_i \cap S_j = \emptyset$ を満たす任意の提携 S_i, S_j について，$v(S_i) + v(S_j) \leq v(S_i \cup S_j)$ が成立する性質のことをいう．特性関数が優加法性を満たす場合，提携の人数が増えることによって個々のエージェントの効用が減少することはない．このとき全体の効用を最大化するには，人数が最大の提携，すなわち全体提携を形成すればよい．しかし，特性関数が優加法性を満たさない場合，全体提携が効用の総和を最大化するとは限らない．そのような場合，それぞれのエージェントが適切な提携に含まれるような，提携の構造を考える必要がある．

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．これを $CS = \{S_1, S_2, \ldots\}$ とするとき，CS を提携構造といい，CS は以下を満たす．

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

提携構造のもたらす効用 $V(CS)$ は，CS に含まれるすべての提携の効用の和，すなわち $V(CS) = \sum_{S_i \in CS} v(S_i)$ である．

提携構造形成問題における解とは，$V(CS)$ が最大となる提携構造，すなわち $\forall CS,\ V(CS^*) \geq V(CS)$ を満たす提携構造 CS^* である．

3. 特性関数の簡略記述法を用いた最適アルゴリズム

本章では，提携構造形成問題に利用可能な特性関数の簡略記述法として，MC-nets を紹介する．そして，MC-nets を用いた提携構造形成問題の解を求めるアルゴリズムを紹介する．

3.1 MC-nets

Ieong と Shoham は，Marginal Contribution networks (MC-nets) と呼ばれる特性関数の簡略記述法を提案した [Ieong 05]．この手法は，提携の効用をルールの集合によって記述することで，特性関数を簡略に記述するものである．

定義 2 (MC-nets) *MC-nets* は，提携が満たすべきルールの集合 R によって表される．任意のルール $r \in R$ は，$(P_r, N_r) \to v_r$ という形で表される．P_r, N_r はそれぞれ "存在しなければならない" および "存在してはならない" エージェントの集合であり，$P_r \subseteq A, N_r \subseteq A,\ P_r \cap N_r = \emptyset$ である．$v_r \in \mathbb{R}$ は，ルール r が満たされた場合の効用を表す．$P_r \subseteq S$ *and* $N_r \cap S = \emptyset$ が成り立つとき，ルール r は提携 S に適用可能であるといい，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 の効用は $v(S) = \sum_{r \in R_S} v_r$ で与えられる．したがって，任意の提携構造の効用は $V(CS) = \sum_{S \in CS} \sum_{r \in R_S} v_r$ である．

従来，MC-nets に含まれるルールは，正負両方の効用を持ちうると定義していた．本論文では，「どのルールも必ず正の効用を持つ」という制限を設ける．また，任意のルール r について $|P_r| \geq 1$ と仮定する．これらの制限の下でも，MC-nets を用いてあらゆる特性関数を記述できる．

例 1 5 人のエージェント a, b, c, d, e による協力ゲームにおいて，$r_1 : (\{b, e\}, \{\}) \to 3$，$r_2 : (\{a, b, c\}, \{d\}) \to 2$，$r_3 : (\{a, d\}, \{\}) \to 1$，$r_4 : (\{c\}, \{e\}) \to 1$ という 4 つのルールが特性関数を簡略に記述しているとする．このとき，$\{a, b, c\}$ という提携にはルール r_2 と r_4 が適用可能であり，この提携の効用は $v(\{a, b, c\}) = v(r_2) + v(r_4) = 2 + 1 = 3$ である．

3.2 MC-nets を用いた提携構造の形成

MC-nets を利用することで，特性関数を簡略に記述可能である．また，任意の提携 S のもたらす効用 $v(S)$ は，S に適用可能なルール r の効用 v_r の総和となる．本節では，MC-nets によって記述されるルールのグラフ表現を定義し，それを基に提携構造形成問題の解を求める方法について述べる．

定義 3 (実現可能なルール集合) ルールの集合 $R' \subseteq R$ について，任意のルール $r \in R'$ が少なくとも *1* つの提携に適用可能であるような提携構造 CS が存在するとき，"R' は実現可能なルール集合である" という．

例 1 より，$\{r_2, r_4\}$ というルール集合は，$\{\{a, b, c\}, \{d, e\}\}$ という提携構造によって実現可能である．一方，$\{r_1, r_2, r_4\}$ や $\{r_2, r_3\}$ というルール集合は実現不可能である．MC-nets を基に最適な提携構造を発見することは，$\sum_{r \in R'} v_r$ を最大化するような，実現可能なルール集合 R' を発見することに等しい．

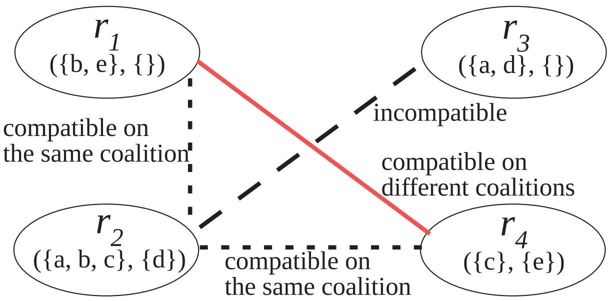

図 1: 例 1 に基づくルールのグラフ表現

定義 4 (ルール間の関係) 任意の 2 つのルール r と r' の間には，以下に示すような 4 種の関係が存在する．これらの関係は，同じルールの組に同時に複数成り立つことはなく，これらの関係のどれにも含まれないようなルールの組は存在しない．

同提携で両立可能 (Compatible on the same coalition) $P_r \cap P_{r'} \neq \emptyset$ *and* $P_r \cap N_{r'} = P_{r'} \cap N_r = \emptyset$ が成立するとき，ルール r と r' は同提携で両立可能である．

両立不可能 (Incompatible) $P_r \cap P_{r'} \neq \emptyset$ *and* $(P_r \cap N_{r'} \neq \emptyset$ *or* $P_{r'} \cap N_r \neq \emptyset)$ が成立するとき，ルール r と r' は両立不可能である．

他提携で両立可能 (Compatible on different coalitions) $P_r \cap P_{r'} = \emptyset$ *and* $(P_r \cap N_{r'} \neq \emptyset$ *or* $P_{r'} \cap N_r \neq \emptyset)$ が成立するとき，ルール r と r' は他提携で両立可能である．

独立 (Independent) $P_r \cap P_{r'} = \emptyset$ *and* $P_r \cap N_{r'} = P_{r'} \cap N_r = \emptyset$ が成立するとき，ルール r と r' は独立である．

MC-nets によるルールの集合は，各ルールをノード，ルール間の関係をエッジとしたグラフで表現できる．図 1 に，例 1 のグラフ表現を示す．ただし，"独立" を示すエッジは省略する．

以下の定理にルール集合が実現可能であるための必要十分条件を記述する．

定理 1 ルールの集合 R' は，以下の条件を同時に満たすとき，またそのときに限り実現可能である．

(a) R' は，"両立不可能" なエッジで連結されたルール（ノード）の組を持たない．

(b) R' に含まれる任意のルール（ノード）の組が "他提携で両立可能" なエッジで連結されているとき，このルール（ノード）の組は "同提携で両立可能" なエッジによって到達不可能である．

定義 5 (MC-nets を用いた CSG の混合整数計画法表現) $\sum_{r \in R'} v_r$ を最大化する実現可能なルールの集合 R' を発見する問題は，混合整数計画法として表現できる．紙幅の都合上，混合整数計画法の定式化は省略する．詳細については *[Ohta 09]* を参照されたい．

4. 貪欲法を用いた近似アルゴリズムの提案

本章では MC-nets において，貪欲法 [Kleinberg 05] を用いた近似アルゴリズムを提案する．前章の簡略記述法を用いた手法でも，提携構造形成問題を解くことは NP 困難で，ルール数に関して最悪ケースの計算量が指数的となることは避けられない．また，混合整数計画法を用いた場合，大規模な問題に関しては，問題を主記憶に格納することが不可能となる．このため本章では準最適解を高速に発見する近似解法の開発を行う．

MC-nets を用いた貪欲法とは，ルールを取捨選択する順番の優先度の高いほうから順にルール集合の実現可能性 (定理 1) の条件を満たすルールを選択する手法である．ルールの優先度については 5 章に後述する．

定義 6 (貪欲法を用いた近似アルゴリズム) *MC-nets* $R = \{r_1, r_2, \ldots\}$ に関して,各ルールに優先度 $p : R \to \mathbb{R}$ が与えられ, $\forall(i > j), p(r_i) \leq p(r_j)$ のように R が優先度の降順になっている．このとき，以下のように実現可能なルールの集合 R' を発見する．

まず，$i = 1, 2, \ldots$ の順に以下に従って実現可能なルールの集合 R' を作成する．初期状態では，$R' = \emptyset$ とする．

1. r_i が R' の任意のルールと "両立不可能" のエッジで連結される場合は r_i は R' に加えない．

2. *1.* 以外で，r_i が R' のいずれのルールとも "同提携で両立可能" のエッジで連結されない場合は r_i を R' に加える．

3. *1. 2.* 以外，すなわち r_i が R' の任意のルールと "同提携で両立可能" のエッジで連結される場合,

  (a) r_i と "同提携で両立可能" のエッジで連結されるルールの集合について，"他提携で両立可能" のエッジで連結されたルールの組を持たない場合，r_i を R' に加える．

  (b) "他提携で両立可能" のエッジで連結されたルールの組を持つ場合，r_i は R' に加えない．

最終的に R' が実現可能なルールの集合となる．

貪欲法を用いた近似アルゴリズムの例は次のようになる．

例 *2* 例 1 の *MC-nets* について考える．$p(r_i) = v_{r_i}$ とし，$p(r_i)$ の降順に並べると，$r_1 : (\{b, e\}, \{\}) \to 3$, $r_2 : (\{a, b, c\}, \{d\}) \to 2$, $r_3 : (\{a, d\}, \{\}) \to 1$ $r_4 : (\{c\}, \{e\}) \to 1$, となる．最初に r_1 を選択する．次に r_2 について，r_1 と実現可能なルールの集合の条件を満たすので選択する．さらに r_3 について，r_2 と "両立不可能" の関係にあるので，選択しない．最後に r_4 について，r_1, r_2 の双方と "同提携で両立可能" の関係にあり，かつ r_1, r_4 間が "他提携で両立可能" の関係にあるので，r_4 は選択しない．以上から，$R' = \{r_1, r_2\}$ となり，$CS = \{\{a, b, c, e\}, \{d\}\}, V(CS) = 5$ となる．

5. 近似アルゴリズムの評価

本章では，提案した近似アルゴリズムを用いて CSG を実際に解く計算機実験を行い，その性能を評価する．本章の実験は Pentium Xeon E5540 2.5GHz プロセッサと 12GB メモリを搭載した Windows Vista Business 64bit Edition マシンで行った．また，各アルゴリズムで表現される線形／整数計画法問題を解くため，市販の最適化エンジンである ILOG CPLEX 11.2 を使用した．本章では，貪欲法を用いた近似アルゴリズムにおいて，各ルールの優先度 $p(r_i)$ を以下の 2 つの方法で決定する．

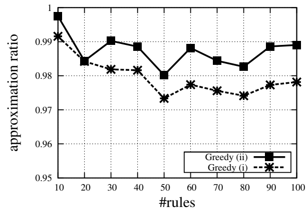

図 2: CATS における各実装の近似率

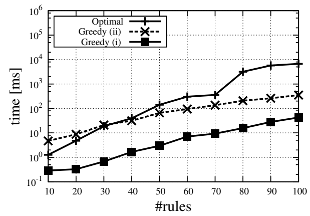

図 3: CATS における各実装の実行時間

貪欲法 (i) ルールの優先度 $p(r_i)$ をそのルールがもつ効用 v_{r_i} を基準にして決定する．すなわち $p(r_i) = v_{r_i}$ とする．

貪欲法 (ii) ルールの優先度 $p(r_i)$ をそのルールがもつ効用 v_{r_i} と線形計画問題の解 $L(r_i)$ との積を基準にして決定する．ここで，線形計画問題の解とは定義 5 において整数制約を外して解くことで得られる値とする．つまり，$p(r_i) = v_{r_i} \times L(r_i)$ とする．

5.1 CATS に基づく実験

本節では，ある種の重み付きナップサック問題を生成するソフトウェアである Combinatorial Auction Test Suite (CATS) [Leyton-Brown 00] を基に生成した提携構造形成問題において貪欲法 (i) および (ii) を評価する．

本論文では以下の手順に基づき CATS を用いて提携構造形成問題を生成する．まず，n 人のエージェントに対して $|R|$ 個のルールを持つ MC-nets を生成するために，$|R|$ 個の提携 S を CATS から生成する．具体的には，CATS が設定する減衰分布 (decay distribution) を用いて，ランダムに選んだエージェント 1 人からなる提携を形成し，その提携に確率 α で他のエージェントを 1 人ずつ追加する．エージェントを追加しなくなるか，これ以上エージェントを追加できない場合は動作を終了し，1 つの提携を決定する．以上の動作を繰り返すことで任意個の提携を生成する．ただし，それぞれの提携の重複を許さないとする．

次に，生成した任意の提携 S に一様分布を用いて効用 $v(S) \in [|S|, 1000|S|]$ をランダムに決定する．加えて，その任意の提携 S に $(S, \{\}) \to v(S)$ というルールを生成する．このルールを本論文で定義した MC-nets に対応させるため，確率 p で "存在しなければならない" エージェントを "存在してはならない" エージェントへと変化させる．ただし，"存在しなければならない" エージェントの集合が空になる，すなわち存在しないことは許さないとする．

以上より，ルール数とエージェント数が等しいという条件で，ルール数を 10 から 100 の間で変化させながら生成した 50 個の問題に対して，問題の最適解に対する近似率 (図 2) と求解に必要な時間 (図 3) を測定した．

図 2 および図 3 では，横軸は常にエージェント数およびルール数を表す．図 2 における縦軸は 50 個の問題に対する近似率の平均を，一方で図 3 における縦軸は 50 個の問題に対する実行時間の中央値を表す．また，$\alpha = 0.55$ および $p = 0.2$ と設定したが，これらを変化させても本質的な結果は変わらなかった．

図 2 において，貪欲法 (i) および (ii) はともにルール数が増加しても 97%以上の高い近似率を示した．図 3 において，貪

欲法 (i) の実行時間が最も速く，貪欲法 (ii) は貪欲法 (i) に比べて十倍程度遅くなっているが，最適解を算出する場合に比べて数十倍速い実行時間を示した．したがって，この実験では近似率および実行時間の両方に関して貪欲法 (i) がもっとも優れた性能を示したといえる．しかし，この結果は貪欲法 (i) が提携構造形成問題がどんな構造をもっていても優れた性能を発揮することを保証する訳ではない．実際，次節では貪欲法 (i) が十分よい性能を示さない問題の構造について議論する．

5.2 ランダムグラフに基づく実験

本節では，貪欲法 (i) が最適解および貪欲法 (ii) に比べて不利になるルールの構造（グラフ構造）の例を示し，それにもとづいて生成したランダムグラフにおいて提案手法を評価する．

例 3 ある協力ゲームを 4 つのルール r_1, r_2, r_3, r_4 で表現できる場合を考える．それぞれのルールが与える効用は $v(r_1) = 10, v(r_2) = 9, v(r_3) = 8, v(r_4) = 7$ とし，r_1, r_2 が "他提携で両立可能"，r_3 が r_1, r_2 双方と "同提携で両立可能"，r_4 が r_1, r_2 双方と "同提携で両立可能" の関係にあるとする．ここで，貪欲法 *(i)* が与える解は r_1, r_2 を適用した $v(CS) = 19$ となる一方で，最適解および貪欲法 *(ii)* が与える解は r_1, r_3, r_4 を適用した $v(CS) = 25$ となる．

この例は，提携構造形成問題において，"他提携で両立可能" のエッジがある一定数存在する状況に，"同提携で両立可能" のエッジが張り巡らされているような構造を示している．このような構造の元では貪欲法 (i) が不利になることが予想される．

そこで，本節では以下の手順で生成した提携構造形成問題において貪欲法 (i) および (ii) を評価する．まず，以下のようなランダムグラフを考える．ランダムグラフとは，MC-nets の各エッジの本数を入力とし，それらのエッジをランダムに配置するグラフである．ここで入力するエッジは次の 3 種類とし，"同提携で両立可能" のエッジの数を same，"両立不可能" のエッジの数を inc，"他提携で両立可能" のエッジの数を dif とする．また，MC-nets $R = \{r_1, r_2, \ldots\}$ における各ルールの効用は整数値をランダムに取るとする．以上より，それぞれのエッジの数 $(same, inc, dif) = (3|R|, 0, |R|/5)$ とし，ルール数 10 から 100 の間で評価した．この結果，貪欲法 (i) では CATS に基づく実験では最低でも 0.97 であったのに対して，この実験では最高でも 0.94 にまで低下した．一方で，貪欲法 (ii) では CATS に基づく実験では 0.98 であったのに対して，この実験でもほぼ同等の近似率を達成した．この結果，貪欲法 (i) が性能を十分に達成できない構造において，貪欲法 (ii) は安定した性能を発揮していることがわかった．

6. 考察

CATS に基づく実験では，貪欲法 (i) が高い近似率および高速な実行時間を達成したため，効率的に解を算出できたといえる．一方で，貪欲法 (ii) はルールの優先度を単にルールの効用にするのではなく，線形計画問題の解との積にすることで，貪欲法 (i) よりも高い近似率を達成している．これは，与えられた提携構造形成問題の最適解に近い解が線形計画問題の解として得られることで，問題の構造を反映できたためと考えられる．その代わり，貪欲法 (ii) は線形計画問題を解いているため，計算量が増加する結果，当然その実行時間は貪欲法 (i) より遅くなる．

さらに，貪欲法 (i) が不利になる構造の例を示し，実際に，その構造に基づいて生成した問題において貪欲法 (i) の近似率が低下することを示した．これは貪欲法 (i) はルールがもつ効用のみを判断基準としているため，優先順位の高い 1 つのルールが与える効用が優先順位の低い複数のルールの効用の合計より小さくなるケースを判断することができないためである．しかし一方で，貪欲法 (ii) は線形計画問題の解を利用することで，このような近視眼的な選択を回避する結果，貪欲法 (i) よりも優れた近似率を達成するといえる．

7. 結論

本論文では，特性関数の簡略記述法である MC-nets を利用した提携構造形成問題の貪欲法に基づく近似アルゴリズムを提案・評価した．計算機実験において，提案手法は精度の高い近似解を算出するだけでなく，混合整数計画法では求解不可能な規模でも求解可能となる．さらに，貪欲法に基づく近似アルゴリズム (貪欲法 (i)) が不利になる構造を示し，それを改善するために線形計画法の解を利用するアルゴリズム（貪欲法 (ii)）を提案・評価した．こちらでも計算機実験の結果，優れた精度の高い近似解と実行時間を観察した．

今後の課題として，示した貪欲法 (i) が不利になる構造の実問題への応用を考えるともに，実問題の構造を推定した上で新たな近似アルゴリズムを提案することなどが挙げられる．

参考文献

[Conitzer 04] Conitzer, V. and Sandholm, T.: Computing Shapley values, manipulating value division schemes, and checking core membership in multi-issue domains., in *Proc. of the 19th National Conf. on Artificial Intelligence (AAAI)*, pp. 219–225 (2004)

[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 *Proc. of the 6th ACM Conf. on Electronic Commerce (ACM EC)*, pp. 193–202 (2005)

[Kleinberg 05] Kleinberg, J. and Tardos, E.: *Algorithm Design*, Addison-Wesley (2005)

[Leyton-Brown 00] Leyton-Brown, K., Pearson, M., and Shoham, Y.: Towards a universal test suite for combinatorial auction algorithms, in *Proc. of the 2nd ACM Conf. on Electronic Commerce (ACM EC)*, pp. 66–76 (2000)

[Ohta 09] Ohta, N., Conitzer, V., Ichimura, R., Sakurai, Y., Iwasaki, A., and Yokoo, M.: Coalition Structure Generation Utilizing Compact Characteristic Function Representations, in *Principles and Practice of Constraint Programming - CP 2009*, pp. 623–638 (2009)

[Rahwan 07] Rahwan, T., Ramchurn, S. D., Dang, V. D., Giovannucci, A., and Jennings, N. R.: Anytime Optimal Coalition Structure Generation., in *Proc. of the 22nd Conf. on Artificial Intelligence (AAAI)*, pp. 1184–1190 (2007)

[岡田 96] 岡田 章：ゲーム理論, 有斐閣 (1996)