1C3-4

分散制約最適化問題に基づく提携構造形成問題

Coalition Structure Generation based on Distributed Constraint Optimization

上田 俊[*1] Suguru Ueda　岩崎 敦[*1] Atsushi Iwasaki　横尾 真[*1] Makoto Yokoo　平山 勝敏[*2] Katsutoshi Hirayama　松井 俊浩[*3] Toshihiro Matsui

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

[*2]神戸大学大学院海事科学研究科
Graduate School of Maritime Sciences, Kobe University

[*3]名古屋工業大学
Nagoya Institute of Technology

Forming effective coalitions is a major research challenge in AI and multi-agent systems. Coalition Structure Generation (CSG) involves partitioning a set of agents into coalitions so that social surplus is maximized. A partition is called a coalition structure (CS). In this paper, we propose a novel formalization of CSG, i.e., we assume that the value of a characteristic function is given by an optimal solution of a distributed constraint optimization problem (DCOP) among the agents of a coalition. Also, we develop an algorithm with parameter k that can find a CS whose social surplus is at least $\max(k/(w^*+1), k/\lfloor n/2 \rfloor)$ of the optimal CS, where w^* is the tree width of a constraint graph. These results illustrate that the locality of interactions among agents, which is explicitly modeled in the DCOP formalization, is quite useful in developing an efficient CSG algorithm with quality guarantees.

1. はじめに

利己的なエージェント間で協調関係を結ぶことが可能な協力ゲーム [岡田 96] において，各エージェントは自らの利得を増やすために他のエージェントと提携を組むことができる．提携構造形成問題 (CSG, Coalition Structure Generation) とは，あるエージェントの集合を社会的余剰 (すべての提携の得る利得の合計) が最大化されるようにいくつかの提携に分割する問題である．この問題は，AI やマルチエージェントの分野において近年注目を集めている．提携構造形成問題の研究は，分散経路決定問題 (distributed vehicle routing) やマルチセンサネットワークなどの分野に応用できる．また，提携構造形成問題は完全集合分割問題と等価であり，これまでに解を得るための様々なアルゴリズムが提案されている [Sandholm 99, Rahwan 08a, Rahwan 08b].

従来の提携構造形成問題の研究では，提携の得る利得は特性関数と呼ばれるブラックボックス関数 (オラクル) により与えられることを仮定していた．ここで，提携の得る利得が現実にはどのように与えられるのかを考える．提携の利得は，提携に属するエージェントが協力して行動したときに達成される最適な利得であり，これは，提携に属するエージェント間のなんらかの最適化問題の解で与えると考えることができる (このようなアイデアは文献 [Sandholm 99] でも指摘されている). 本論文では，提携の利得は提携に属するエージェント間の分散制約最適化問題 (DCOP, Distributed Constraint Optimization Problem) [Modi 03] の最適解で与えられると仮定する．すなわち，特性関数を分散制約最適化問題を用いて表現するアプローチを提案する[*4].

分散制約最適化問題は人工知能やマルチエージェントの様々な問題を表現する代表的な枠組みであり，ADOPT [Modi 03], DPOP [Petcu 05] 等の様々なアルゴリズムが提案されている．この問題では，エージェントは行動の選択肢を持ち，エージェントの行動の組合せに対して利得やコストが定義されている．それらを制約と呼び，その利得やコストの総和を最大化もしくは最小化するエージェントの行動の組合せを探索する．分散制約最適化問題はきわめて一般的な手法であり，マルチエージェントシステムにおける様々な問題を表現できる．

既存の提携構造形成問題を解くアルゴリズムでは，すべての提携に関してその利得が既知であることを仮定している．提案手法では，分散制約最適化問題の最適解を提携の利得とするため，既存のアルゴリズムを用いる場合，あらかじめ提携の利得を計算しておかなければならない．しかし，分散制約最適化問題は NP 困難な問題であるため，エージェント数を n とすると，NP 困難な問題を $O(2^n)$ 回も解かなければならず，計算量が膨大となる． しかしながら，本論文では，全体提携 (エージェント全員からなる提携) の利得を求める場合と同程度の計算量で，比較的良い提携構造を求めることができる近似アルゴリズムを提案する．提案アルゴリズムは解の精度を理論的に保証できる．k をアルゴリズムのパラメータ，w^* を制約グラフの樹状幅とすると，最悪時での得られた解の利得と最適解の比率は $\max(k/(w^*+1), k/\lfloor n/2 \rfloor)$ で与えられる．さらに，計算機実験により，提案アルゴリズムの平均的な解の精度は，最悪時の理論的に保証される値よりもはるかに良いことを示す.

2. 分散制約最適化問題による特性関数の表記法

2.1 モデル

エージェントの集合を $T = \{1, 2, \ldots, n\}$ と表す．特性関数 $v : 2^T \to \mathbb{R}$ は，エージェントの提携 S に対し，S に属するエージェントが協力して行動する際に得る利得 $v(S)$ を与える．本論文では，提携 S の利得 $v(S)$ は提携 S に属するエージェント間での分散制約最適化問題の最適解により与えられると仮定する．

ここでの分散制約最適化問題は以下のように定式化される．

連絡先: 上田 俊，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, ueda@agent.is.kyushu-u.ac.jp

*4 紙幅の都合上，各定理の証明は省略する．詳細については [Ueda 10] を参照されたい.

エージェント i はエージェントの選んだ選択肢 (行動) を表す変数 x_i を持つ．エージェント i は有限で離散的な領域 D_i から変数 x_i の値を選択する．これらの変数の値に対して，次のような制約が存在する．変数 x_i の値 $d_i \in D_i$ に対して単項制約 $r_i : D_i \to \mathbb{R}$ が存在し，エージェント i が行動 d_i を選択したときの利得 $r_i(d_i)$ を与える．ただし，$r_i(d_i) \geq 0$ となる値 $d_i \in D_i$ が少なくとも一つは存在すると仮定する．また，変数 x_i, x_j の間に二項制約 $r_{i,j} : D_i \times D_j \to \mathbb{R}$ が存在する．この制約は，エージェント i, j が同じ提携に属し，それぞれのエージェントが d_i と d_j を選択したときの利得 $r_{i,j}(d_i, d_j)$ を与える．本論文では，簡単化のため変数の値に対する制約は単項制約および二項制約に限定するが，本論文で示す結果は $k(\geq 3)$ 項制約が存在するときも成り立つ．すべての変数への値の割り当てを $A \in \Pi_{i \in S} D_i$ とすると，提携 S の利得 $v(S)$ は次の式で与えられる：

$$\max_A \{\sum_{i \in S} r_i(d_i) + \sum_{i,j \in S} r_{i,j}(d_i, d_j)\}.$$

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 $CS = \{S_1, S_2, \dots\}$ とするとき，提携構造 CS は以下を満たす：

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

つまり，提携構造 CS において，それぞれのエージェントはただ一つの提携に属し，複数の提携に属することはない．

例えば，3 人のエージェント a, b, c によるゲームにおいて，$\{a\}$，$\{b\}$，$\{c\}$，$\{a, b\}$，$\{b, c\}$，$\{a, c\}$，$\{a, b, c\}$ の 7 つの提携が存在し，$\{\{a\}, \{b\}, \{c\}\}$，$\{\{a, b\}, \{c\}\}$，$\{\{a\}, \{b, c\}\}$，$\{\{b\}, \{a, c\}\}$，$\{\{a, b, c\}\}$ の 5 つの提携構造が存在する．

提携構造 CS の利得 $V(CS)$ は，CS に含まれる提携の利得の和，すなわち $V(CS) = \sum_{S_i \in CS} v(S_i)$ で表される．これを単項制約および二項制約を用いて表すと次のようになる：

$$V(CS) = \max_A \{\sum_{i \in S \in CS} r_i(d_i) + \sum_{i,j \in S \in CS} r_{i,j}(d_i, d_j)\}.$$

提携構造の中で利得が最大になる提携構造を CS^* と表し，次の式を満たす：

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

また，CS^* は単項制約および二項制約を用いて次のように表される：

$$CS^* = \arg\max_{CS} \max_A \{\sum_{i \in S \in CS} r_i(d_i) + \sum_{i,j \in S \in CS} r_{i,j}(d_i, d_j)\}.$$

通常の分散制約最適化問題では，全体提携 T の利得 $v(T)$ を最大にする変数の値の割り当てを探索する．そのため，得られる利得が最大となるような提携構造 CS^* と，このときの変数の値の割り当てを探索する．また，提携の利得はどのエージェントが属しているかのみで決まり，他にどのような提携が形成されているかには影響されない．これは，提携形ゲームでよく用いられる，外部性が存在しないという仮定に対応している．

特性関数には，優加法性や単調性といった性質が存在する．特性関数が優加法的であるとは，$S_i \cap S_j = \emptyset$ を満たす任意の提携の組 S_i, S_j について，$v(S_i) + v(S_j) \leq v(S_i \cup S_j)$ が成立することを意味する．また，単調性とは，$S \supset S'$ を満たす提携 S, S' について，$v(S) \geq v(S')$ が成立する性質のことをいう．特性関数が優加法性や単調性を満たす場合，提携構造形成問題の解は自明であり全体提携が最適な提携構造となる．そのため本論文では，特性関数をこのような性質を満たすものに限定しない．また，提案手法は任意の特性関数を表現可能であるため，優加法性や単調性を満たさない特性関数も表現できる．

ここで，分散制約最適化問題に基づく提携構造形成問題の例を示す．

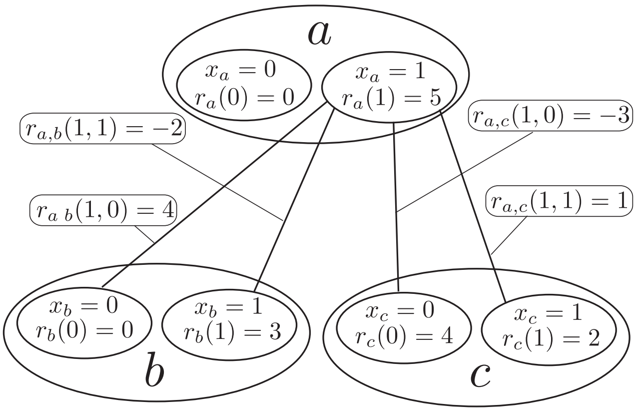

図 1: 例 1 における単項・二項制約

例 1 3 人のエージェント a, b, c が存在し，それぞれのエージェントは 2 つの行動の選択肢をもつとする．すなわち，3 つの変数 x_a, x_b, x_c が存在し，それぞれの変数の領域は $\{0, 1\}$ であるとする．単項・二項制約を以下のように与える．ここに示していない単項・二項制約が与える利得は *0* とする（図 *1*）：

$$\begin{aligned} &r_a(1) = 5, && r_b(1) = 3, \\ &r_c(0) = 4, && r_c(1) = 2, \\ &r_{a,b}(1,0) = 4, && r_{a,b}(1,1) = -2, \\ &r_{a,c}(1,0) = -3, && r_{a,c}(1,1) = 1. \end{aligned}$$

提携 $\{a, b, c\}$ の利得 $v(\{a, b, c\})$ は 3 人のエージェント間の分散制約最適化問題の最適解で与えられる．このとき，最適な変数の割り当ては，$x_a = 1, x_b = 0, x_c = 1$ となり，提携 $\{a, b, c\}$ の利得は，$v(\{a, b, c\}) = r_a(1) + r_b(0) + r_c(1) + r_{a,b}(1,0) + r_{b,c}(0,1) + r_{a,c}(1,1) = 12$ となる．また，提携 $\{a, b\}$ の利得 $v(\{a, b\})$ はエージェント a，b 間の分散制約最適化問題の最適解で与えられる．このとき，最適な変数の割り当ては，$x_a = 1, x_b = 0$ となり，提携 $\{a, b\}$ の利得は，$v(\{a, b\}) = r_a(1) + r_b(0) + r_{a,b}(1,0) = 9$ となる．この場合，エージェント c が単独で行動するとき，$x_c = 0$ とすることで 4 の利得を得ることができる．したがって，提携構造 $\{\{a, b\}, \{c\}\}$ の利得は 13 となる．この場合は全体提携が最適な提携構造とはならず，最適な提携構造は $\{\{a, b\}, \{c\}\}$ である．

2.2 提案手法の表記量および計算量

ブラックボックス関数を用いて特性関数を表記するためには，2^n 個存在するすべての提携の利得を表記しなければならず，$O(2^n)$ の表記量が必要となる．一方で，分散制約最適化問題を用いて表記する場合は，その制約の数が表記量となる．制約の数が少ないとき，つまり単項・二項制約のみが存在する場合，制約の数は $O(n^2)$ 個となるため，表記量は $O(n^2)$ となる．したがって，エージェント数に関する多項式のサイズで特性関数を簡略に表現できる．

また，従来から特性関数を簡略に表記する手法は様々なものが提案されており，Marginal contribution networks (MC-nets) [Ieong 05] や Synergy coalition group (SCG) [Conitzer 06] といった手法が存在する．これらの手法では，ルールの集合を用いて特性関数を簡略に表記し，ルールの個数が特性関数の表記量となる．

MC-nets や SCG で簡略に表記できる特性関数は，提案手法でも同様に簡略に表記できる．その一方で，MC-nets や SCG で簡略に表記できないが，提案手法で簡略に表記できる特性関数が存在する．すなわち，以下の 2 つの定理が成り立つ．

定理 1 任意の *MC-nets* または *SCG* で簡略表記された特性関数は，分散制約最適化問題を用いて簡略に表記できる．すなわち，表記法を変換するときに必要な制約の数は，$O(n)$ でしか増加しない．

定理 2 *MC-nets* または *SCG* では簡略に表記できないが，分散制約最適化問題を用いて簡略に表記できる特性関数が存在する．

特性関数の簡略表記法における研究において，コアやシャプレイ値といった解概念の存在性の証明や値の計算に要する計算量は重要な分析対象である．分散制約最適化問題による表記法は，この観点からは良い表記法であるとは言えない．実際に，次の定理が示すように，ある配分 (提携に属するエージェントについての利得の分配) が実行可能 (配分の合計が提携の利得と等しいかより小さい) か判定する問題といった簡単な問題でも，分散制約最適化問題を解く必要があるため，NP 完全な問題である．

定理 3 特性関数を分散制約最適化問題を用いて表記するとき，ある配分が実行可能か判定する問題は，*NP* 完全である．

また，提案手法での提携構造形成問題は NP 困難な問題となる．すなわち，以下の定理が成り立つ．

定理 4 分散制約最適化問題に基づく提携構造形成問題の決定問題，すなわち，あるしきい値に等しい利得 $V(CS)$ を得る提携構造 CS が存在するかどうか判定する問題は *NP* 完全である．

3. 近似アルゴリズム

3.1 基本的なアイデアとアルゴリズムの詳細

提案アルゴリズムの基本となるアイデアは，複数エージェント提携 (複数エージェントで形成される提携) の数が制限された提携構造を探索することである．提案アルゴリズムは，高々 k 個の複数エージェント提携と残りのエージェントの単独提携 (1 人のエージェントのみで形成する提携) で形成される提携構造のみを探索する．この前提に対して，以下のように特性関数として与えられた分散制約最適化問題に変更を加え，新たな問題を作成する． 高々k 個の複数エージェント提携を含む提携構造を探索するために，各エージェントの値の領域に新たな値を加え，加えた値に関する制約を追加する．新しく加える値はエージェントの行動と属する提携の両方を表す値であり，最適解での変数の値から提携構造を作成する．

まず，各エージェントの持つ変数の値の領域に，新たな値 “independent” を追加する．この値はそのエージェントが独立して行動することを表す値である．つまり，この値を選んだエージェントは単独提携を形成する．一方，この値以外を選んだエージェントは，それらで提携を形成する．“independent” に対する制約を以下のように設定する．“independent” に対する単項制約の利得は，他の値に対する単項制約の値の最大値とする．また，少なくとも片方の値が “independent” である二項制約の値は 0 とする．次に，各エージェントの持つ変数の値の領域に，自身の領域のコピーを追加する．エージェント i は，“independent” に加えて，その変数の領域 D_i の k 倍のコピーを持つ．例えば，$D_i = \{d_1, d_2\}$ であり，$k = 2$ の場合を考える．このとき，エージェント i は新しい領域 $D'_i = \{d_{1,1}, d_{2,1}, d_{1,2}, d_{2,2}, independent\}$ を持つ．それぞれの値は，エージェントの選んだ選択肢 (行動) と，属する提携を同時に表す．これらの値に対する制約を以下のように設定する．コピーされた値に対する単項制約の値は，元の値に対するものと等しい値とする．上記の例では，$r_i(d_{1,1}) = r_i(d_{1,2}) = r_i(d_1)$ であり，$r_i(d_{2,1}) = r_i(d_{2,2}) = r_i(d_2)$ である．同じ提携に属していることを表す値どうしの二項制約は，元の値のものと等しい値とし，そうでなければ 0 とする．

このようにして作成した分散制約最適化問題は，ADOPT や DPOP といった様々な既存のアルゴリズムを用いて解くことができる．(また，集中型のアルゴリズム [Dechter 03] を用いて解くこともできる)．提案アルゴリズムは，この分散制約最適化問題の最適解を基に提携構造を作成する．具体的には，“independent” を選択しているエージェントは単独提携を形成し，j 番目のコピーされた値を選択しているエージェントが j 番目の複数エージェント提携に属している提携構造がこのアルゴリズムの解である．

元の分散制約最適化問題の変数の領域の大きさを d とすると，新しい分散制約最適化問題の探索空間は $(kd + 1)^n$ である．$k = 1$ のときは $(d + 1)^n$ であり，元の分散制約最適化問題を解くのと同程度の計算量で解を得ることができる．

3.2 最悪時の解の精度

提案アルゴリズムの最悪時の解の精度は，提携構造に含まれる複数エージェント提携の数に基づいて与えられる．

定理 5 近似アルゴリズムの最悪時の解の精度は k/l である．すなわち，得られた提携構造 CS^{ap}_k の利得 $V(CS^{ap}_k)$ と，最適な提携構造の利得 $V(CS^*)$ に対して，$V(CS^{ap}_k)/V(CS^*) \geq k/l$ が成り立つ．ただし，l は CS^* に含まれる複数エージェント提携の数である．

提携構造に含まれる複数エージェント提携の数は高々$\lfloor n/2 \rfloor$ であるため，直ちに以下の定理を得る．

定理 6 近似アルゴリズムの最悪時の解の精度は $k/\lfloor n/2 \rfloor$ である．すなわち，得られた提携構造 CS^{ap}_k の利得 $V(CS^{ap}_k)$ と，最適な提携構造の利得 $V(CS^*)$ に対して，$V(CS^{ap}_k)/V(CS^*) \geq k/\lfloor n/2 \rfloor$ が成り立つ．

制約グラフの樹状幅を w^* とすると，近似アルゴリズムの最悪時の解の精度について，$\lfloor n/2 \rfloor$ の代わりに $w^* + 1$ を用いることができる．樹状幅は，制約充足問題や制約最適化問題を含む，グラフに基づく最適化アルゴリズムの複雑性を表現する値として知られている [Dechter 03]．

樹上幅は，グラフがどの程度木に近いかという性質を表す値である [Diestel 05] *5．グラフが木である場合，樹状幅は 1

*5 樹上幅の定義は複雑であるため，紙幅の都合上省略する．詳細はグラフ理論の教科書 ([Diestel 05] 等) を参照されたい．

である．また，分散制約最適化問題を解くアルゴリズムでよく用いられるグラフ構造に，擬似木 (pseudo-tree) が存在する [Modi 03, Petcu 05]．擬似木にも，どの程度木に近いかという性質を表す値として induced width が存在する．樹状幅は，擬似木の induced width と密接に関連しており，樹状幅が w^* であれば，グラフ G の任意の擬似木の induced width は少なくとも w^* となる [Dechter 03]．

エージェントの集合をノード，二項制約をエッジとみなした制約グラフを考える．このとき，次の定理が成り立つ．

定理 7 制約グラフの樹状幅が w^* であるとき，高々w^*+1 個の複数エージェント提携を含む最適な提携構造 CS^* が存在する．

定理 5 および定理 7 から，直ちに以下の定理を得る．

定理 8 制約グラフの樹状幅が w^* であるとき，提案アルゴリズムの最悪時の解の精度は $k/(w^*+1)$ である．すなわち，得られた提携構造 CS_k^{ap} の利得 $V(CS_k^{ap})$ と最適な提携構造の利得 $V(CS^*)$ に対して，$V(CS_k^{ap})/V(CS^*) \geq k/(w^*+1)$ が成り立つ．

制約グラフの樹上幅は，エージェント間の関係の局所性を表している．例えば，樹上幅が小さいときにはエージェント間の関係が局所的であり，意味のある複数エージェント提携の数が制限される．そのため，探索空間を削減しても比較的良い提携構造を求めることができる．w^* を計算する問題は NP 困難であるが，w^* の上限は簡単に得ることができ，それを用いて解の精度を保証できる．

さらに，w^* は分散制約最適化問題を解くアルゴリズムの計算量を特徴付ける値であり，次の定理が成り立つ．

定理 9 制約グラフの樹状幅が w^* であるとき，計算量が $O(n \cdot ((w^*+1)d+1)^{w^*})$ である，最適解を得るための集中型の（分散でない）アルゴリズムを作成できる．

また，DPOP といった樹状幅を利用したアルゴリズムを用いて，計算量を削減した分散アルゴリズムを作成できる．

4. 実験

この章では，提案したアルゴリズムの実際の解の性能を実験により評価する．以下の設定で実験を行った．エージェント数 n を $10, 20, \ldots, 50$ に設定し，変数の値の領域の大きさ d を 2 とした．また，制約グラフの構造を木に限定した，つまり $w^* = 1$ に限定した．このときの理論的な最悪時の解の精度は，$1/(w^*+1) = 0.5$ である．この場合は，最適解を多項式時間で得られ，エージェント数の大きな問題でも最適解を得ることができる．単項制約および二項制約の値は [-10,10] の範囲からランダムに設定した．

この条件で，エージェント数ごとに 100 個の問題を作成し，最適解と $k=1$ の近似アルゴリズムの解とを比較したところ，近似率の平均は約 98%から 96%に単調に減少していることを観察した．これらの値は論理的な保証値である 50%に比べると大幅に良い値となっている．

5. おわりに

本論文では，特性関数の値が分散制約最適化問題の最適解で与えられる，新しい提携構造形成問題の表現方法を提案した．既存の特性関数の簡略表記手法である MC-nets や SCG と比較し，分散制約最適化問題による表記はこれらの簡略表記法よりも特性関数を簡略に表記できることを示した．さらに，全体提携と同程度の計算量で，最悪時の解の精度を理論的に保証できる近似アルゴリズムを提案した．このアルゴリズムの解の利得と最適解の比率の最悪値は，$\max(k/(w^*+1), k/\lfloor n/2 \rfloor)$ で与えられる．さらに，実験により，実際の近似率は理論的な最悪時の精度に比べて大幅に良くなることを確かめた．これらの結果は，分散制約最適化問題による特性関数の表現がエージェント間の関係の局所性を利用して効率的なアルゴリズムを実現できる，優れた定式化手法であることを示している．

今後の研究課題として，問題の構造を利用したより効率的なアルゴリズムを提案することや，外部性が存在する場合を扱えるように提案手法を拡張すること，実際の問題への応用例を提案すること等が挙げられる．

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)

[Dechter 03] Dechter, R.: *Constraint Processing*, Morgan Kaufmann (2003)

[Diestel 05] Diestel, R.: *Graph Theory*, Springer (2005), 邦訳：根上生也, 太田克弘訳,『グラフ理論』, シュプリンガー・ジャパン (2000)

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

[Modi 03] Modi, P. J., Shen, W.-M., Tambe, M., and Yokoo, M.: An Asynchronous Complete Method for Distributed Constraint Optimization, in *AAMAS*, pp. 161–168 (2003)

[Petcu 05] Petcu, A. and Faltings, B.: A Scalable Method for Multiagent Constraint Optimization, in *IJCAI*, pp. 266–271 (2005)

[Rahwan 08a] Rahwan, T. and Jennings, N. R.: Coalition structure generation: dynamic programming meets anytime optimisation, in *AAAI*, pp. 156–161 (2008)

[Rahwan 08b] Rahwan, T. and Jennings, N. R.: An improved dynamic programming algorithm for coalition structure generation, in *AAMAS*, pp. 1417–1420 (2008)

[Sandholm 99] Sandholm, T., Larson, K., Andersson, M., Shehory, O., and Tohmé, F.: Coalition structure generation with worst case guarantees, *Artificial Intelligence*, Vol. 111, No. 1-2, pp. 209–238 (1999)

[Ueda 10] Ueda, S., Iwasaki, A., Yokoo, M., Silaghi, M. C., Hirayama, K., and Matsui, T.: Coalition Structure Generation based on Distributed Constraint Optimization, in *AAAI* (2010), to appear

[岡田 96] 岡田 章：ゲーム理論, 有斐閣 (1996)