MC-netsを用いた提携構造形成問題のMaxSAT符号化の改良

越村 三幸　廖 暁鵑　野本 一貴　上田 俊　櫻井 祐子　横尾 真

提携構造形成問題は協調ゲーム理論の問題の一つで、エージェント集合を社会的効用が最大となるよう分割する問題である．MC-nets は、問題を簡潔に表現する手法の一つであり、ルールの集合で問題を表現する．我々は以前、MC-nets で表現された問題の MaxSAT 符号化による解法を示した．符号化された問題は、MaxSAT ソルバーによって解かれる．この符号化では、MC-nets のルール間関係の推移律を表す制約数がルール数が増加するにつれて膨大になる問題があった．本発表では、この制約数の削減手法を提案する．また、本手法でどの程度、削減されるのか見積もりを示す．

Coalition Structure Generation (CSG), a main research issue in the domain of coalition games, involves partitioning agents into exhaustive and disjoint coalitions so that the social welfare is optimized. MC-net is one of the compact representation schemes for representing CSG. An MC-net consists of a set of rules. We developed a method which encodes CSG with MC-net into Boolean logic and utilize an off-the-shelf MaxSAT solver as an optimization tool for solving the CSG problem. The method has a problem that the number of constraints, which represents a transitive relation over rules, becomes enormous as rules increases. This paper proposes a new method to decrease the number of constraints. We also show an estimate of its reduction rate.

1　はじめに

提携構造形成問題（CSG:Coalition Structure Generation）は協力ゲーム理論の問題の一つで、提携のもたらす効用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．提携とは分割された個々の集合のことである．従来の研究では、各提携のもたらす効用は特性関数あるいは分割関数と呼ばれるブラックボックスの関数（オラクル）によって与えられていた．特性関数は提携の効用が他の提携によらずそれ自身によって決まる場合に用いられ、分割関数は他の提携の影響を受ける場合に用いられる．特性関数と分割関数の記述量はそれぞれ $\Theta(2^n)$ と $O(n^n)$（n はエージェント数）であり、いずれも現実の問題を記述するのはエージェントが増えるに従い困難になる．

この記述困難性に対処するために、特性関数や分割関数の簡潔記述法の提案がマルチエージェントシステムの研究者らを中心に行われている．本研究で扱う MC-nets(Marginal Contribution Nets) [1] は、特性関数の簡潔記述法の一つで、ルールの集合で提携の効用を表現する．Liao らは、MC-nets で表現された CSG を MaxSAT 符号化して解く方法を二つ提案している [3] [5] [4]．一つは rule-based、もう一つは agent-based と呼ばれものである．本稿では、前者の MaxSAT 符号化の制約数を削減する手法を提案する．

MaxSAT 符号化した問題は、MaxSAT ソルバーによって解かれる．ここで、MaxSAT（Maximum Satisfiability）は、SAT（Boolean satisfiability）を最適化問題に拡張したものである [2] [6]．

An Improvement of MaxSAT Encoding for Coalition Structure Generation Using MC-nets.

Miyuki Koshimura, Kazuki Nomoto, Makoto Yokoo, 九州大学, Kyushu University.

Xiaojuan Liao, 中国・西南科技大学, Southwest University of Science and Technology (China).

Suguru Ueda, 佐賀大学, Saga University.

Yuko Sakurai, 産業技術総合研究所, National Institute of Advanced Industrial Science and Technology.

本稿の構成は以下の通りである．まず、2 節にて提携構造形成問題を、3 節にて MC-nets を紹介する．次に、4.1 節にて従来の MaxSAT 符号化を説明し、4.2 節にてその改良手法を提案する．最後に 5 節にてまとめと今後の課題を述べる．

2 提携構造形成問題

エージェントの全体集合を $A = \{a_1, \ldots, a_n\}$ とする．エージェントの部分集合 $C \subseteq A$ を提携と呼ぶ．提携は空でない（$C \neq \emptyset$）とする．A を提携に分割することで得られる提携の組み合わせを提携構造と呼ぶ．つまり、提携構造 CS はエージェトの部分集合の集合 $CS = \{C_1, \ldots, C_k \mid C_i \subseteq A\}$ で $C_i \cap C_j = \emptyset\ (i \neq j)$ と $\bigcup_{C_i \in CS} C_i = A$ を満たす．

特性関数 v は、提携 C の効用を与えるもので、C から実数への関数として与えらえる（$v : 2^A \to \mathbb{R}$）．提携構造 CS の効用 $V(CS)$ は、CS に含まれる提携の効用の総和である（$V(CS) = \sum_{C_i \in CS} v(C_i)$）．提携構造形成問題は、最大の効用をもたらす提携構造 CS^* を見つける問題である．

3 MC-nets

Ieong と Shoham は、特性関数を簡潔に記述する手法として、MC-nets を提案した．これは、特性関数をルールの集合によって簡略に記述する手法である．

定義 1 (MC-nets) MC-nets は提携が満たすべきルールの集合 R によって記述される．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 と N_r はそれぞれ「存在しなければならない」及び「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、r が満たされた場合の利得を表す．提携 C について $P_r \subseteq C$ かつ $N_r \cap C = \emptyset$ が成り立つとき、r は C に適用可能であるという．C に適用可能なルールの全体集合を R_C とする時、C の効用は $v(C) = \sum_{r \in R_C} v_r$ で与えられる．

例 1 $A = \{a_1, a_2, a_3, a_4\}$ として、次の 4 つのルールが与えられているものとする．

$r_1 : (\{a_1, a_2\}, \emptyset) \to 2 \quad r_2 : (\{a_1, a_2\}, \{a_4\}) \to -2$
$r_3 : (\{a_1, a_4\}, \emptyset) \to 1 \quad r_4 : (\{a_3\}, \{a_2\}) \to 3$

この時、r_1 と r_3 は提携 $\{a_1, a_2, a_4\}$ に適用可能であるが、r_2 と r_4 は適用可能でない．こうして、$v(\{a_1, a_2, a_4\}) = 2 + 1 = 3$ となる．すべての提携構造を列挙すると、最適な提携構造 CS^* は $\{\{a_1, a_2, a_4\}, \{a_3\}\}$ であることが分かり、$V(CS^*) = V(\{\{a_1, a_2, a_4\}, \{a_3\}\}) = 2 + 1 + 3 = 6$ となる．

Ohta らは、MC-nets によって特性関数が記述されている場合に、提携構造形成問題を混合整数計画法（mixed integer programming）を利用して解く手法を提案している [7]．この手法はルールの利得はすべて正であると仮定しているが、後に、Ueda らにより負の利得にも対処できるように拡張された [10] [8]．

これらは、MC-nets のルール間の関係に着目して、提携構造形成問題を解いている．4 節で述べる MaxSAT 符号化もこの形式化に基づく．これはルール間の関係に着目した符号化なので rule-based と呼ばれる．これに関して、ここでは 4 節の議論に必要な定義と定理を述べる．

定義 2 (実現可能なルール集合) ルール集合 $R' (\subseteq R)$ について、任意のルール $r \in R'$ が必ずいずれか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．

提携構造形成問題は、ルールの利得の総和 $\sum_{r \in R'} v_r$ を最大化する実現可能なルール集合 R' を探索する問題に等しい．

定義 3 (ルール間の関係) 任意の 2 つのルール r と r' の間の関係を次の 4 つに分類する．

- **同提携で両立可能** $P_r \cap P_{r'} \neq \emptyset \ \wedge \ P_r \cap N_{r'} = P_{r'} \cap N_r = \emptyset$
- **両立不可能** $P_r \cap P_{r'} \neq \emptyset \ \wedge \ (P_r \cap N_{r'} \neq \emptyset \vee P_{r'} \cap N_r \neq \emptyset)$
- **他提携で両立可能** $P_r \cap P_{r'} = \emptyset \ \wedge \ (P_r \cap N_{r'} \neq \emptyset \vee P_{r'} \cap N_r \neq \emptyset)$
- **独立** $P_r \cap P_{r'} = \emptyset \ \wedge \ P_r \cap N_{r'} = P_{r'} \cap N_r = \emptyset$

MC-nets のルール集合を、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することができる．次の定理は、ルール集合が実現可能であるための必要十分条件を示す．

定理 1 ルール集合 R' は、以下の条件を同時に満た

すとき、かつそのときに限り実現可能である.

1. R' は、「両立不可能」なエッジで連結されたルール対を含まない.
2. R' に含まれる任意のルール対が「他提携で両立可能」なエッジで連結されているとき、このルール対は「同提携で両立可能」なエッジによって到達不可能である.

4 MaxSAT 符号化

MaxSAT は、SAT の最適化版で、SAT では解に優劣はないが、MaxSAT ではあり、最適解が MaxSAT 解となる.

MaxSAT では、問題はハード節とソフト節の集合として表される. ソフト節には重みがあり、重みは正整数で表される. 重みはそのソフト節の重要性の度合いを示しており、ハード節はどのソフト節よりも重要で必ず満たさなければならない制約を表す.

MaxSAT の目的は、変数の値割当の中で、全てのハード節を満たし、かつ満たされるソフト節の重みの総和が最大となるものを見つけることである.

ここで、節（clause）はリテラル（literal）の選言（論理和）、リテラルはブール変数あるいはその否定である. 節中、否定記号を伴ったリテラルを負リテラル、そうでないリテラルを正リテラルと呼ぶ. 本稿では節 C の重さが w の時、それらの組 (C, w) でソフト節を表す.

以下、本稿では、ルールの効用は正値であるとして話を進めるが、負値に拡張するのは容易である.

$R = \{r_1, r_2, \ldots, r_n\}$ を MC-nets のルール（その効用は正）の集合とする. 各ルール r_i ごとに一つのブール変数 B_i を導入する. 直感的には、$B_i = 1$ は r_i が実現可能なルール集合の要素であることを意味する.

定理 1 を利用するために、到達可能性を表すブール変数 $S(i, j)$ をルールの対 (r_i, r_j) ごとに用意する（$i < j$）. 直感的には、$S(i, j) = 1$ は r_i と r_j が共に実現可能なルール集合の要素であり、かつ、「同提携で両立可能」なエッジによって到達可能である、ことを意味する.

4.1 従来手法

到達可能性の推移律を表す次のようなハード節をルールの 3 つ組 (r_i, r_j, r_k) ごとに用意する（$1 \le i < j < k \le n$）.

1. $\neg S(i, j) \vee \neg S(j, k) \vee S(i, k)$
2. $\neg S(i, j) \vee \neg S(i, k) \vee S(j, k)$
3. $\neg S(i, k) \vee \neg S(j, k) \vee S(i, j)$

このような推移律のためのハード節は全部で ${}_nC_3 \cdot 3 = n \cdot (n-1) \cdot (n-2)/2$ 個必要となる.

定義 4 (従来の MaxSAT 符号化) R を MC-nets のルール（その効用は正）の集合とする. 各ルール $(r_i : (P_i, N_i) \to w_i) \in R$ に対し、ソフト節 (B_i, w_i) を導入する. また、R の各ルール対 (r_i, r_j) に対して、ルール間の関係に基づいて次のようにハード節を導入する ($i < j$).

同提携で両立可能 $\neg B_i \vee \neg B_j \vee S(i, j)$, $\neg S(i, j) \vee B_i$, $\neg S(i, j) \vee B_j$.

両立不可能 $\neg B_i \vee \neg B_j$.

他提携で両立可能 $\neg B_i \vee \neg B_j \vee \neg S(i, j)$.

独立 ハード節は導入しない.

4.2 提案手法

従来手法では、推移律を表すハード節の数は $O(n^3)$ のオーダーで増加する. 他のハード節の数のオーダーは $O(n^2)$ であり、n が増えるに従い、推移律を表すハード節の数が全体の節の大半を占めるようになる. 例えば、ルール数が 100 なら 485,100 個、200 なら 3,940,200 個のハード節が推移律のために必要となる. ハード節の増加は、MaxSAT ソルバーの求解速度の低下につながる.

本節では、推移律を表すハード節を削減する MaxSAT 符号化を提案する. この手法では、推移律を表すハード節で $S(i, j)$ を正リテラルとして持つものを導入するかどうかを、r_i と r_j の関係に着目して判定する.

説明を簡単にするために、次のように定義される $S(\{i, j\})$ を導入する（$i \ne j$）.

$$S(\{i, j\}) = \begin{cases} S(i, j) & i < j \text{ のとき} \\ S(j, i) & j < i \text{ のとき} \end{cases}$$

定義 5 (新 MaxSAT 符号化) R を MC-nets のルール（その効用は正）の集合とする．ソフト節については、従来手法と同じように導入する．R の各ルール対 (r_i, r_j) に対して、ルール間の関係に着目して次のようにハード節を導入する $(i < j)$.

1. 同提携で両立可能:$\neg B_i \vee \neg B_j \vee S(i,j)$, $\neg S(i,j) \vee B_i$, $\neg S(i,j) \vee B_j$.
2. 両立不可能：$\neg B_i \vee \neg B_j$, $\neg S(i,j)$.
3. 上記以外で、r_i と同提携で両立可能なルールがない場合：$\neg S(i,j)$.
4. 同様に上記以外で、r_j と同提携で両立可能なルールがない場合：$\neg S(i,j)$.
5. 上記以外の場合 †1：r_j と同提携で両立可能な全てのルール r_k に対して、ハード節 $\neg S(\{i,k\}) \vee \neg S(\{k,j\}) \vee S(i,j)$ を導入する．加えて、r_i と r_j が他提携で両立可能であるとき、ハード節 $\neg B_i \vee \neg B_j \vee \neg S(i,j)$ を導入する．

提案手法で推移律を表すハード節が導入されるのは、定義 5 の case 5 の場合のみである．さらに、r_j と同提携で両立可能なルール r_k に限定したハード節のみ作成するので相当数のハード節の削減が期待できる．

例 2 $A = \{a_1, a_2, a_3, a_4\}$ で、MC-nets ルールが次のように 4 個あるとする．

$r_1 : (\{a_1, a_2\}, \emptyset) \to 2 \quad r_2 : (\{a_1, a_4\}, \{a_2\}) \to 2$
$r_3 : (\{a_2, a_4\}, \emptyset) \to 1 \quad r_4 : (\{a_3\}, \{a_2\}) \to 3$

このとき、ルール間の関係は次のようになる．

- r_1 と r_2： 両立不可能
- r_1 と r_3： 同提携で両立可能
- r_1 と r_4： 他提携で両立可能
- r_2 と r_3： 両立不可能
- r_2 と r_4： 独立
- r_3 と r_4： 他提携で両立可能

ルールが 4 個あるので 4 個のソフト節が導入される：$(B_1, 2)$, $(B_2, 2)$, $(B_3, 1)$, $(B_4, 3)$.

4 個のルールの 6 通りの対から、次のようにハード節が導入される．ここで、r_2 と r_4 には同提携で両立可能なルールがないことに注意されたい．

- r_1 と r_2 （定義 5 の case 2 を適用）：
  $\neg B_1 \vee \neg B_2$, $\neg S(1,2)$.
- r_1 と r_3 （定義 5 の case 1 を適用）：
  $\neg B_1 \vee \neg B_3 \vee S(1,3)$, $\neg S(1,3) \vee B_1$, $\neg S(1,3) \vee B_3$.
- r_1 と r_4 （定義 5 の case 4 を適用）：
  $\neg S(1,4)$.
- r_2 と r_3 （定義 5 の case 2 を適用）：
  $\neg B_2 \vee \neg B_3$, $\neg S(2,3)$.
- r_2 と r_4 （定義 5 の case 3 を適用）：
  $\neg S(2,4)$.
- r_3 と r_4 （定義 5 の case 4 を適用）：
  $\neg S(3,4)$.

この例では、定義 5 の case 5 は適用されないので、推移律を表すハード節は作成されない．

例 3 $A = \{a_1, a_2, a_3, a_4\}$ で、MC-nets ルールが次のように 4 個あるとする．

$r_1 : (\{a_1\}, \{a_2\}) \to 2 \quad r_2 : (\{a_2, a_3\}, \emptyset) \to 2$
$r_3 : (\{a_1, a_4\}, \emptyset) \to 1 \quad r_4 : (\{a_3\}, \emptyset) \to 3$

このとき、ルール間の関係は次のようになる．

- r_1 と r_2： 他提携で両立可能
- r_1 と r_3： 同提携で両立可能
- r_1 と r_4： 独立
- r_2 と r_3： 独立
- r_2 と r_4： 同提携で両立可能
- r_3 と r_4： 独立

4 個のソフト節が導入される:$(B_1, 2)$, $(B_2, 2)$, $(B_3, 1)$, $(B_4, 3)$.

4 個のルールの 6 通りの対から、次のようにハード節が導入される．

- r_1 と r_2 （定義 5 の case 5 を適用）：
  $\neg S(1,4) \vee \neg S(2,4) \vee S(1,2)$, $\neg B_1 \vee \neg B_2 \vee \neg S(1,2)$.
- r_1 と r_3 （定義 5 の case 1 を適用）：
  $\neg B_1 \vee \neg B_3 \vee S(1,3)$, $\neg S(1,3) \vee B_1$, $\neg S(1,3) \vee B_3$.
- r_1 と r_4 （定義 5 の case 5 を適用）：
  $\neg S(1,2) \vee \neg S(2,4) \vee S(1,4)$.
- r_2 と r_3 （定義 5 の case 5 を適用）：
  $\neg S(1,2) \vee \neg S(1,3) \vee S(2,3)$.

†1 この場合、r_i と r_j は「他提携で両立可能」か「独立」のいずれかである．また、r_j と同提携で両立可能なルールが少なくとも一つはある．

- r_2 と r_4 （定義 5 の case 1 を適用）：
  $\neg B_2 \vee \neg B_4 \vee S(2,4)$, $\neg S(2,4) \vee B_2$, $\neg S(2,4) \vee B_4$.
- r_3 と r_4 （定義 5 の case 5 を適用）：
  $\neg S(2,3) \vee \neg S(2,4) \vee S(3,4)$.

r_1 と r_2 の対からは、推移律を表すハード節は 1 個だけ作成される．これは、r_2 と同提携で両立可能なルールが r_4 の 1 個だけだからである．同様に、r_1 と r_4、r_2 と r_3、r_3 と r_4 の対から作成される推移律を表すハード節もそれぞれ 1 個である．こうして、この例で作成される推移律を表すハード節は、全部で 4 個である．従来手法だと、作成されるハード節は $12(= (4 \cdot 3 \cdot 2)/2)$ 個なので、8 個削減されることが分かる．

定義 5 によって導入されるハード節の数を見積もるため、次のようにルール間の関係の確率を考える．

a： 同提携で両立可能となる確率
b： 両立不可能となる確率
c： 他提携で両立可能となる確率
d： 独立となる確率

確率の定義より $a+b+c+d=1$ である．

このとき、定義 5 のそれぞれの場合で導入されるハード節の数は次のように見積もることができる．

1. $a \cdot {}_nC_2 \cdot 3$
2. $b \cdot {}_nC_2 \cdot 2$
3. $(1-a-b) \cdot {}_nC_2 \cdot (1-a)^{n-2}$
4. $(1-a-b) \cdot {}_nC_2 \cdot (1-(1-a)^{n-2}) \cdot (1-a)^{n-2}$
   3 と 4 は、n が十分大きければ滅多に起こらないと考えられるので、以降これらの場合を無視する．
5. $(c+d) \cdot {}_nC_2 \cdot (n-2) \cdot a$. これは推移律を表すハード節の数で、他提携で両立可能な場合は、これに加えて、$c \cdot {}_nC_2$ 個のハード節が加わる．

従来手法では、推移律を表すハード節は $n \cdot (n-1) \cdot (n-2)/2$ 個であったから、提案手法による減少率は、

$$(c+d) \cdot {}_nC_2 \cdot (n-2) \cdot a \div n \cdot (n-1) \cdot (n-2)/2 = n \cdot (n-1) \cdot (c+d) \cdot a/2 \div n \cdot (n-1)/2 = (c+d) \cdot a$$

となる．$a + (c+d) \leq 1$ なので、$(c+d) \cdot a \leq 1/4$. つまり、提案手法によって、推移律を表すハード節の数は確率的には、少なくとも 1/4 には減らすことができる．

5 おわりに

本稿では、MC-net を用いた提携構造形成問題の MaxSAT 符号化の節数の削減手法を提案した．削減されるのは、MC-nets のルール間の推移律を表す節で、確率的には、従来手法の少なくとも 1/4 に削減できることを示した．1/4 になるのは、$a = 1/2$ かつ $c + d = 1/2$ のときであり、実際にこの値になることは稀であると考えられるので、現実の問題ではより多く削減される、と予想される．我々の予備的実験では、MC-nets のルール数が 150（効用は負の場合もあり）のあるインスタンスでは、節の数が約 400 万から十数万に削減され、MaxSAT ソルバーの計算時間も 10 秒から 1 秒に短縮できた．

今後、従来手法との詳細な定量的な比較を行う予定である．また、混合整数計画法を利用する解法との比較も行いたいと考えている．さらに、分割決定木[9]を用いた提携構造形成問題の解法[11][12] にも提案手法を適用していきたい．

謝辞 本研究は、JSPS 科研費 JP16K00304、JP17K00307 の助成を受けたものです．

参考文献

[1] Ieong, S. and Shoham, Y.: Marginal contribution nets: a compact representation scheme for coalitional games, *Proceedings of the 6th ACM Conference on Electronic Commerce (ACM EC)*, 2005, pp. 193–202.

[2] Li, C. M. and Manyà, F.: MaxSAT, Hard and Soft Constraints, *Handbook of Satisfiability*, Biere, A., Heule, M., van Maaren, H., and Walsh, T.(eds.), Frontiers in Artificial Intelligence and Applications, Vol. 185, IOS Press, 2009, pp. 613–631.

[3] Liao, X., Koshimura, M., Fujita, H., and Hasegawa, R.: Solving the Coalition Structure Generation Problem with MaxSAT, *IEEE 24th International Conference on Tools with Artificial Intelligence, ICTAI 2012, Athens, Greece, November 7-9, 2012*, IEEE Computer Society, 2012, pp. 910–915.

[4] Liao, X., Koshimura, M., Fujita, H., and Hasegawa, R.: Extending MaxSAT to Solve the Coalition Structure Generation Problem with Externalities Based on Agent Relations, *IEICE Trans-*

actions, Vol. 97-D, No. 7(2014), pp. 1812–1821.

[5] Liao, X., Koshimura, M., Fujita, H., and Hasegawa, R.: MaxSAT Encoding for MC-Net-Based Coalition Structure Generation Problem with Externalities, *IEICE Transactions*, Vol. 97-D, No. 7(2014), pp. 1781–1789.

[6] 越村三幸, 藤田博: SAT 技術の進化と応用 〜パズルからプログラム検証まで〜：6. MaxSAT：SAT の最適化問題への拡張 -MaxSAT ソルバーの活用法-, 情報処理, Vol. 57, No. 8(2016), pp. 730–733.

[7] Ohta, N., Conitzer, V., Ichimura, R., Sakurai, Y., Iwasaki, A., and Yokoo, M.: Coalition Structure Generation Utilizing Compact Characteristic Function Representations, *Proceedings of the 15th International Conference on Principles and Practice of Constraint Programming (CP)*, 2009, pp. 623–638.

[8] 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,* 情報・システム, Vol. 94, No. 11(2011), pp. 1707–1715.

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

[10] Ueda, S., Hasegawa, T., Hashimoto, N., Ohta, N., Iwasaki, A., and Yokoo, M.: Handling negative value rules in MC-net-based coalition structure generation, *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2012, pp. 795–804.

[11] 野本一貴, 大田一徳, 上田俊, 櫻井祐子, 横尾真: 分割関数ゲームの提携構造形成アルゴリズム, *JAWS 2016*, 9 月 2016.

[12] 越村三幸, 査澳龍, 野本一貴, 櫻井祐子, 横尾真: 分割関数ゲームを対象とした提携構造形成問題の MaxSAT 符号化, *2017* 年度人工知能学会全国大会 論文集, 5 月 2017.