1M3-1

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

A MaxSAT Encoding of Coalition Structure Generation for Partition Function Games

越村 三幸 *1 Miyuki KOSHIMURA
査 澳龍 *2 Aolong ZHA
野本 一貴 *2 Kazuki NOMOTO
櫻井 祐子 *1 Yuko SAKURAI
横尾 真 *1 Makoto YOKOO

*1九州大学 大学院システム情報科学研究院
Faculty of Information Science and Electrical Engineering, Kyushu University

*2九州大学 大学院システム情報科学府
Graduate School of Information Science and Electrical Engineering, Kyushu University

This paper proposes a MaxSAT encoding of Coalition Structure Generation (CSG) problems for Partition Function Games (PFGs) which are coalition games with externalities. In this work, the CSG problems are represented with Partition Decision Trees (PDTs) which are graphical representations for PFGs. MaxSAT is an optimization version of SAT which consists in finding an assignment that maximizes the number of satisfied clauses. It is used for solving combinatorial optimization problems. Experimental results show that we obtain good and stable performance especially when there exist both positive and negative values for coalitions.

1. はじめに

提携構造形成問題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．ここで、分割された個々の集合が提携と呼ばれる．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）によって与えられる．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．

本研究で扱う分割決定木 [Skibski 15] は、分割関数の簡潔記述法の一つで、複数の多分木で提携の効用を表現する．野本らは、分割決定木で表現された提携構造形成問題を解くアルゴリズムを複数提案している [野本 16b, 野本 16a]．これらは、提携の効用が正の値のみである場合は比較的簡単な分枝限定アルゴリズムであるが、負の値の効用をもつ提携が存在する場合、手続きが複雑になる．

本稿では、MaxSAT 符号化による解法を提案する．これは、MC-nets[Ieong 05] で記述された提携構造形成問題の MaxSAT 符号化 [Liao 12] と同じアイディアに基づく．ここで、MC-nets は特性関数の簡潔記述法の一つである．MaxSAT 符号化した問題は、MaxSAT ソルバーによって解かれる．MaxSAT（Maximum Satisfiability）は、SAT（Boolean satisfiability）の最適化版である [越村 16]．提案する MaxSAT 符号化では、正の値の取り扱い、負の値の取り扱い、いずれも簡潔である．これが、前述のアルゴリズムを使った解法と比べた時の MaxSAT を利用する優位な点である．

本稿では、MaxSAT 符号化を形式的に導入するために、先ず分割決定木について形式的に説明する（3. 節）．これは、文献 [Skibski 15] の該当する部分をほぼなぞっている．MaxSAT 符号化の形式的ではない説明については、文献 [越村 17] を参照されたい．

連絡先: 越村三幸，九州大学大学院システム情報科学研究院，福岡市西区元岡 744，092-802-3599，koshi@inf.kyushu-u.ac.jp

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 $\bigcup_{C_i \in CS} C_i = CS$ を満たす．

分割関数 w は提携 C の効用を与えるもので C と（C を含む）提携構造 CS から実数 $\mathbb{R}$ への関数として表される（$w\colon (C, CS) \rightarrow \mathbb{R}$）．提携構造 CS の効用 $W(CS)$ は、CS に含まれる提携の効用の総和である（$W(CS) = \sum_{C_i \in CS} w(C_i, CS)$）．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. 分割決定木（Partition Decision Trees）

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表現する手法である．分割決定木によって、任意の分割ゲームが表現できる．

3.1 PDT ルール

分割決定木では、PDT ルールと呼ばれる多分木の集合で分割関数ゲームを表現する．PDT ルールにおいて内部ノード（葉でないノード）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る．葉ノードは、根ノードからその葉ノードに至る経路上に現れるエージェントらの分割がもたらす効用を表す．

形式的には一つの PDT ルール T は、タプルを用いて次のように定義される．$T = \langle V, E, x, f_V, f_E \rangle$．ここで、$V$ はノードの集合、$E \subseteq V \times V$ は有向枝の集合、(V, E) は x を根とする有向木（directed tree）で、根ノードを除く任意のノード $z \in V$ について、z への有向枝を持つノード $y \in V$ が丁度一つ存在する．$f_V\colon V \rightarrow A \cup \mathbb{R}^{\mathbb{N}}$ はノードのラベル付け関数で、葉ノード v については $f_V(v) \in \mathbb{R}^{\mathbb{N}}$、葉でないノード v につい

ては $f_V(v) \in A$ である．$f_E\colon E \to \mathbb{N}$ はエッジのラベル付け関数である．

一つの PDT ルール T が与えられたとき、$\Pi(T)$ で T の根ノードから葉ノードに至る経路の集合を表す．一つの経路 $\pi = (v_1, v_2, \ldots, v_k) \in \Pi(T)$ は、エージェント集合の分割を表している．ここで、$f_E(v_i, v_{i+1})$ は、$f_V(v_i)$ が属する提携の番号を表している．また、$last(\pi)$ で経路 π の最後、つまり葉ノードを表すものとする．なお、$v_1 = x$ であることに注意されたい．

PDT ルールは、その任意の経路 π が次の性質を持つように作られる．

- π 上の内部ノードは、全て異なるエージェントでラベル付けされる．即ち、$|\{f_V(v_1), f_V(v_2), \ldots, f_V(v_{k-1})\}| = k-1$ が成立する．
- 根 x（つまり v_1）から v_i（$i > 0$）に至る任意の経路について、そのエッジのラベル集合は、1 から始まる連続した自然数の集合となる．つまり、$f_E(v_1, v_2) = 1$ であり、$1 \leq i < k$ なる i について、$f_E(v_i, v_{i+1}) \leq max_{1 \leq j < i} f_E(v_j, v_{j+1}) + 1$ が成立する．
- 葉ノード v_k のラベル $f_V(v_k)$ のサイズは、π が表す分割に含まれる提携数に等しい．即ち、$|f_V(v_k)| = max_{1 \leq j < k} f_E(v_j, v_{j+1})$ が成立する．

また、$\Pi(T)$ 中の各経路がそれぞれ異なる分割を表すようにするため、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ようにする．したがって、v_i, v_j, v_l が $f_E(v_i, v_j) = f_E(v_i, v_l)$ を満たすなら、$v_j = v_l$ が成立する．

3.2 PDT ルールの充足性

提携構造 CS が経路 $\pi = (v_1, v_2, \ldots, v_k) \in \Pi(T)$ で表される分割を被覆（cover）するとき、CS は π を充足する、といい、$CS \sim \pi$ と表記する．したがって、経路上に現れる同一提携に含まれる二つのエージェントについて、そこから出るエッジのラベルは等しい．つまり、CS に含まれる任意の提携 C 中の任意のエージェント a と b について、$f_V(v_i) = a$ かつ $f_V(v_j) = b$ がある i と j（$1 \leq i, j < k$）について成り立つなら、$f_E(v_i, v_{i+1}) = f_E(v_j, v_{j+1})$ が成立する．また、T 中の経路はそれぞれ別の分割を表すので、CS が充足する T 中の経路は高々一つである．

3.3 提携の効用

$CS \sim \pi$ の時、CS に含まれる提携から π 上のエッジのラベル（提携番号）に 0 を加えたものへの写像 $g_\pi^{CS}\colon CS \to \{1, 2, \ldots, max_{1 \leq j < k} f_E(v_j, v_{j+1})\} \cup \{0\}$ が存在する．ここで、0 にはそのエージェントのいずれも π 上に現れない提携が割当てられる．

例 1 図 1 左の PDT ルールの左の経路 $\pi\colon a_3 \xrightarrow{1} a_4 \xrightarrow{2} a_1 \xrightarrow{2} [3,4]$ は、提携構造 $CS = \{\{a_1, a_2, a_4\}, \{a_3\}, \{a_5\}, \{a_6\}\}$ で充足され、$g_\pi^{CS}(\{a_1, a_2, a_4\}) = 2$、$g_\pi^{CS}(\{a_3\}) = 1$、$g_\pi^{CS}(\{a_5\}) = g_\pi^{CS}(\{a_6\}) = 0$ が成立する．

このように、g_π^{CS} は提携構造 CS に含まれる提携 C を入力すると、経路 π が表す分割内で対応する提携を表すラベル（提携番号）を返す．これを用いて、提携構造 CS に含まれる提携 C の効用 $w_\pi^{CS}(C)$ は次のように表される．

$$w_\pi^{CS}(C) = \begin{cases} f_V(last(\pi))[g_\pi^{CS}(C)] & g_\pi^{CS}(C) > 0 \text{ のとき} \\ 0 & \text{それ以外のとき} \end{cases}$$

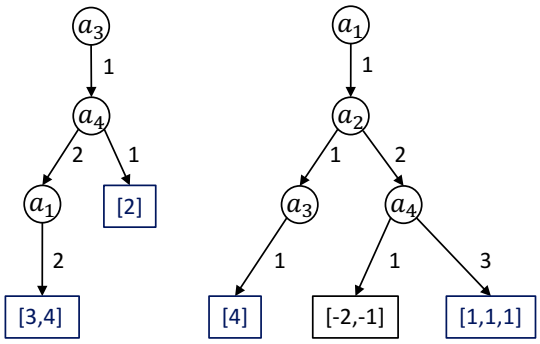

図 1: PDT ルールの例

ここで、$f_V(last(\pi))$ は経路 π の葉ノードのラベルであり、一般に実数の並びである．例 1 の経路では $[3,4]$ となる．また、$g_\pi^{CS}(C)$ は自然数であり、$f_V(last(\pi))[g_\pi^{CS}(C)]$ は実数の並び $f_V(last(\pi))$ の $g_\pi^{CS}(C)$ 番目の実数を表す．

例 2 例 1 と同じ経路 π と提携構造 CS では、次のようになる．

$$w_\pi^{CS}(\{a_1, a_2, a_4\}) = [3,4][2] = 4$$
$$w_\pi^{CS}(\{a_3\}) = [3,4][1] = 3$$
$$w_\pi^{CS}(\{a_5\}) = w_\pi^{CS}(\{a_6\}) = 0$$

次に、提携構造 CS に含まれる提携 C の PDT ルール T による効用 $w^T(C, CS)$ を次のように定める．

$$w^T(C, CS) \stackrel{\text{def}}{=} \sum_{\substack{\pi \in \Pi(T) \\ CS \sim \pi}} w_\pi^{CS}(C)$$

このとき、PDT ルールの集合 $\mathcal{T} = \{T_1, T_2, \ldots\}$ による $C \in CS$ の効用 $w^{\mathcal{T}}(C, CS)$ は次のように定義される．

$$w^{\mathcal{T}}(C, CS) \stackrel{\text{def}}{=} \sum_{T \in \mathcal{T}} w^T(C, CS) = \sum_{T \in \mathcal{T}} \sum_{\substack{\pi \in \Pi(T) \\ CS \sim \pi}} w_\pi^{CS}(C)$$

4. MaxSAT 符号化

MaxSAT（Maximum satisfiability）は、SAT（Boolean satisfiability）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．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．

符号化に先立って、ブール変数 $C(i, j)$ を用意する．これはエージェント i と j が同じ提携の属することを表す．なお、$C(i, j)$ と $C(j, i)$ は同じことを表すので、エージェント間に適当に全順序 $\prec$ を導入して $C(i, j)(i \prec j)$ とする．ゲームに現れ

るエージェントが a_1 から a_4 の四つなら、$C(a_1,a_2)$, $C(a_1,a_3)$, $C(a_1,a_4)$, $C(a_2,a_3)$, $C(a_2,a_4)$, $C(a_3,a_4)$ の 6 個のブール変数を用意する．

符号化では、PDT ルールの葉一つに対してソフト節一つを作る．したがって、図 1 の左の PDT ルールからは二つ、右の PDT ルールからは三つのソフト節が作られる．ソフト節の作成に先立って、PDT ルールの各経路

$$\pi\colon x_1 \xrightarrow{n_1} x_2 \xrightarrow{n_2} \dots x_{k-1} \xrightarrow{n_{k-1}} [r_1, r_2, \dots, r_m]$$

からリテラルの集合 Ls^π を作る．以下で、$first(i)$ は π 中に現れる最初の提携番号 i にラベル付けされたエッジがでるノードにラベル付けされたエージェントを表すものとする $(1 \le i \le m)$．即ち、$first(i)$ は $n_j = i$ を満たす最小の j に対応する x_j である．

例 3 図 1 左の PDT ルールの左の経路では、$first(1) = a_3$、$first(2) = a_4$ である．右の PDT ルールの最右経路では、$first(1) = a_1$、$first(2) = a_2$、$first(3) = a_4$ である．

Ls^π は $1 < i < k$ なる i について次の (1)(2) を満たす最小の集合である．

(1) $first(n_i) = x_i$ のとき、
$1 \le j < n_i$ を満たす全ての j について、
$first(j) \prec x_i$ なら $\neg C(first(j), x_i) \in Ls^\pi$、
$first(j) \succ x_i$ なら $\neg C(x_i, first(j)) \in Ls^\pi$．

(2) $first(n_i) \neq x_i$ のとき、
$first(n_i) \prec x_i$ なら $C(first(n_i), x_i) \in Ls^\pi$、
$first(n_i) \succ x_i$ なら $C(x_i, first(n_i)) \in Ls^\pi$．

例 4 図 1 左の PDT ルールの左の経路 π_1 では、$Ls^{\pi_1} = \{\neg C(a_3,a_4), C(a_1,a_4)\}$．右の PDT ルールの最右経路 π_2 では、$Ls^{\pi_2} = \{\neg C(a_1,a_2), \neg C(a_1,a_4), \neg C(a_2,a_4)\}$．

経路 π に対応するソフト節とハード節は、$Ls^\pi = \{L_1, L_2, \dots, L_l\}$ を用いて次のように作られる．なお、$last(\pi)(= [r_1, r_2, \dots, r_m])$ の要素の総和 $R = \sum_{i=1}^{m} r_i$ の正負によって扱いが異なる．

1. $R > 0$ のとき、新しいブール変数 b_π を用意し、ソフト節 (b_π, R) を作る．これに加え、l 個のハード節 $\neg b_\pi \vee L_i (1 \le i \le l)$ を作る．
2. $R < 0$ のとき、ソフト節 $(\neg L_1 \vee \neg L_2 \vee \dots \vee \neg L_l, -R)$ を作る．
3. $R = 0$ のとき、ソフト節、ハード節共に作らない．

例 5 図 1 左の PDT ルールの左の経路からは、新変数 b を用意してソフト節 $(b, 7)$、二つのハード節 $\neg b \vee \neg C(a_3,a_4)$ と $\neg b \vee C(a_1,a_4)$ が作られる．右の PDT ルールの真ん中の経路からは、ソフト節 $(C(a_1,a_2) \vee \neg C(a_1,a_4), 3)$ が作られる．

以上の他に、提携関係は推移律を満たすので、次のようなハード節を加える．

$$\neg C(a_1,a_2) \vee \neg C(a_2,a_3) \vee C(a_1,a_3)$$

推移律を表すハード節の数は、エージェント数が n のとき $\dfrac{n(n-1)(n-2)}{2}$ となる．

5. 計算機実験

先行研究 [野本 16a] と同じ問題を用いて MaxSAT 符号化の評価を行った．実験は、Intel Xeon E5 Quad-Core 3.7GHz プロセッサで 16GB メモリを搭載したマシンで行った．OS は Mac OS X 10.10.5 である．

5.1 分割決定木の作成

分割決定木は以下のように作成した．詳細は先行研究を参照されたい．

PDT ルールの作成法：先ず、PDT ルール i に含まれるエージェント集合 A_i と葉の数 l を決める [*1]．これと 3.1 節で述べたルールの定義に合うように、ルール（多分木）を葉のラベルを除いてランダムに作る．

効用の決定法：確率 q で分割の効用を負とする．提携 C の効用を二つの分布関数から決定する．

1. 一様分布：正値であれば、$[0, |C|]$ の範囲から、負値であれば、$[-|C|, 0]$ の範囲から一様ランダムに選択する．
2. NDCS(Normally Distributed Coalition Structure)：正値であれば、平均 $|C|$、標準偏差 $\sqrt{|C|}$ の分布で選択する．負値であれば、平均 $|C|$、標準偏差 $\sqrt{|C|}$ の分布で選択し負に反転する．

MaxSAT では、整数しか扱えないため、上記のように定めた値を 10,000 倍して、小数点以下を切り捨てた値を用いた．

実験 1：エージェント数を 100、PDT ルールの数を 8 とした．各 PDT ルールの葉ノード数は、平均が 20 となるように $[15, 25]$ の範囲で決定した．また、各ルールに含まれるエージェント数を $[7, 13]$ の範囲から一様ランダムで決定した．分割の効用が負の確率となる q は、0、0.05、0.1、0.15、0.2 と変化させ、提携の効用は一様分布あるいは NDCS で決定した．

実験 2：エージェント数を 100、PDT ルールの数を 8 とした．各 PDT ルールの葉ノード数は、平均が 20、25、30、35、40 となるように、それぞれ、$[15, 25]$、$[20, 30]$、$[25, 35]$、$[30, 40]$、$[35, 45]$ の範囲で決定した．また、各ルールに含まれるエージェント数を $[7, 13]$ の範囲から一様ランダムで決定した．分割の効用が負の確率となる q は 0.05 とし、提携の効用は一様分布で決定した．

5.2 実験結果

それぞれの問題設定ごとに 100 インスタンスを生成した．MaxSAT ソルバーは QMaxSAT[Koshimura 12] の Ver.201702 を使用した．また、比較のため先行研究 [野本 16a] で最も性能の良かった Algorithm for a modified PDT（以下、MPDT と略す）でも解いた．MPDT は分枝限定法に基づく解法であり、PDT ルールの探索順が効率に大きく影響する．今回は、先行研究で最も性能の良かった探索順を用いた．

表 1 に実験 1、表 2 に実験 2 の結果を示す．平均実行時間とは、それぞれ 100 インスタンスを解くのに要した時間の平均である．

表より、負の効用がない問題（$q = 0$）については、MPDT が優位であるが、負の効用がある問題（$q = 0.05, 0.1, 0.15, 0.2$）では MaxSAT が概ね優位であるのが分かる．負の効用がある問題では、$l \in [35, 45]$ のときのみ MPDT が優位であるが、こ

*1 $i \neq j$ なら $A_i \subseteq A_j$ あるいは $A_i \supseteq A_j$ とならないようにする．

表 1: 実験 1 の平均実行時間（単位：秒）

解法	分割の効用が負となる確率 q				
	0	0.05	0.1	0.15	0.2
	（効用：一様分布）				
MaxSAT	4.90	4.76	4.77	4.08	3.90
MPDT	0.70	8.27	10.82	5.44	11.27
	（効用：NDCS）				
MaxSAT	5.50	5.62	5.75	4.83	4.51
MPDT	1.12	7.24	8.98	8.36	9.56

葉ノード数 l の範囲：$[15, 25]$

表 2: 実験 2 の平均実行時間（単位：秒）

解法	PDT ルールの葉ノード数 l の範囲				
	$[15, 25]$	$[20, 30]$	$[25, 35]$	$[30, 40]$	$[35, 45]$
MaxSAT	4.95	6.52	7.99	9.18	10.45
MPDT	6.57	14.76	11.73	13.46	6.60

確率 $q = 0.05$、効用：一様分布

れは、[野本 16a] によれば、MPDT 探索への負の効用の影響が l が大きくなることで薄れるためである．

MPDT の実行時間は、q の増減によって大きく変動する（特に一様分布の場合）が、概ね q が大きくなるに連れて大きくなっていくように見える．一方 MaxSAT は、q の増減にあまり影響を受けず安定的な性能を示している．また q が大きくなると、少しずつではあるが、実行時間が短くなっていくようである．q が 0.2 を超えてもこの傾向が続くか、実験で確かめる必要がある．

表 2 から、葉ノード数 l を増やしていくと MaxSAT の実行時間は増えていくのが分かる．これは、l が増えるとソフト節が増えていくためであると考えられる．QMaxSAT は、ソフト節から擬似ブール（PB:Pseudo Boolean）制約 [Roussel 09] を作りそれを SAT 符号化することにより、SAT ソルバーを利用して MaxSAT 問題を解いている．ソフト節が増えると SAT 符号化した PB 制約の処理の負荷が増すのが影響していると思われる．

一方 MPDT は、l を増大しても実行時間が増大するわけではない．むしろ $l \in [35, 45]$ になると実行時間が明らかに短くなっている．これは MPDT では、負の効用に関する処理が重く、これが性能を支配する主要因となっており、前述のように $l \in [35, 45]$ だとこの負担が軽減されているからだと思われる．

6. おわりに

本稿では、分割決定木で表現された提携構造形成問題の MaxSAT 符号化を形式的に述べた．計算機実験では、負の効用を含む問題については、先行研究の分枝限定法に基づくアルゴリズムより良い性能を示した．効用が負となる確率 q の増減に対しても安定的な性能を示した．これは、現実の問題を解く際にも有用な性質だと思われる．

提案手法では、MaxSAT 符号化後の節数は、PDT ルールの数に対しては線形に増加するが、エージェント数に対しては 3 乗のオーダで増加する．今回は、エージェント数を 100 に、PDT ルールの数を 8 に固定したが、これらを増加させたら性能がどうなるか、今後評価していきたい．

謝辞 本研究の一部は JSPS 科研費 16K00304 の助成を受けたものです．

参考文献

[Ieong 05] Ieong, S. and Shoham, Y.: Marginal contribution nets: a compact representation scheme for coalitional games, in Riedl, J., Kearns, M. J., and Reiter, M. K. eds., *Proceedings 6th ACM Conference on Electronic Commerce (EC-2005), Vancouver, BC, Canada, June 5-8, 2005*, pp. 193–202, ACM (2005)

[Koshimura 12] Koshimura, M., Zhang, T., Fujita, H., and Hasegawa, R.: QMaxSAT: A Partial Max-SAT Solver, *JSAT*, Vol. 8, No. 1/2, pp. 95–100 (2012)

[Liao 12] Liao, X., Koshimura, M., Fujita, H., and Hasegawa, R.: Solving the Coalition Structure Generation Problem with MaxSAT, in *IEEE 24th International Conference on Tools with Artificial Intelligence, ICTAI 2012, Athens, Greece, November 7-9, 2012*, pp. 910–915, IEEE Computer Society (2012)

[Roussel 09] Roussel, O. and Manquinho, V. M.: Pseudo-Boolean and Cardinality Constraints, in Biere, A., Heule, M., Maaren, van H., and Walsh, T. eds., *Handbook of Satisfiability*, Vol. 185 of *Frontiers in Artificial Intelligence and Applications*, pp. 695–733, IOS Press (2009)

[Skibski 15] Skibski, O., Michalak, T. P., Sakurai, Y., Wooldridge, M., and Yokoo, M.: A Graphical Representation for Games in Partition Function Form, in Bonet, B. and Koenig, S. eds., *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pp. 1036–1042, AAAI Press (2015)

[越村 16] 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, pp. 730–733 (2016)

[越村 17] 越村 三幸, 査 澳龍, 野本 一貴, 櫻井 祐子, 横尾 真：分割決定木で表現された提携構造形成問題の MaxSAT 符号化, 情報処理学会第 79 回全国大会 講演論文集 (2017)

[野本 16a] 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, in *JAWS 2016* (2016)

[野本 16b] 野本 一貴, 伊原 尚正, 櫻井 祐子, 横尾 真：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, 2016 年度人工知能学会全国大会 論文集 (2016)