

MC-nets を用いた提携構造形成問題の MaxSAT 符号化の改良と評価

An Improvement of MaxSAT Encoding for Coalition Structure Generation Using MC-nets and Its Evaluations

越村 三幸^{1*} 廖 曉鵬² 野本 一貴¹
Miyuki Koshimura¹ Xiaojuan Liao² Kazuki Nomoto¹
上田 俊³ 櫻井 祐子⁴ 横尾 真¹
Suguru Ueda³ Yuko Sakurai⁴ Makoto Yokoo¹

¹ 九州大学 ² 中国・成都理工大学 ³ 佐賀大学
¹ Kyushu University ² Chengdu University of Technology ³ Saga University

⁴ 産業技術総合研究所
⁴ National Institute of Advanced Industrial Science and Technology

Abstract: 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-nets 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-nets into Boolean logic and utilize an off-the-shelf MaxSAT solver as an optimization tool for solving the CSG problem. The method has the disadvantage that the number of constraints representing a transitive relation over rules becomes enormous as rules increased. This paper proposes a new method to decrease the number of constraints. Experiments show that the new method significantly outperforms the old one.

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、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、ルールで問題を表現する [4]。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 [9, 10]。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-net 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。本稿では、この制約数の削減手法を提案し、その効果を見積もる。また、計算機実験によりその効果を示す。

なお紙面の都合上、本稿では用語の定義のみを与え、例題は載せないが、興味ある読者は文献 [5] を参照されたい。

2 提携構造形成問題

エージェントの全体集合を $A = \{a_1, \dots, a_n\}$ 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 はエージェントの部分集合の集合 $CS = \{C_1, \dots, C_k \mid C_i \subseteq A\}$ 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れる ($v: 2^A \rightarrow \mathbb{R}$)。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。

*連絡先：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 7 4 4
E-mail: koshi@inf.kyushu-u.ac.jp

3 MC-nets とルール間の関係

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、 $(P_r, N_r) \rightarrow v_r$ 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。

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 [12]。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 [16, 3]。

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、ルールの効用は正值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。

$R = \{r_1, r_2, \dots, r_n\}$ を MC-net 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 ($1 \leq i < j < k \leq n$)。

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の数 T_r は、

$$T_r = {}_n C_3 \cdot 3 = n \cdot (n-1) \cdot (n-2)/2$$

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 $(r_i : (P_i, N_i) \rightarrow w_i) \in R$ 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、300 なら 13,365,300 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について、これを導入するかどうかを、 r_j と同提携で両立可能なルールに着目して判定する。

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: $\neg B_i \vee \neg B_j.$
3. 上記以外で、 r_i あるいは r_j と同提携で両立可能なルールがない場合: ハード節は導入しない。

4. 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 $\neg S(\{i, k\}) \vee \neg S(\{k, j\}) \vee S(i, j)$ ² 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 4 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。

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. $a \cdot {}_n C_2 \cdot 3$
2. $b \cdot {}_n C_2 \cdot 2$
3. 0

3. は、 n が十分大きければ滅多に起こらないと考えられる。次の 4. は、3. が起こる確率を 0 とした時の見積もりである。0 でない場合、見積もりはこれより小さくなる。

4. $(c+d) \cdot {}_n C_2 \cdot (n-2) \cdot a$. これは推移律を表すハード節の数で、これを T'_r と置く。他提携で両立可能な場合は、これに加えて、 $c \cdot {}_n C_2$ 個のハード節が加わる。

したがって、提案手法と従来手法の推移律を表すハード節の数の比 T'_r/T_r は、

$$(c+d) \cdot {}_n C_2 \cdot (n-2) \cdot a \div n \cdot (n-1) \cdot (n-2)/2 = n \cdot (n-1) \cdot (c+d) \cdot a/2 \div n \cdot (n-1)/2 = (c+d) \cdot a$$

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。

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。

² $S(\{i, j\}) = \begin{cases} S(i, j) & i < j \text{ のとき} \\ S(j, i) & j < i \text{ のとき} \end{cases}$

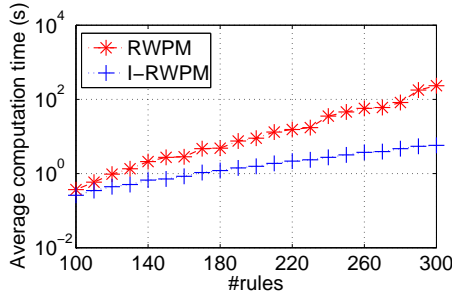


図 1: RWPM と I-RWPM の平均求解時間の推移

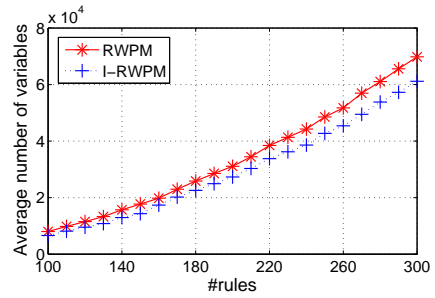
5 計算機実験

本節では、提案手法の効果を計算機実験により評価する。まず、文献 [12, 9, 10] と同じ方法で問題インスタンス (MC-nets のルール集合) を作成した。その際に、各インスタンスに含まれるルール数とエージェント数が等しくなるようにした。ルール数を 100 から 300 まで 10 刻みで増やししながら、それぞれ 100 インスタンスを作成した。つまり全部で 2100 (= 100 × 21) インスタンスを実験に用いた。それぞれのインスタンスは、負の利得を持つルールを 2 割程度含んでいる。計算は全て、メモリ 4GB の 3.7GHz quad-core Intel i3-4170 プロセッサの計算機上で行なった。なお本節で、RWPM (rule relation-based WPM) は従来手法、I-RWPM (improved rule relation-based WPM) は提案手法を表す。

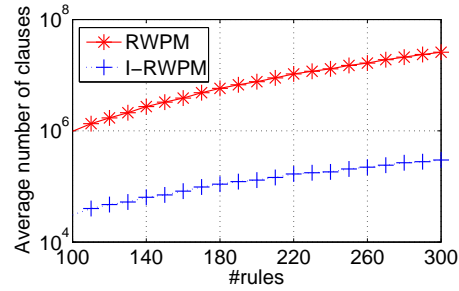
5.1 最適な MaxSAT ソルバーの選択

まず、幾つかの代表的な MaxSAT ソルバーを用いて全 2100 インスタンスを解いてみた。実験には 6 個の MaxSAT ソルバー、MaxHS [2]、LMHS [13]、NaPS [14]、Maxino [1]、QMaxSAT [7]、SAT4j [8] を用いた。制限時間は 1 インスタンス当たり 900 秒とした。

表 1 に結果を示す。数値は、各 100 インスタンスの求解時間の平均である。カッコ内の数は、対応するソルバー (その列の最上段) が、対応するルール数 (その行の最左欄) のインスタンス中、制限時間内で解けたインスタンス数を表している。100 インスタンス全て解けた場合は、この数は省略した。右の 3 つのソルバーはルール数 290 の 100 インスタンスの内、1 インスタンスが解けなかったため、ルール数 300 のインスタンスは解けなかった。表中の “/” はそのことを表わす。この実験より、I-RWPM に最も適しているのは、MaxHS であることが分かる。



(a) ブール変数の数



(b) 節の数

図 2: RWPM と I-RWPM の変数と節数の推移

5.2 従来手法と提案手法の比較

提案手法 (I-RWPM) と従来手法 (RWPM) を比較するため、それぞれの方法で符号化したインスタンスを MaxHS を用いて解いた。図 1 にその結果を示す。ルール数が増えるに従い両者の性能が少しずつ開いていくことが分かる。ルール数 300 のインスタンスでは、I-RWPM の方が 10 倍以上、高速に問題を解いている。

次に、I-RWPM と RWPM により作成したインスタンス中のブール変数の数と節の数を比較した。図 2(a) は変数の数の比較、図 2(b) は節の数の比較、を示している。これもルール数が増えるに従い両者の差が少しずつ開いている。図 2(b) は縦軸が対数の片対数グラフなので、両者の違いはより顕著である。これは、本研究の目的である制約数 (節数) の削減に実験的にも成功していることを示している。

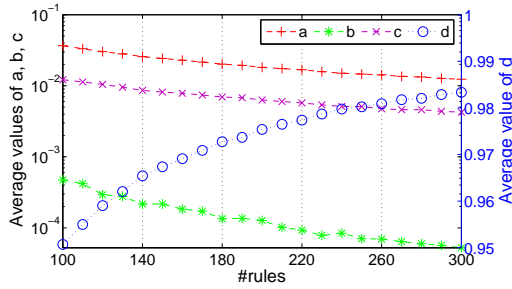
4 節のハード節数の見積もりで用いた確率 a 、 b 、 c 、 d の確率が実際には幾ら位かを実験で用いたインスタンスで計算した。図 3(a) にそれを示す。 a 、 b 、 c の値は左軸の目盛、 d の値は右軸の目盛、で示される。ルール数が増えるに従い、 a 、 b 、 c は減り、 d は増えていくことが分かる。ルール数 300 では、 d の値は 0.98 を超える。これは、ある一つのルールに着目すると 98% 超のルールはそのルールと独立であることを示している。

図 3(b) は、推移律を表わすハード節の数の比 $T'_r/T_r (= (c+d) \times a)$ の推移を示している。ルール数が増えるに従い、この比は減少している。つまり、ルール数が増えるにつれて、提案手法によるハード節の削減効果は

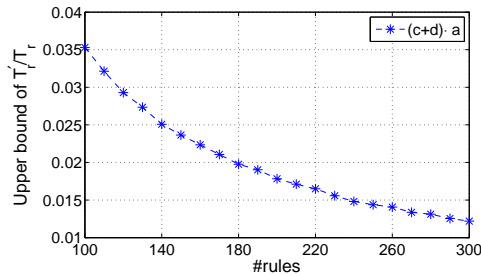
表 1: 各ソルバーの I-RWPM 符号化によるインスタンスの平均求解時間 (秒) の推移

ルール数	MaxHS	LMHS	Maxino	QMaxSAT	NaPS	SAT4j
100	0.259	0.412	0.114	0.070	2.068	0.203
110	0.347	0.571	0.162	0.101	2.503	0.241
120	0.442	0.719	0.234	0.162	2.988	0.302
130	0.506	0.833	0.277	0.218	3.321	0.332
140	0.664	1.081	0.469	0.341	4.136	0.379
150	0.715	1.191	0.461	0.420	4.396	0.419
160	0.842	1.418	0.509	0.343	4.374	0.419
170	1.065	1.810	0.741	0.616	5.309	0.561
180	1.200	2.038	0.831	0.537	5.247	0.537
190	1.417	2.379	1.066	0.844	5.816	0.614
200	1.565	2.694	1.203	1.006	6.151	0.707
210	1.880	3.134	1.699	1.475	7.490	0.863
220	2.165	3.557	1.987	1.614	8.614	1.007
230	2.344	3.903	2.190	1.746	9.132	1.053
240	2.727	4.390	3.586	4.879	14.131	1.548
250	3.173	4.964	3.674	5.395	14.351	2.282
260	3.727	5.515	4.555	5.974	15.955	3.148
270	3.915	5.826	4.705	6.145	17.982	3.805
280	4.680	6.629	6.454	8.715	24.293	5.414
290	5.413	7.493	10.361	14.532(99)	33.895(99)	8.397(99)
300	5.765	8.005	11.052	/	/	/

高まっている。



(a) a, b, c, d の平均



(b) $T'_r/T_r = (c + d) \cdot a$ の見積

図 3: T'_r/T_r に影響を与えるパラメータ値の推移

6 おわりに

本稿では、MC-nets 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。計算機実験では、ルール数が 300 のとき、節数は 10^7 超から 10^5 程度に削減、計算時間は 100 秒超から 10 秒以内に短縮できた。

今後、分割決定木 [15] を用いた提携構造形成問題の解法 [11, 6, 17] にも提案手法を適用したいと考えている。

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] M. Alviano, C. Dodaro, and F. Ricca. A maxsat algorithm using cardinality constraints of bounded size. In *IJCAI*, pp. 2677–2683, 2015.
- [2] J. Davies and F. Bacchus. Exploiting the power of mip solvers in maxsat. In *International conference on theory and applications of satisfiability testing*, pp. 166–181. Springer, 2013.
- [3] 一村, 長谷川, 上田, 岩崎, 横尾. 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, 94(11):1707–1715, nov 2011.
- [4] S. Jeong and Y. Shoham. Marginal contribution nets: a compact representation scheme for coalitional games. In *Proceedings of the 6th ACM Conference on Electronic Commerce (ACM EC)*, pp. 193–202, 2005.
- [5] 越村, 廖, 野本, 上田, 櫻井, 横尾. Mc-nets を用いた提携構造形成問題の MaxSAT 符号化の改良. 日本ソフトウェア科学会第 34 回大会 (2017 年度) 講演論文集, 2017.
- [6] 越村, 查, 野本, 櫻井, 横尾. 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.
- [7] M. Koshimura, T. Zhang, H. Fujita, and R. Hasegawa. Qmaxsat: A partial max-sat solver. *Journal on Satisfiability, Boolean Modeling and Computation*, 8:95–100, 2012.
- [8] D. Le Berre and A. Parrain. The sat4j library, release 2.2, system description. *Journal on Satisfiability, Boolean Modeling and Computation*, 7:59–64, 2010.
- [9] X. Liao, M. Koshimura, H. Fujita, and R. Hasegawa. 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.
- [10] X. Liao, M. Koshimura, H. Fujita, and R. Hasegawa. Maxsat encoding for mc-net-based coalition structure generation problem with externalities. *IEICE Transactions*, 97-D(7):1781–1789, 2014.
- [11] 野本, 大田, 上田, 櫻井, 横尾. 分割関数ゲームの提携構造形成アルゴリズム. In *JAWS 2016*, 9 月 2016.
- [12] N. Ohta, V. Conitzer, R. Ichimura, Y. Sakurai, A. Iwasaki, and M. Yokoo. Coalition structure generation utilizing compact characteristic function representations. In *Proceedings of the 15th International Conference on Principles and Practice of Constraint Programming (CP)*, pp. 623–638, 2009.
- [13] P. Saikko, J. Berg, and M. Järvisalo. Lmhs: a sat-ip hybrid maxsat solver. In *International Conference on Theory and Applications of Satisfiability Testing*, pp. 539–546. Springer, 2016.
- [14] M. Sakai and H. Nabeshima. Construction of an robdd for a pb-constraint in band form and related techniques for pb-solvers. *IEICE TRANSACTIONS on Information and Systems*, 98(6):1121–1127, 2015.
- [15] O. Skibski, T. Michalak, Y. Sakurai, M. Wooldridge, and M. Yokoo. A Graphical Representation for Games in Partition Function Form. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pp. 1036–1042, 2015.
- [16] S. Ueda, T. Hasegawa, N. Hashimoto, N. Ohta, A. Iwasaki, and M. Yokoo. Handling negative value rules in mc-net-based coalition structure generation. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 795–804, 2012.
- [17] A. Zha, K. Nomoto, S. Ueda, M. Koshimura, Y. Sakurai, and M. Yokoo. Coalition structure generation for partition function games utilizing a concise graphical representation. In B. An, A. L. C. Bazzan, J. Leite, S. Villata, and L. W. N. van der Torre eds., *PRIMA 2017: Principles and Practice of Multi-Agent Systems - 20th International Conference, Nice, France, October 30 - November 3, 2017, Proceedings*, Vol. 10621 of *Lecture Notes in Computer Science*, pp. 143–159. Springer, 2017.