3C-02

分割決定木で表現された提携構造形成問題のMaxSAT符号化

越村 三幸 † 査 澳龍 ‡ 野本 一貴 ‡ 櫻井 祐子 † 横尾 真 †
† 九州大学大学院システム情報科学研究院 ‡ 九州大学大学院システム情報科学府

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．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．本研究で扱う分割決定木 [1] 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 [4, 3]．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．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 は、SAT（Boolean satisfiability）を最適化問題に拡張したものである [2]．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．

本稿では用語の厳密な定義などは避け、具体例を用いて MaxSAT 符号化のあらましを述べる．

A MaxSAT Encoding of Partition Decision Trees for Coalition Structure Generation
Miyuki KOSHIMURA†, Aolong ZHA‡, Kazuki NOMOTO‡, Yuko SAKURAI† and Makoto YOKOO†
†Faculty of Information Science and Electrical Engineering, Kyushu University
‡Graduate School of Information Science and Electrical Engineering, Kyushu University

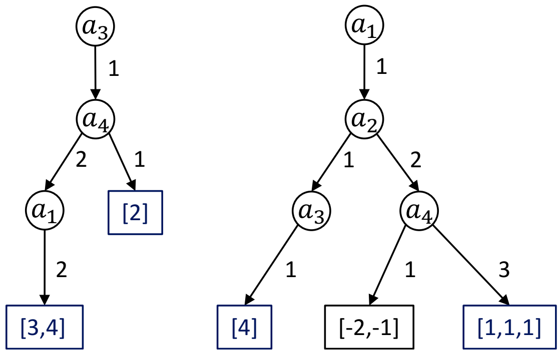

図 1: 分割決定木による記述例

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（$w : (C, CS) \rightarrow \mathbb{R}$）．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

分割決定木は分割関数の簡潔表現の一つで、複数の多分木によって関数を表現する．図 1 に分割決定木の例を示す．例では二つの木によって分割関数を表現している．

内部ノードはエージェントを表し、各枝にはその枝が出ているノードのエージェントがどの提携に含まれるかを示す番号が付与されている．葉は、根からその葉に至る経路上に存在するエージェントらで構成される分割の各提携の効用ベクトルを表す．

Copyright ©2017 Information Processing Society of Japan.
All Rights Reserved.

図１左の木の最左の葉から根に至る経路では、a_3 の下の枝には 1、a_4 の下の枝には 2、a_1 の下の枝には 2 が付与されている．これは、第１の提携に a_3、第２の提携に a_1 と a_4 が属していれば、第１の提携に 3、第２の提携に 4 の効用があることを表している．ここで効用の 3 は葉の効用ベクトルの第１要素、4 は第２要素として示されている．なお、第１の提携と第２の提携は異なることが要請される．つまり、a_3 と a_1（あるいは a_4）は、異なる提携に属していなければならない．

紙面の都合上、その他の経路の説明は割愛する．

4　MaxSAT 符号化

MaxSAT による問題解決では、問題をハード節とソフト節の集合で表現する．これを MaxSAT 符号化と呼ぶ．ソフト節は正整数の重みを持つ．MaxSAT の目的は、全てのハード節を満たし、満たされるソフト節の重みの和が最大となるような解を求めることである．

このように、通常の MaxSAT では一般の論理式ではなく、節[1]と呼ばれる制限された論理式のみで問題を表現する．また、ソフト節の重みも正数に限られる．本稿では、符号化を理解しやすくするために、MaxSAT を一般の論理式や負数も扱えるように拡張し、これへの符号化を先ず考える．そして、これを通常の MaxSAT 形式に変換する．

先ず、原子論理式 $C(i,j)$ 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 $<$ を導入して $C(i,j)(i<j)$ とする．例では a_1 から a_4 の４つのエージェントが現れるので、$C(a_1,a_2)$, $C(a_1,a_3)$, $C(a_1,a_4)$, $C(a_2,a_3)$, $C(a_2,a_4)$, $C(a_3,a_4)$ の６個の原子論理式を用意する．

符号化では、木の葉１つに対してソフト論理式１つを作る．従って例では、５個のソフト論理式が作られる．先ず、左の木の最左の葉より、$(\neg C(a_3,a_4) \wedge C(a_1,a_4), 7)$ なるソフト論理式[2]が作られる．これは、エージェント a_1 と a_4 を含むような提携があり、エージェント a_3 がその提携に属さないような提携構造には $7(=3+4)$ の効用がある、ことを表す．以下、左の葉から右へ、同様にして次のようなソフト論理式が得られる．

(1)　$(\neg C(a_3,a_4) \wedge C(a_1,a_4), 7)$

(2)　$(C(a_3,a_4), 2)$

(3)　$(C(a_1,a_2) \wedge C(a_1,a_3), 4)$

(4)　$(\neg C(a_1,a_2) \wedge C(a_1,a_4), -3)$

(5)　$(\neg C(a_1,a_2) \wedge \neg C(a_1,a_4) \wedge \neg C(a_2,a_4), 3)$

提携関係は推移律を満たすので、この他に次のようなハード論理式を加える．

$$C(a_1,a_2) \wedge C(a_2,a_3) \Rightarrow C(a_1,a_3)$$

例の場合、このようなハード論理式は合わせて１２個作られる．

次に、負の重み $w(<0)$ のソフト論理式 (F,w) を正の重みのソフト論理式 $(\neg F, -w)$ に変換する．こうしてソフト論理式 (4) は、

(4')　$(C(a_1,a_2) \vee \neg C(a_1,a_4), 3)$

に変換される．こうして得られるソフト論理式 (1) (2) (3) (4') (5) のうち、(2) と (4') は既にソフト節の形式になっているので、残りの (1) (3) (5) をソフト節に変換すれば良い．

一般に $(l_1 \wedge \cdots \wedge l_n, w)$ の形のソフト論理式は、原子論理式 b を新たに導入して、１つのソフト節 (b,w) と n 個のハード論理式 $b \Rightarrow l_i (i=1,\ldots,n)$ に変換すれば良い．(1) (3) (5) は次のように変換される．

(1')　$(b_1, 7)$　$b_1 \Rightarrow \neg C(a_3,a_4)$　$b_1 \Rightarrow C(a_1,a_4)$

(3')　$(b_3, 4)$　$b_3 \Rightarrow C(a_1,a_2)$　$b_3 \Rightarrow C(a_1,a_3)$

(5')　$(b_5, 3)$　$b_5 \Rightarrow \neg C(a_1,a_2)$　$b_5 \Rightarrow \neg C(a_1,a_4)$　$b_5 \Rightarrow \neg C(a_2,a_4)$

なお、ハード論理式 $a_1 \wedge \cdots \wedge a_n \Rightarrow l$ は、ハード節 $\neg a_1 \vee \cdots \vee \neg a_n \vee l$ と等価なので、最終的に得られるハード論理式は、提携関係の推移律を表すものも含めて全てハード節とみなせる．

5　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 符号化を述べた．今後、計算機実験により性能を評価したい．

謝辞　本研究は JSPS 科研費 16K00304 の助成を受けたものです．

参考文献

[1] O. Skibski, T. P. Michalak, Y. Sakurai, M. Wooldridge, M. Yokoo. A Graphical Representation for Games in Partition Function Form. In Proc. of AAAI, pp.1036-1042, 2015

[2] 越村、藤田．MaxSAT：SAT の最適化問題への拡張．情報処理、Vol.57、No.8、pp.730-733、2016

[3] 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．JAWS 2016

[4] 野本、伊原、櫻井、横尾．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．人工知能学会全国大会、2D4-4、2016

[1] リテラルの選言（論理和）を節と呼ぶ．原子論理式あるいはその否定をリテラルと呼ぶ．

[2] 重み w のソフト論理式 F を (F,w) と表すことにする．

Copyright ©2017 Information Processing Society of Japan.
All Rights Reserved.