情報処理学会論文誌　Vol.53　No.8　2006–2017 (Aug. 2012)

自動メカニズムデザインを利用した組合せオークションのルール抽出アルゴリズムの提案

毛利 貴之[1,a)]　杉町 勇和[1,b)]　東藤 大樹[1,c)]　岩崎 敦[1,d)]　横尾 真[1,e)]

受付日 2011年11月10日, 採録日 2012年5月12日

概要：一般に組合せオークションのためのメカニズムは割当て規則と支払い規則の 2 つの関数によって構成される．メカニズムに関する著名な研究成果として，理論的に優れた性質を持つ Vickrey-Clarke-Groves（VCG）メカニズムが知られている．しかしながら，VCG メカニズムにはいくつかの問題点があると近年指摘されている．これまでに，その問題に対して頑健性が保証されるメカニズムがいくつか提案されている．従来，これらの関数は人手で設計されてきたが，多大な時間と労力がかかる．そこで近年，整数計画法を利用した自動設計の手法（自動メカニズムデザイン）が提案されている．しかしながら，自動メカニズムデザインは小規模な問題にしか対応できず，現実の問題サイズを扱うことができない．そのため従来研究では，小規模な問題を解き，出力された結果の中から特徴的な結果を抽出し，一般に適用可能なルールを求めようとしている．それでも一般的なルールを得るには人手による部分がまだまだ多い．また，得られたルールが必ずしも適切だとは限らず，別途検証する必要がある．本論文では，自動メカニズムデザインの出力結果から一般的なルールを抽出するアルゴリズムを提案する．具体的には商品の割当て方法や支払額を決定する関数は，あるしきい値を超えているかどうかで勝敗が決まるとし，そのしきい値を出力するアルゴリズムである．

キーワード：組合せオークション，ゲーム理論，メカニズムデザイン，架空名義入札，最適化

A Rule Extraction Algorithm for Combinatorial Auctions with Automated Mechanisms Design

Takayuki Mouri[1,a)]　Toshikazu Sugimachi[1,b)]　Taiki Todo[1,c)]
Atsushi Iwasaki[1,d)]　Makoto Yokoo[1,e)]

Received: November 10, 2011, Accepted: May 12, 2012

Abstract: Automated mechanism design (AMD) provides a novel scheme to design social choice rules (e.g., combinatorial auction mechanisms) by using mathematical programming technique. However, it only returns a discretized set of outcomes, i.e., allocations and the associated payments given possible type profiles. Therefore, for large combinatorial auction problems, there has been no scheme to effectively generalize a social choice rule for a continuous set of outcomes in the literature of mechanism design. In this paper, we formalize the problem as a rule extraction problem. We then propose a new algorithm to automatically extract a payment rule of a combinatorial auction mechanism from the discretized set of outcomes obtained from AMD. Our experiments reveal that the proposed algorithm successfully extracts payment rules of two existing combinatorial auction mechanisms that is either strategy-proof or false-name-proof.

Keywords: combinatorial auctions, game theory, mechanism design, false-name-bidding, optimization

[1] 九州大学大学院システム情報科学府情報学専攻
Graduate School of ISEE, Kyushu University, Fukuoka 819–0395, Japan
[a)] mouri@agent.inf.kyushu-u.ac.jp
[b)] sugimachi@agent.inf.kyushu-u.ac.jp
[c)] todo@agent.inf.kyushu-u.ac.jp
[d)] iwasaki@inf.kyushu-u.ac.jp
[e)] yokoo@inf.kyushu-u.ac.jp

1. 序論

ある環境に存在する戦略的な行動をするエージェントの集団に対して，集団としての意思決定方式（メカニズム）を導入すると，何らかの社会的な結果が得られる．社会的に望ましい結果を得られるようにメカニズムを設計するこ

とは，メカニズムデザイン（制度設計）と呼ばれ，ミクロ経済学，ゲーム理論の一分野として活発に研究が行われている [10], [11], [16].

本研究では，オークション方式（オークションメカニズム）の設計に注目する．インターネットの環境下において，大規模なオークションが実行可能となり，不特定多数の人々が参加可能となった．一方で，ネットワークの匿名性を利用した様々な不正行為が指摘されている．そのため，オークションメカニズムの設計にあたっては，様々な不正行為に対する頑健性やオークションの結果に関する，何らかの理論付け等が重要となる．

インターネットオークションに関連する研究の 1 つに組合せオークション [2], [3] がある．通常のオークションは 1 度に 1 つの商品（財）を販売することに対し，組合せオークションでは，価値に依存性のある（代替性や補完性）複数種類の異なる財を販売する．各エージェントは財の組合せ（バンドル）に対して入札を行う．エージェントの複雑な選好を考慮することで，エージェントおよびオークション主催者の利益（効用）を増加させることができる．メカニズム設計に関する著名な研究成果として，理論的に優れた性質を持つメカニズムである Vickrey-Clarke-Groves（VCG）メカニズム [15] が知られている．VCG メカニズムはエージェントが真の評価値を申告することが最適な戦略（支配戦略）となることを保証するメカニズムである．

しかしながら，VCG メカニズムは万能ではなく，いくつかの問題点が指摘されている．たとえば，あるエージェントが，複数のメールアドレス等の複数の名義を用いてオークションに参加可能な場合（このような不正行為は架空名義入札と呼ばれる），各エージェントにとって，単一の名義を用いて真の評価値を申告することは，もはや最適な戦略とはならない．これまでに，架空名義入札に対して頑健性が保証される，いくつかのメカニズム [5], [8] が提案されている．

従来，メカニズム設計は人手によって行われてきたが，多大な労力と膨大な時間がかかってしまうことが難点である．近年では，メカニズム設計を最適化問題として定式化し，整数計画法を用いてメカニズムを自動設計するアイデア（自動メカニズムデザイン）[1], [9], [12] が提案されている．具体的には，メカニズムが入力（参加者のタイプの集合）と出力（財の割当てと支払額）の関係を示す表であると考える．表の各項目を整数計画法の変数とし，制約条件（真の評価値を入札することが最適，単一の名義を用いて入札することが最適等）のもとで，参加者の利益や収入の最大化を目的関数として最適解を求める．

しかしながら，整数計画法を用いた自動メカニズム設計では表の項目数/変数の数は参加者数に関して指数的に増加し，大規模な問題では最適解を得ることが不可能となる．このため，本来は連続，もしくは多数の可能性のある参加者のタイプ，たとえばオークションなら商品の価値を，ごく少数の離散的な候補値に絞ることにより，整数計画法の最適解を得ることを可能にしている．たとえば，本来の評価値が 0 から 100 の間の任意の整数値である場合に，非常に高い評価値である 100，中間的な評価値である 50，非常に低い評価値である 0 の 3 通りの可能性に限って最適化を行うといったことが必要になる．

現状では，このような代表的な値に対する自動メカニズム設計の出力結果を，人間が解析して一般的なルールを求めようとしている．しかしながら，表の項目が数百程度となった場合，人手により結果を解析し一般的なルールを得ることは非常に困難となる．また，自動メカニズム設計の結果は絞り込みを行った特定の入力に特化して最適化されており，必ずしも一般的なルールが得られるとは限らない．例外も考慮しながら一般的なルールの抽出を試みる必要があり，適切なルールを抽出するためにはメカニズム設計に関する知識やスキルが必要となる．さらに，得られたルールの候補を検証し，異なる入力で自動メカニズム設計を実行するといった繰返しが必要となる．

そこで本論文では，自動メカニズムデザインで得られた解から一般的なルールを自動的に抽出するアルゴリズムを開発する．このように膨大なデータから何らかの一般的なルール/法則を発見するアルゴリズムはデータマイニングや知識発見の分野でさかんに研究されている [4], [6], [7]. しかしながら，メカニズムを表す関数，具体的には商品の割当て方法や支払額を決定する関数は，あるしきい値を超えているかどうかで勝敗が決まる等，不連続，非線形であり，科学的法則発見等の分野で扱われてきた法則と比較すると，複雑で理論的に扱いにくい形式であることが通例である．このため，本論文では，組合せオークションメカニズムの設計に特化した，集合被覆アルゴリズムや条件分岐を利用したアルゴリズムを提案する．提案手法により，従来開発された架空名義入札に頑健なメカニズムを再発見することが可能となっている．現在，架空名義入札に頑健な新しいメカニズムの発見，さらに，集団に対する無羨望性 [14] 等の，新しい要求条件を満たすメカニズムの発見のために，提案手法の適用を進めている．

2. 準備

本章では，組合せオークションの定義を行う．まず問題の定式化として，変数の定義を行い，オークションメカニズムが社会的に望ましい結果をもたらすための満たすべき性質について述べる．そして，整数計画法を用いた自動メカニズムデザインについて説明する．

2.1 問題の定式化

n 人のエージェントが m 種類の財を競り合う組合せオークションを考える．エージェントの集合を $N = \{1, 2, \ldots, n\}$,

財の集合を $M=\{g_1,g_2,\ldots,g_m\}$ とする．各エージェント i は財の組合せ（バンドル）$B\subseteq M$ に対する評価値を持つ．i の $B\subseteq M$ に対する評価値は，タイプと呼ばれる $\theta_i\in\Theta$ を用いて $v(B,\theta_i)$ で表現する．バンドルに関する評価値は $B'\supseteq B$ において，つねに $v(B',\theta_i)\geq v(B,\theta_i)$ が成立する．本論文における評価値は整数であると仮定する．各エージェントが申告するタイプの組を $\theta=(\theta_1,\ldots,\theta_n)\in\Theta^n$ とする．エージェントのタイプに関する仮定として単一バンドル指向を定義する．

定義 1（単一バンドル指向）．単一バンドル指向なエージェントとは，唯一のバンドル（もしくは，そのスーパセット）のみを欲しがるエージェントを意味する．すなわち，もし，あるエージェント i が単一バンドル指向であるならば，あるバンドル B_i が存在し，$B'_i\supseteq B_i$ となる任意の B'_i に対して $v(B'_i,\theta_i)=v(B_i,\theta_i)>0$ が成立する，かつすべての $B'_i\not\supseteq B_i$ において $v(B'_i,\theta_i)=0$ が成立する．

組合せオークションメカニズム $\mathcal{M}(X,p)$ は一般的に割当て規則 X と支払い規則 p の 2 つの関数から構成される．A を可能な割当ての集合とすると，割当て規則 $X:\Theta^n\rightarrow A$ は各エージェントの申告したタイプの組を入力とし，各エージェントへの財の割当てを出力する．支払い規則 $p:\Theta^n\rightarrow\mathbb{R}^n$ は各エージェントの申告したタイプの組を入力とし，各エージェントが支払う金額を出力する．$\mathbb{R}^n$ とは，n 次元の実数の集合である．次に i 以外のタイプの組を θ_{-i} とする．このとき，i が θ_i を申告したときの i への割当ておよび支払額を $X_i(\theta_i,\theta_{-i})$，$p_i(X_i(\theta_i,\theta_{-i}),\theta_{-i})$ と表す．さらに i が θ_i を申告して財を獲得できるときの効用を準線形と仮定し，$v(X_i(\theta_i,\theta_{-i}),\theta_i)-p(X_i(\theta_i,\theta_{-i}),\theta_{-i})$ とする．ただし，入札者 i が何も獲得できない（$X_i(\theta_i,\theta_{-i})=\emptyset$）ときの支払額 $p_i(\emptyset,\theta_{-i})$ は 0 とする．

メカニズム設計者はその目的に応じて，望ましい結果を与える割当て規則と支払い規則を設計する．ここでは，オークションメカニズムが満たすべき代表的な性質について述べる．

割当て可能性 割当て可能性とは，財を割り当てる際に，異なるエージェント同士が同一の財を取り合わないようにしなければならないという性質である．

個人合理性 個人合理性とは，オークションに参加することで得られる効用が，オークションに参加しない場合の効用以上でなければならないという性質である．すなわち，オークションに参加することで効用が負になるならば，そのようなオークションに各エージェントは参加しない．

準匿名性 準匿名性とは，各エージェントに任意につけられる名前や番号には意味がないという性質である．すなわち，同じタイプを申告したエージェント同士の効用は等しくならなければならない．

入札と財の対称性 入札と財の対称性とは，各財は対等であり，財の名前を入れ替えてもエージェントの効用は変化しないという性質である．

戦略的操作不可能性 戦略的操作不可能性とは，各エージェントにとって，各財に対して真の評価値を入札することが支配戦略となるという性質である．すなわち，エージェントが自分のタイプを偽ったときの効用が，真のタイプを申告したときの効用よりつねに小さくなければならないことを意味する．この定義により，エージェントは自身の真のタイプを入札することで最大の効用を得られることになる．戦略的操作不可能なオークションメカニズムには各エージェントが財を獲得するために必要最低限の値がそれぞれ存在する．この値をしきい値（critical value）と呼ぶ．しきい値は他のエージェントの申告を固定した場合に一意に定まり，エージェントはそのしきい値を超える値を申告することで，財を獲得できる．その際，しきい値を支払うことになる．

架空名義操作不可能性 架空名義操作不可能性とは，各エージェントにとって，単一名義によって真の評価値を入札することが支配戦略となるという性質である．すなわち，各エージェントが複数の名義を用いたときの効用が，単一名義を用いたときの効用よりもつねに小さくなければならないことを意味する．ここでもし，エージェントがただ 1 つの名義しか使えないと仮定すると，戦略的操作不可能性と架空名義操作不可能性は等価となる．

本論文では，割当て可能性，個人合理性，準匿名性，入札と財の対称性はつねに制約として導入されているものと仮定する．

2.2 自動メカニズムデザイン

従来，理論的なメカニズムは人手で設計されてきたが，近年，メカニズム設計の問題自体を最適化問題に帰着し，最適化手法（整数計画法）を用いてメカニズムを設計する手法である自動メカニズムデザイン（Automated mechanism design，AMD）[1], [9], [12] が提案されている．AMD は目的・設定に応じて最適なメカニズムを構築でき，メカニズム設計の労力を人手から機械に移すことができる．

AMD では，まずエージェントの数 n，財の数 m，もしくはエージェントの集合 N，財の集合 M を入力する．さらにエージェントに与えられうるタイプの組合せ Θ^n を入力する．ここで Θ は離散的集合とする．

次に目的関数を設定する．本研究で実装している目的関数は以下の 3 つである．

社会的余剰の期待値の最大化 社会的余剰とは，各エージェントおよびオークション主催者を含むすべての参加者の効用の総和である．社会的余剰の期待値の最大化を目的関数に設定することで，自動メカニズムデザイン

はすべての参加者の効用の総和が可能な限り最大化されるようにメカニズムを設計する．

主催者収入の期待値の最大化 主催者収入の期待値の最大化を目的関数に設定することで，自動メカニズムデザインは各エージェントの支払う金額の総和が可能な限り最大化されるようにメカニズムを設計する．

最悪実行時における社会的余剰の比率の最大化 可能な財の割当ての集合の中で，すべての参加者の効用の総和が最大となる割当てをパレート効率的な割当てと呼ぶ．しかしながら，2.1 節で述べた社会的に望ましい性質をオークションメカニズムが満たすためには，ある程度の効用を犠牲にしなければならない場合が存在する．最悪実行時における社会的余剰の最大化を目的関数に設定することで，自動メカニズムデザインは，自動メカニズムデザインにより決定された割当てとパレート効率的な割当てにおけるそれぞれの社会的余剰の比率の最低値を可能な限り最大化するメカニズムを設計する．

本研究では，その中でも全体の余剰を最大化する社会的余剰の期待値の最大化に限定して議論する．

目的関数の設定をした後，制約式を設定する．ここでは，2.1 節で述べたオークションメカニズムが満たすべき性質を制約式として導入するか否かを設定する．

このようにエージェント数，財の数，タイプの組を入力し，目的関数，制約条件を設定する．この入力から，AMD は n 人のエージェントが申告しうるタイプの組 Θ^n のそれぞれに対応する割当てと支払額の表をオークションの結果の集合 $\mathcal{O}$ として出力する．

ここで 3 人 2 財の組合せオークションを考える．ある入札者 i に与えられるタイプ $\theta_i \in \Theta$ をバンドル $\{g_1\}, \{g_2\}$ および $\{g_1, g_2\}$ に対する評価値で表す．たとえば，タイプ $(4, 0, 4)$ はそれぞれ $(\{g_1\}, \{g_2\}, \{g_1, g_2\})$ に対する評価値を表す．エージェントのタイプ集合 Θ を $\{(0, 0, 0), (0, 0, 3), (0, 0, 5), (0, 0, 7), (2, 0, 2), (4, 0, 4), (6, 0, 6), (0, 2, 2), (0, 4, 4), (0, 6, 6)\}$ と与えたとき，戦略的操作不可能性を制約条件として，社会的余剰を可能な限り最大化した AMD の出力結果が**表 1** である．実際の AMD は 10^3 通りのタイプの組に対する割当てと支払額を出力するが，その中から 3 組に対する割当てのみを示している．表 1 において，エージェントのタイプと書かれた列はエージェント 0，1，2 の申告したタイプ，すなわち，どのバンドルにいくら入札したかを表している．財の割当てと書かれた列は，そのタイプの組に対して，誰にどの財を割り当てるかを表している．たとえば，AMD の結果 o^1 では，3 人のエージェントがそれぞれタイプを申告した際，すなわち，エージェント 0 が $\{g_1, g_2\}$ に 4，エージェント 1 が $\{g_1\}$ に 3，エージェント 2 が $\{g_2\}$ に 2 を申告した際，エージェント 0 には何も割り当てず，エージェント 1 に $\{g_1\}$，エージェント 2 に $\{g_2\}$ を割り当てることになる．

自動メカニズムデザインはこのように与えられた範囲のタイプに特化して最適化されたメカニズムが設計可能である．しかし，自動メカニズムデザインによるメカニズム設計では，表の項目数/変数の数はエージェント数に関して指数的に増加してしまうことが問題となっている．各エージェントに与えられうるタイプは $t = |\Theta|$ 種類あるとき，ありうるタイプの組合せは t^n 通りとなり，この混合整数計画問題の決定変数は，$t^n \times (n+1)^m$ 個の割当て決定関数と $t^n \times n$ 個の支払額決定変数を表現しなければならない．たとえば，4 人 3 財 9 タイプと入力したところ 82 万以上の変数を生成してしまう．加えて，戦略的操作不可能性等の性質のための制約式もタイプの組合せに対して追加される．この結果，Intel Xeon E5540@2.53 GHz，24.0 GB RAM を搭載した PC において実験したところ，4 日以上かけても問題を解くことはできなかった．近年では，変数の数や制約の数を減らす研究が行われており [13]，4 日以上かけても解くことができなかった問題をおよそ 13 時間で解けるようになった例も存在するが，現実的な規模の問題では最適解を得ることが不可能である．

そこで，AMD の現実的な使い方として，本来は連続となるエージェントのタイプをごく少数の離散的な候補値に絞り込むことで，整数計画法の最適解を得ている．たとえば，本来の評価値が 0 から 100 の間の任意の整数値である場合に，高い値である 100，中間的な値である 50，低い値である 0 の 3 つに限って最適化を行うといったことが必要になる現状では，このような代表的な値に対する AMD の出力結果を人手で解析して一般的なルールを求めてきた．この手法によって，文献 [5] では，適応的留保価格（adaptive reserve price，ARP）メカニズムの設計に成功している．

表 1 戦略的操作不可能性を制約とした AMD の出力結果
Table 1 Output of AMD constrained with strategy-proofness.

AMD の結果	エージェントのタイプ			財の割当て		
	0	1	2	0	1	2
o^1	$\{g_1, g_2\}$ - 4	$\{g_1\}$ - 3	$\{g_2\}$ - 2	$\emptyset$	$\{g_1\}$	$\{g_2\}$
o^2	$\{g_1, g_2\}$ - 6	$\{g_1\}$ - 3	$\{g_2\}$ - 2	$\{g_1, g_2\}$	$\emptyset$	$\emptyset$
o^3	$\{g_1, g_2\}$ - 6	$\{g_1\}$ - 5	$\{g_2\}$ - 2	$\emptyset$	$\{g_1\}$	$\{g_2\}$
$\vdots$		$\vdots$			$\vdots$	

しかしながら，表の項目数が数百程度となった時点で，人手により解析し，一般的なルールを抽出することは困難となる．また，AMD の結果は絞り込みを行った特定の入力に特化して最適化されており，必ずしも一般的なルールが得られるとは限らない．そのために，得られたルールの候補を検証し，さらに異なる入力で AMD を実行するといった繰返しが必要となる．そこで，本研究ではこれらのルールを自動的に求める手法を提案する．

3. ルール抽出

本章では，戦略的操作不可能なオークションメカニズムのルールを自動的に抽出する手法を説明する．オークションメカニズムにとって，戦略的操作不可能であることは意思決定の方法において社会的に望ましいため，戦略的操作不可能性の制約を導入したオークションメカニズムに限定して議論する．

本提案手法は，自動メカニズムデザインが小規模な問題しか適用できないことに対し，一般的な状況に適用可能なルールを求める手法である．本手法によって得られたルールは任意の状況に適用可能であり，参加者の数やタイプに限定されない．一般的な状況に適用可能であることは，客観的に検証可能な必須の条件であり，簡潔であることは副次的な条件である．

ここでは，一般的なルールを抽出するために，オークションメカニズムにおけるしきい値に着目する．しきい値は他のエージェントの申告を固定した場合に一意に定まる，つまり，θ_{-i} を入力とする関数として表すことができる．エージェントはそのしきい値を超える値を申告することで，しきい値を支払い，財を獲得できることが知られている．提案手法は，あるエージェント i がバンドル B_i を獲得するために必要なしきい値を AMD の出力から推定する．ここで，本論文で扱うしきい値の候補関数を定義する．

- $ss_{N\setminus\{i\}}$：エージェント i を除く社会的余剰の最大値を指す．
- $ss_{M\setminus B_i}$：エージェント i が欲しがるバンドル B_i を除くときの社会的余剰の最大値を指す．
- $os_{s,t}$：s 個の財を持つ異なるバンドルの中で t 番目に大きな評価値を指す．たとえば，$\{g_1\}$ に 3，$\{g_1\}$ に 2，$\{g_2\}$ に 1，$\{g_1, g_2\}$ に 4，$\{g_2, g_3\}$ に 5 という入札が存在するとき，$os_{1,1} = 3$，$os_{1,2} = 1$，$os_{2,1} = 5$，$os_{2,2} = 4$ とする．
- 上記の候補関数の定数倍した関数．
- 任意の 2 つの候補関数の線形和・線形差の関数．たとえば，$ss_{N\setminus\{i\}} - ss_{M\setminus B_i}$ は VCG メカニズム [15] の支払い規則を示している．

しきい値の候補関数は，ルールの条件の候補関数と同じものを利用している．ルールの条件およびしきい値の候補関数の選択は重要であり，これらが異なれば得られるルールは異なり，適切な候補が与えられていなければ，望ましいルールが得られない可能性がある．現状では，既存のメカニズムを解析し，できる限り網羅的に候補関数を与えている．本論文では，目的関数を社会的余剰の期待値の最大化に限定して議論しているが，支払額最大化や最悪実行時における社会的余剰の比率の最大化等の常識的な目的関数の範囲であれば，現状の候補関数で十分であると考えている．

次節以降で説明するルール抽出フローは以下のような流れになる．まず，AMD を通して出力されたオークション結果を読み込み，設定した候補関数を用いて結果の集合の要素にラベルを付ける．ラベルを付けた結果それぞれに対して，各エージェント i がバンドル B_i を獲得するためのしきい値の候補範囲（上限と下限）を求める．次に，求めたしきい値の上限と下限から，集合被覆アルゴリズムを用いて，目的となるルールを抽出する．

3.1 ルール抽出フロー

本節では，オークション結果を読み込んでからあるエージェント i がバンドル B_i を獲得するためのしきい値を求めるまでのフローを紹介する．使用するオークション結果は AMD を通して出力されたオークション結果を用いることとする．

まず，AMD によるオークション結果の集合 $\mathcal{O} = \{o^1, \ldots, o^u\}$ を読み込む．各 o^k $(k \in \{1, \ldots, u\})$ は，エージェントの申告したタイプ θ と割当て結果 a を情報として持つ．割当て結果 a は，各エージェントにどの財が割り当てられたかという情報を持つ．すなわち，エージェント i が θ_i を申告したときの i に割り当てられるバンドルを $X_i(\theta_i, \theta_{-i})$ としたとき，$a = \{X_1(\theta_1, \theta_{-1}), \ldots, X_i(\theta_n, \theta_{-n})\}$ である．ここで，$\mathcal{O}$ から，エージェント i がバンドル B_i に入札しているオークション結果 o^k のみを抜き出し，その集合を $\mathcal{O}'$ とする．

次に $\mathcal{O}'$ に含まれるオークション結果 o^k にラベル付けを行う．o^k が持つエージェントのタイプ θ に着目し，i 以外のエージェントのタイプの組 θ_{-i} が一致するオークション結果 o^k に同一のラベル $l \in \{1, \ldots, L\}$ を付ける．一致するものがないときは，その o^k は単独のラベルを持つことになる．ラベル l を付けられたオークション結果が持つ i を除くタイプの組 θ_{-i} を θ^l_{-i} と表す．

ラベル付けの後，各ラベル l について，ラベル l を付けられた結果におけるエージェント i のしきい値の候補範囲 $CV^l \subseteq \mathbb{R}^2$ を求める．この候補範囲は，上限値 $\overline{cv}^l \in \mathbb{R}$ と下限値 $\underline{cv}^l \in \mathbb{R}$ によって，$CV^l = [\underline{cv}^l, \overline{cv}^l]$ と規定される．$\overline{cv}^l$ は，ラベル l を付けられた結果において（すなわち，i 以外のエージェントの入札が θ^l_{-i} である場合に），エージェント i がバンドル B_i を獲得できる最小の入札額を表す：

$$\overline{cv}^l = \min\{\theta_i \in \Theta_i | X_i(\theta_i, \theta_{-i}) = B_i\}$$

一方 $\underline{cv}^l$ は，ラベル l を付けられた結果において，エー

ジェント i がバンドル B_i を獲得できない最大の入札額を表す：

$$\underline{cv}^l = \max\{\theta_i \in \Theta_i | X_i(\theta_i, \theta_{-i}) = \emptyset\}$$

このとき，戦略的操作不可能性より，i が B_i を獲得するためのしきい値は，範囲 $[\underline{cv}^l, \overline{cv}^l]$ の中に存在する．

ここで，各候補範囲 CV^l について，あらかじめ定義された有限の候補関数の集合 $\mathcal{F} = \{f_1, f_2, \ldots\}$ のそれぞれを評価する．具体的には，すべてのラベル $l \in \{1, \ldots, L\}$ に対して

$$f_j(B_i, \theta^l_{-i}) \leq \overline{cv}^l \tag{1}$$

を満たし，少なくとも 1 つのラベル $l \in \{1, \ldots, L\}$ に対して

$$f_j(B_i, \theta^l_{-i}) \geq \underline{cv}^l \tag{2}$$

を満たす候補関数 f_j を記憶していく．このとき，式 (1) および (2) を満たす候補関数 f_j に，式 (2) が成立するラベルの集合を $S_j \subseteq \{1, \ldots, L\}$ として保持させるものとする：

$$S_j = \{l \in \{1, \ldots, L\} | f_j(B_i, \theta^l_{-i}) \in CV^l\}$$

式 (1) および (2) を満たす候補関数 f_j が保持するラベルの集合 S_j によって，ラベルの集合 $\{1, \ldots, L\}$ を被覆する最小被覆問題を解き，最小被覆に用いられた候補関数の集合を $\mathcal{G}$ とする．この問題を解くための集合被覆アルゴリズムについては次節で説明する．ここで得られた候補関数の集合 $\mathcal{G}$ を用いて，i 以外のエージェントのタイプの組が θ_{-i} の場合に，エージェント i がバンドル B_i を獲得するためのしきい値 $cv(B_i, \theta_{-i})$ は次式になる：

$$cv(B_i, \theta_{-i}) = \max_{f_j \in \mathcal{G}} f_j(B_i, \theta^l_{-i})$$

3.2 集合被覆アルゴリズム

本節では，前節で述べた最小被覆問題と，最小被覆問題を解く集合被覆アルゴリズムについて説明する．被覆する頂点の全体集合を $\mathbb{L} = \{1, 2, \ldots, L\}$，$\mathbb{L}$ の部分集合の族を $\mathcal{S} = \{S_1, S_2, \ldots\}$ とする．最小被覆問題とは，この $\mathbb{L}$ の要素をすべて被覆する，コスト最小の $\mathcal{S}$ の部分族を選ぶ問題である．ただし本研究では，$\mathbb{L}$ の各部分集合 S_j のコストは 1 である（$c(S_j) = 1$）とし，最小数で $\mathbb{L}$ を被覆する $\mathcal{S}$ の部分族を探す問題として定義する．

本研究では，貪欲法に基づく集合被覆アルゴリズムを用いて，この最小被覆問題を解く．集合被覆アルゴリズムはまず，すでに被覆されている要素の集合を $\mathcal{C}$ とし，$\mathcal{C}$ の初期状態を $\emptyset$ と定める．次に，$\frac{c(S_j)}{|S_j - \mathcal{C}|}$ が最小となる候補関数 f_j を探し，S_j を被覆する．すなわち，$\mathcal{C} \leftarrow \mathcal{C} \cup S_j$ となる．ただし，最小となる候補関数が複数存在する場合，抽出されるルールは簡潔であることが重要なため，可能な限り，線形和・線形差や定数倍を選ばないようにしなければならない．そこで本研究では，各候補関数，各候補関数の定数倍，各候補関数どうしの線形和・線形差の順番で優先順位をつけている．$\mathcal{C} = \mathbb{L}$ となるまで以上の手順を繰り返し，最後に，選ばれた候補関数 f_i の集合を出力する．ここでの候補選択基準はヒューリスティックなものであるため，これが正しいという厳密な基準が存在しない．そのため，ある解が求まった後に候補選択の優先順位を変化させて別途検証する必要がある．

例 1． 3 人 2 財の設定のもと，以下の目的関数と制約条件で実行した AMD の結果に対する，ルール抽出のフローを説明する．

目的関数： 社会的余剰の期待値の最大化

制約条件： 戦略的操作不可能性

また，エージェント 0 はバンドル $\{g_1, g_2\}$，エージェント 1 はバンドル $\{g_1\}$，エージェント 2 はバンドル $\{g_2\}$ にそれぞれ入札を行うものとする．このとき，AMD の出力結果の一部が表 1 である．

$\{g_1, g_2\}$ に入札したエージェント 0 のしきい値を求める．エージェント 0 以外のタイプの組を θ_{-0} として，表 1 よりエージェント 1 が $\{g_1\}$ に 3 を，エージェント 2 が $\{g_2\}$ に 2 を入札している結果を選び，ラベル $l_1 \in \mathbb{L}$ とする．つまり，o_1，o_2 をラベル l_1 とする．AMD の結果のうち，ラベル l_1 を付けられた結果に注目すると，エージェント 0 は $\{g_1, g_2\}$ に 6 で入札をすれば $\{g_1, g_2\}$ を獲得できるが，4 で入札を行えば獲得できない，という結果であった．この結果から，しきい値の候補範囲 $CV^{l_1} = [\underline{cv}^{l_1}, \overline{cv}^{l_1}]$ は，以下のように計算される：

$$CV^{l_1} = [4, 6]$$

その他のラベルに関しても同様の手順を繰り返し，しきい値の候補範囲を求める．

候補関数の全体集合 $\mathcal{F}$ の中から，式 (1) および (2) を満たす候補関数の集合 $\mathcal{F}^*$ をあげていく．このとき，候補関数の 1 つである $ss_{N \setminus \{i\}} - ss_{M \setminus B_i}$（VCG の支払い規則）は，すべてのラベルについて式 (1) および (2) を満たす（たとえばラベル l_1 については，$(3 + 2) - 0 = 5 \in CV^{l_1}$ である）．ここであげた，式 (1) および (2) を満たす候補関数を用いて，集合被覆アルゴリズムによるラベルの集合被覆を行う．ラベルの集合被覆を視覚化したものを**図 1** に示す．円はそれぞれのオークションの結果を表し，点線の楕

Algorithm 貪欲法に基づく集合被覆アルゴリズム

```
1: C ← ∅, G ← ∅
2: while C ≠ L do
3:     choose f_j s.t. S_j ∈ argmin_{f_j∈F} c(S_j)/|S_j−C|     // c(S_j)/|S_j−C| が最小となる f_j を選ぶ
4:     C ← C ∪ S_j, G ← G ∪ f_j                              // S_j を被覆し，f_j を記憶する
5: end while
6: return G
```

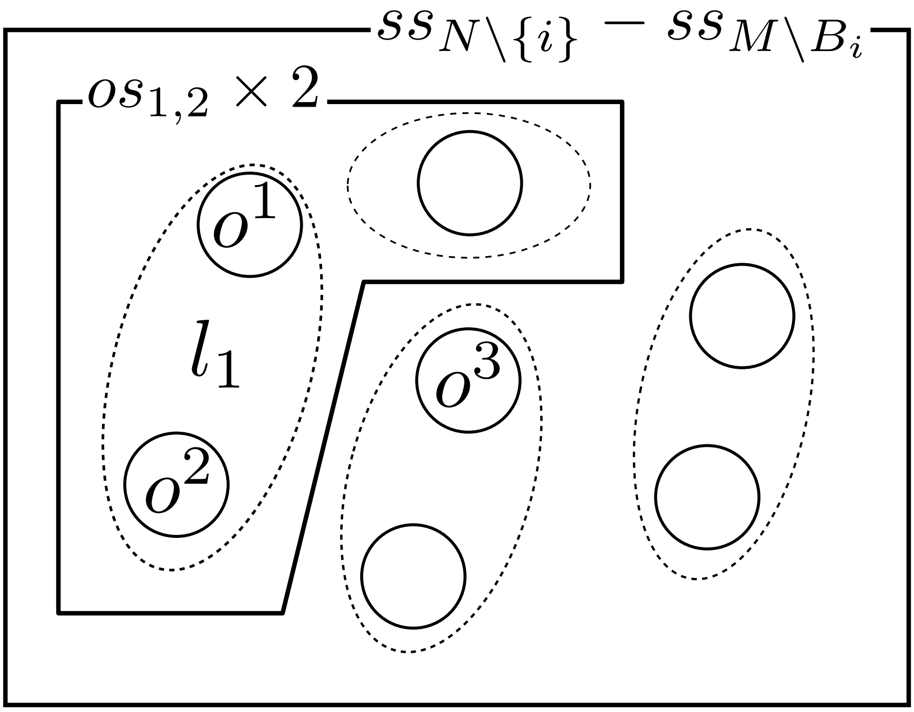

図 1 エージェント 0 におけるラベルの集合被覆
Fig. 1 Set cover of labels of agent 0.

円はラベル分けした状態を表している．実線によって囲まれたものは候補関数によって被覆された範囲を表す．図 1 より，このとき，候補関数である $ss_{N\backslash\{i\}} - ss_{M\backslash B_i}$ はすべてのラベルを被覆しており，$os_{1,2} \times 2$ は 2 つのラベルを被覆していることが分かる．集合被覆アルゴリズムでは，$\frac{c(S_j)}{|S_j - \mathcal{C}|}$ が最小となる候補関数を探す．すなわち，まだ被覆されていないラベルを最も多く被覆できる候補関数を探すので，$ss_{N\backslash\{i\}} - ss_{M\backslash B_i}$ が選ばれる．よって，集合被覆アルゴリズムは出力結果 $\mathcal{G}$ として $ss_{N\backslash\{i\}} - ss_{M\backslash B_i}$ のみからなる候補関数の集合を返す．これは，VCG メカニズムの支払い規則と同値であることから，エージェント 0 が $\{g_1, g_2\}$ を獲得するためのしきい値は VCG メカニズムの支払い規則で与えられることが分かる．

$\{g_1\}$ に入札したエージェント 1 のしきい値を求める．エージェント 1 以外のタイプの組を θ_{-1} として，エージェント 0 が $\{g_1, g_2\}$ に 6 を，エージェント 2 が $\{g_2\}$ に 2 を入札しているものを選び，ラベル $l_2 \in \mathbb{L}$ とする．つまり，o_2，o_3 をラベル l_2 とする．AMD の結果のうち，ラベル l_2 を付けられた結果に注目すると，エージェント 1 は $\{g_1\}$ に 5 で入札をすれば $\{g_1\}$ を獲得できるが，3 で入札を行えば獲得できない，という結果であった．よって，しきい値の候補範囲 CV^{l_2} は，以下のように計算される：

$$CV^{l_2} = [3, 5]$$

その他のラベルに関しても同様の手順を繰り返し，しきい値の候補範囲を求める．

このとき $ss_{N\backslash\{i\}} - ss_{M\backslash B_i}$ は，エージェント 0 の場合と同様に，すべてのラベルについて式 (1) および (2) を満たす（たとえばラベル l_2 については，$6 - 2 = 4 \in CV^{l_2}$）．集合被覆アルゴリズムを適用すると，出力結果 $\mathcal{G}$ として $ss_{N\backslash\{i\}} - ss_{M\backslash B_i}$ のみですべてのラベルが被覆できると返す．したがってエージェント 1 が財 $\{g_1\}$ を獲得するためのしきい値は $ss_{N\backslash\{i\}} - ss_{M\backslash B_i}$，すなわち VCG メカニズムの支払い規則で与えられる．エージェント 2 については，準匿名性および入札と財の対称性の仮定より，同様に VCG メカニズムの支払い規則のみですべてのラベルが被覆できる．すなわち，この AMD によって構成されたメカニズムは VCG メカニズムと同じものであることが分かる．

例 2. 次に 3 人 2 財の設定のもと，以下の目的関数と制約条件で実行した AMD の結果に対する，ルール抽出のフローを説明する．

目的関数：社会的余剰の期待値の最大化

制約条件：戦略的操作不可能性，架空名義操作不可能性

例 1 と異なり，制約条件に架空名義操作不可能性を加えている．その他の設定は変わらず，エージェント 0 はバンドル $\{g_1, g_2\}$，エージェント 1 はバンドル $\{g_1\}$，エージェント 2 はバンドル $\{g_2\}$ にそれぞれ入札を行うものとする．**表 2** は各エージェントの申告したタイプと財の割当て結果の一部である．

$\{g_1, g_2\}$ に入札したエージェント 0 のしきい値を求める．エージェント 0 以外の入札 θ_{-0} として，表 2 よりエージェント 1 が $\{g_1\}$ に 4 を，エージェント 2 が $\{g_2\}$ に 2 を入札しているものを選び，ラベル $l_3 \in \mathbb{L}$ とする．つまり，o_1，o_3 をラベル l_3 とする．AMD の結果のうち，ラベル

表 2 架空名義操作不可能性を制約として追加した場合の AMD の出力結果
Table 2 Output of AMD constrained with false-name-proofness.

AMD の結果	エージェントのタイプ			財の割当て		
	0	1	2	0	1	2
o^1	$\{g_1, g_2\}$ - 3	$\{g_1\}$ - 4	$\{g_2\}$ - 2	$\emptyset$	$\{g_1\}$	$\{g_2\}$
o^2	$\{g_1, g_2\}$ - 5	$\{g_1\}$ - 2	$\{g_2\}$ - 3	$\{g_1, g_2\}$	$\emptyset$	$\emptyset$
o^3	$\{g_1, g_2\}$ - 5	$\{g_1\}$ - 4	$\{g_2\}$ - 2	$\{g_1, g_2\}$	$\emptyset$	$\emptyset$
o^4	$\{g_1, g_2\}$ - 5	$\{g_1\}$ - 4	$\{g_2\}$ - 3	$\emptyset$	$\{g_1\}$	$\{g_2\}$
o^5	$\{g_1, g_2\}$ - 5	$\{g_1\}$ - 6	$\{g_2\}$ - 2	$\emptyset$	$\{g_1\}$	$\emptyset$
o^6	$\{g_1, g_2\}$ - 7	$\{g_1\}$ - 4	$\{g_2\}$ - 3	$\{g_1, g_2\}$	$\emptyset$	$\emptyset$
$\vdots$		$\vdots$			$\vdots$	

l_3 を付けられた結果に注目すると，しきい値の候補範囲 $CV^{l_3} = [\underline{cv}^{l_3}, \overline{cv}^{l_3}]$ は，以下のように計算される：

$$CV^{l_3} = [3, 5]$$

次に，エージェント 1 が $\{g_1\}$ に 4 を，エージェント 2 が $\{g_2\}$ に 3 を入札しているものを選び，ラベル $l_4 \in \mathbb{L}$ とする．つまり，o_4，o_6 をラベル l_4 とする．AMD の結果のうち，ラベル l_4 を付けられた結果に注目すると，しきい値の候補範囲 $CV^{l_4} = [\underline{cv}^{l_4}, \overline{cv}^{l_4}]$ は，以下のように計算される：

$$CV^{l_4} = [5, 7]$$

その他のラベルに関しても同様の手順を繰り返し，しきい値の候補範囲を求める．

候補関数の全体集合 $\mathcal{F}$ の中から，式 (1) および (2) を満たす候補関数をあげていく．このとき，$os_{1,1}$ に関してはラベル l_3 を満たすがラベル l_4 は満たしていない（ラベル l_3 については，$os_{1,1} = 4 \in CV^{l_3}$ だが，ラベル l_4 については，$os_{1,1} = 4 \notin CV^{l_4}$）．一方で，$os_{1,2} \times 2$ はラベル l_4 を満たす（ラベル l_4 については，$os_{1,2} \times 2 = 6 \in CV^{l_4}$）．

ここであげた，式 (1) および (2) を満たす候補関数を用いて，集合被覆アルゴリズムによるラベルの集合被覆を行う．このとき，集合被覆アルゴリズムは出力結果 $\mathcal{G}$ として $os_{1,1}$，$os_{1,2} \times 2$，$os_{2,1}$ からなる候補関数の集合を返す．したがってエージェント 0 が $\{g_1, g_2\}$ を獲得するためのしきい値は $\max\{os_{1,1}, os_{1,2} \times 2, os_{2,1}\}$ で与えられる．

$\{g_1\}$ に入札したエージェント 1 のしきい値を求める．エージェント 1 以外のタイプの組を θ_{-1} として，エージェント 0 が $\{g_1, g_2\}$ に 5 を，エージェント 2 が $\{g_2\}$ に 2 を入札しているものを選び，ラベル $l_5 \in \mathbb{L}$ とする．つまり，o_3，o_5 をラベル l_5 とする．AMD の結果のうち，ラベル l_5 を付けられた結果に注目すると，しきい値の候補範囲 CV^{l_5} は，以下のように計算される：

$$CV^{l_5} = [4, 6]$$

次にエージェント 0 が $\{g_1, g_2\}$ に 5 を，エージェント 2 が $\{g_2\}$ に 3 を入札しているものを選び，ラベル $l_6 \in \mathbb{L}$ とする．つまり，o_2，o_4 をラベル l_6 とする．ラベル l_6 に関して，しきい値の候補範囲 CV^{l_6} は，以下のように計算される：

$$CV^{l_6} = [2, 4]$$

その他のラベルに関しても同様の手順を繰り返し，しきい値の候補範囲を求める．

このとき，集合被覆アルゴリズムを適用すると，すべてのラベルを被覆できないと出力された．これは**図 2** から，ラベル l_5 を被覆できる候補関数が存在しないことが問題となっている．たとえば，$CV^{l_5} = [4, 6]$ を満たすような候補関数として $os_{2,1}$ があげられる（ラベル l_5 に関して，$os_{2,1} = 5$ である）．しかし，候補関数として記憶するためには，すべてのラベルにおいて式 (1) を満たさなければならない．ここで，ラベル l_6 に注目すると，$os_{2,1} < \overline{cv}^{l_6}$ を満たしていない（ラベル l_6 に関して，$os_{2,1} = 5$ であり，$\overline{cv}^{l_6} = 4$ である）．したがって，$os_{2,1}$ は候補関数の集合に選ばれないのである．他の候補関数も同様の手順を踏むと，ラベル l_5 を被覆できる候補関数が存在しない．

4. 条件分岐

AMD を通して出力されたオークション結果によっては，しきい値が推定できない場合があることを例 2 によって示した．そこで，ラベルの集合 $\mathbb{L} = \{1, \ldots, L\}$ を複数に分割してしきい値を求める手法，条件分岐を提案する．条件分岐では，ラベルの全体集合 $\mathbb{L}$ を入力とし，しきい値の集合 $\mathcal{CV}$ を出力する．以下に条件分岐を用いたアルゴリズムのフローを紹介する．

まず，3.1 節に従い，しきい値を求める．もし，ラベルの全体集合 $\mathbb{L}$ に対して被覆できないラベルが存在するならば $\mathbb{L}$ を大きく 2 つの集合に分割する．具体的には，任意の 2 つの候補関数 $f, g \in \mathcal{F}$ を選び，$f > g$ となるラベルの集合を X とし，$f \leq g$ のラベルの集合を Y とする．f，g は，候補関数のうち，より優先順位の高いものから選ばれる．このとき，$X \cup Y = \mathbb{L}$，$X \cap Y = \emptyset$ である．

集合 X について集合被覆アルゴリズムを適用する．もし被覆できないラベルが存在したならば f，g を選び直し，

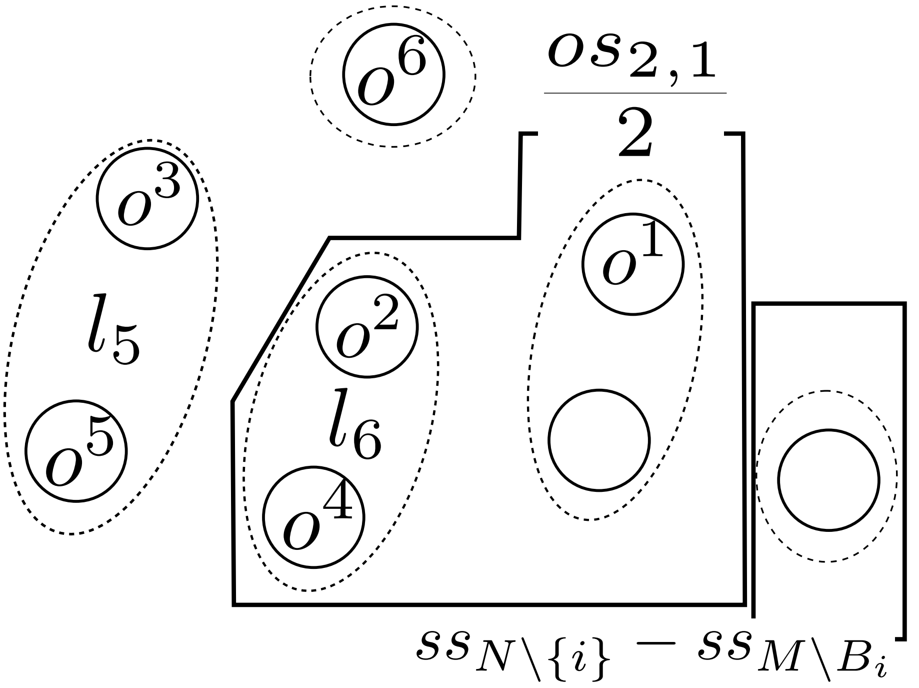

図 2 エージェント 1 におけるすべてのラベルを被覆できない例
Fig. 2 Example that not all labels of agent 1 is covered.

Algorithm 条件分岐

$\mathcal{CV} \leftarrow \emptyset, \mathcal{G} \leftarrow \emptyset$
$\mathcal{G} \leftarrow$ 集合被覆アルゴリズム $(\mathbb{L})$ //しきい値を受け取る
if $\mathcal{G} \neq \emptyset$ **then**
  $\mathcal{CV} \leftarrow \mathcal{CV} \cup \{\max_{f_l \in \mathcal{G}} f_l\}$ //しきい値を記憶する
else
  $\mathcal{G}' \leftarrow \emptyset, cv_X \leftarrow \emptyset, \mathcal{CV}^Y \leftarrow \emptyset$
  choose $f, g \in \mathcal{F}$ //任意の 2 つの候補関数を選ぶ
  $X \leftarrow \{\mathbb{L} | f > g\}$ //$f > g$ を満たすラベルを記憶する
  $Y \leftarrow \mathbb{L} \setminus X$ //$f > g$ を満たさないラベルを記憶する
  $\mathcal{G}' \leftarrow$ 集合被覆アルゴリズム (X) //X に関するしきい値を受け取る
  if $\mathcal{G}' = \emptyset$ **then**
    go to 6 //被覆できない場合，f, g を選び直す
  else
    $cv_X \leftarrow \{\max_{f_l \in \mathcal{G}'} f_l\}$
    $\mathcal{CV} \leftarrow \mathcal{CV} \cup \{cv_X\}$ //X に関するしきい値を記憶する
    $\mathcal{CV}^Y \leftarrow$ 条件分岐 (Y) //Y に関するしきい値の集合を受け取る
    $\mathcal{CV} \leftarrow \mathcal{CV} \cup \mathcal{CV}^Y$
  end if
end if
return $\mathcal{CV}$

ラベルの集合 $\mathbb{L}$ を集合 X と集合 Y に再度分ける．X に関するすべてのラベルを被覆できたならば，X におけるしきい値 cv_X を保持する．

次に集合 Y に関して，条件分岐アルゴリズムを再帰的に行うと，出力結果としてしきい値の集合である $\mathcal{CV}^Y$ が返ってくる．このとき，条件分岐アルゴリズムは，しきい値の集合 $\mathcal{CV} = \{cv_X\} \cup \mathcal{CV}^Y$ を出力する．

例 3. 例 2 のエージェント 1 が $\{g_1\}$ を獲得する場合について説明する．なお，例 2 のエージェント 0 については，集合被覆アルゴリズムのみですべてのラベルを被覆できるため，ここではエージェント 1 に関してのみ説明する．

まず，集合被覆アルゴリズムに適用し $\{g_1\}$ に入札したエージェント 1 のしきい値を推定する．しかしこのとき，集合被覆アルゴリズムではすべてのラベルを被覆できない．

そこで，ラベルの集合を大きく 2 つに分ける．たとえば $os_{2,1} > os_{1,1} \times 2$ を満たすラベルの集合を X とし，それ以外のラベルの集合を Y とする．このとき，**図 3** のようにラベルの集合を分けることができる．具体的には，l_5 は X に含まれ，l_6 は Y に含まれることになる．

X に関して，候補関数の集合 $\mathcal{F}$ の中から，式 (1) および (2) を満たす候補関数をあげていき，集合被覆アルゴリズムによるラベルの集合被覆を行う．よって集合被覆アルゴリズムは出力結果として $os_{2,1}$ を返す．これはラベルの集合 X に含まれるすべてのラベルが候補関数 $os_{2,1}$ のみで被覆できることを意味している（ラベル l_5 に関して，$os_{2,1} = 5$ から，$CV^{l_5} = [4, 6]$ を満たす）．

次に Y に関して，候補関数の集合 $\mathcal{F}$ の中から，式 (1) および (2) を満たす候補関数をあげていき，集合被覆アルゴリズムによるラベルの集合被覆を行う．集合被覆アルゴリズムは出力結果として $os_{2,1}/2$ と $ss_{N\setminus\{i\}} - ss_{M\setminus B_i}$ を返す．これはラベルの集合 Y に含まれるすべてのラベルが候補関数 $os_{2,1}/2$ と $ss_{N\setminus\{i\}} - ss_{M\setminus B_i}$ の 2 つの関数で被覆できることを意味している（ラベル l_6 に関して，$os_{2,1}/2 = 2.5$ から，$CV^{l_6} = [2, 4]$ を満たす）．

したがって，エージェント 1 が $\{g_1\}$ を獲得するための

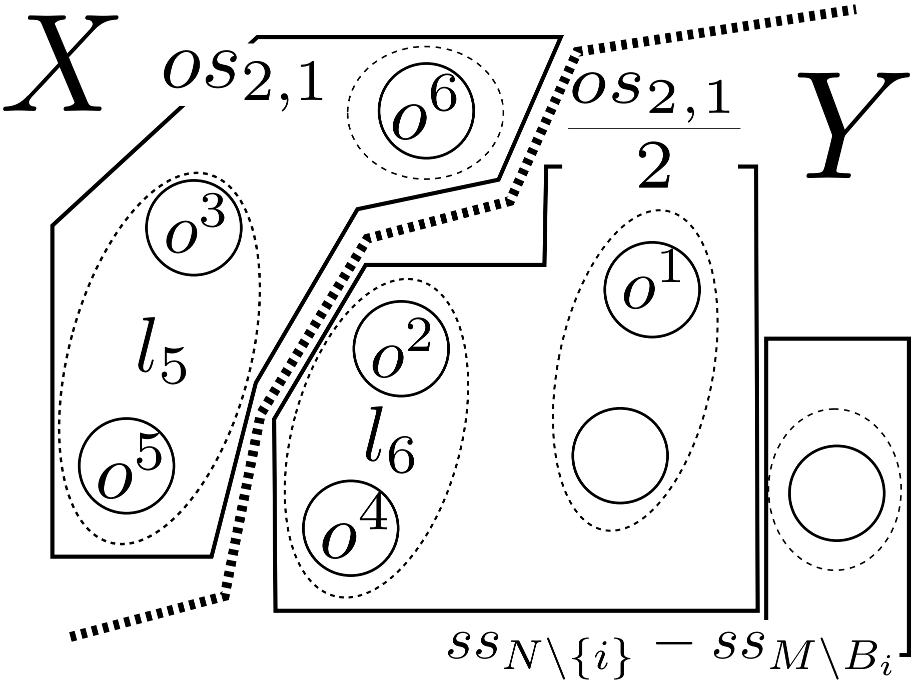

図 3　条件分岐を利用してラベルを被覆した例
Fig. 3　Example that cover all labels by using conditional branches.

しきい値は，$os_{2,1} > os_{1,1} \times 2$ のとき $os_{2,1}/2$ で与えられ，$os_{2,1} \leq os_{1,1} \times 2$ のとき $\max\{os_{2,1}/2, ss_{N\backslash\{i\}} - ss_{M\backslash B_i}\}$ で与えられる．エージェント 2 については，準匿名性および入札と財の対称性の仮定より，同様のしきい値が与えられる．この AMD によって構成されたメカニズムは，3 人 2 財の組合せオークションにおいて適応的留保価格（adaptive reserve price，ARP）メカニズム [5] と等価である．

本提案手法により既存のメカニズムが再発見できたことは，本手法がまったく新しいメカニズムを発見するためには有望であることを示唆していると考えている．

計算時間に関しては，たとえば，3 人 2 財 10 タイプの架空名義操作不可能性を制約とした自動メカニズムデザインの出力結果から提案手法でルールを得る際の実行時間は，Intel(R) Core(TM)2 Quad CPU Q9650@3.00 GHz，16.0 GB RAM を搭載した PC において 1 分とかからなかった．本手法は，自動メカニズムデザインの結果からのルール抽出に関する最初の提案であるため，比較対象となる他の手法が，従来の人手による方法しか存在しない．また，人手による手法は個人のスキルに依存するところが大きく，場合によっては，どれだけ長い時間がかかろうとも適切なルールは抽出できない．3 人 2 財 10 タイプにおける自動メカニズムデザインが出力する項目はおよそ 1,000 通りとなるが，これだけでも人手で解析するにはなかなか困難であり，さらに人数や財の数が増加したならば，人手による解析は現実的に不可能である．したがって，本手法が有効であると考えている．

5.　結論

本論文では，自動メカニズムデザインと発見科学における法則発見の技術を組み合わせることで，大規模な問題に適用可能なルールを抽出する手法を提案した．提案手法は，小規模な問題に対する自動メカニズムデザインによるオークション結果を読み込み，法則発見を繰り返し実行することで，あるエージェント i の財を獲得するためのしきい値を求めることができる．本手法を用いることで，3 人 2 財の組合せオークションにおいて，戦略的操作不可能なメカニズムとして理論的に優れた性質を持つ VCG メカニズム，そして架空名義操作不可能なメカニズムとして適応的留保価格メカニズムが出力された．既存のメカニズムを出力できたことから，この手法はメカニズムを設計するうえで有効であるといえる．

本手法の利点として，異なる制約条件や目的関数のもとでも利用可能であるということがあげられる．これまでの検討は，制約条件として戦略的/架空名義操作不可能性，目的関数として社会的余剰の最大化という，すでに従来研究が多数存在する場合に関して行っていたため，まったく新規なメカニズムを発見することが困難であったと考えている．これに対して，従来研究が少ない問題設定，たとえば，無羨望性といった制約条件や，売り手の収入最大化といった目的関数を考慮した場合に，まったく新しいメカニズムが発見できる可能性が高いと考えている．

しかしながら，提案手法を用いることで，つねに望ましいルールを得ることを保証することは不可能である．メカ

ニズムデザインの理論では，数多くの不可能性定理が示されており，いくつかの制約条件を同時に満たす一般的なメカニズムが存在しえないことが分かっている．たとえば，パレート効率性と架空名義操作不可能性を同時に満たす一般的なメカニズムは存在しない．このような状況で本手法を適用した場合，与えられた特殊な状況には対応可能なルールが得られるが，それを一般化することは本質的に不可能となる．また，実際には望ましいルールが存在する場合でも，候補関数が不足している場合には適切なルールが得られない可能性がある．そこで今後の課題としては，新たな候補関数の提案，条件分岐に利用する条件の提案があげられる．

メカニズムを表す関数は，不連続，非線形なものであり，従来の科学的法則発見の分野で扱われてきた法則と比較すると，複雑で理論的に扱いにくい形式である．そのため，既存の法則発見の技術を集団合意形成ルールの自動設計に適用可能なように拡張することは挑戦的な課題であり，2つの独立な研究領域をつなぐことで，これらの研究分野のさらなる活性化を期待できる．

謝辞 本研究の遂行にあたり，日本学術振興会科学研究費補助金基盤研究（A）（課題番号 20240015）の助成を受けました．ここに深く感謝いたします．また，非常に有益なコメントをくださった電子情報通信学会情報・システムソサイエティ人工知能と知識処理（AI）の2名の査読者に深く感謝いたします．

参考文献

[1] Conitzer, V. and Sandholm, T.: Automated mechanism design: Complexity results stemming from the single-agent setting, *Proc. 5th International Conference on Electronic Commerce* (*ICEC*), pp.17–24 (2003).

[2] Cramton, P., Shoham, Y. and Steinberg, R. (Eds.): *Combinatorial Auctions*, MIT Press (2005).

[3] de Vries, S. and Vohra, R.V.: Combinatorial auctions: A survey, *INFORMS Journal on Computing*, Vol.15, No.3, pp.284–309 (2003).

[4] 福田剛志，森本康彦，徳山 豪：データマイニング，共立出版 (2001).

[5] Iwasaki, A., Conitzer, V., Omori, Y., Sakurai, Y., Todo, T., Guo, M. and Yokoo, M.: Worst-case efficiency ratio in false-name-proof combinatorial auction mechanisms, *Proc. 9th International Joint Conference on Autonomous Agents and Multiagent Systems* (*AAMAS*), pp.633–640 (2010).

[6] 森下真一，宮野 悟：発見科学とデータマイニング，共立出版 (2001).

[7] 元田 浩，津本周作，山口高平，沼尾正行：データマイニングの基礎，オーム社 (2006).

[8] 毛利貴之，東藤大樹，岩崎 敦，横尾 真：架空名義操作不可能な組合せオークションメカニズム：VCGメカニズムの改良，情報科学技術フォーラム講演論文集，Vol.9, No.2, pp.51–58 (2010).

[9] 大森由総，斎藤恭昌，岩崎 敦，横尾 真：自動メカニズムデザインによる架空名義入札に頑健な組合せオークションメカニズムの構築，情報科学技術フォーラム講演論文集，Vol.7, No.2, pp.399–402 (2008).

[10] ポール・ミルグラム：オークション 理論とデザイン，東洋経済新報社 (2007).

[11] 坂井豊貴，藤中裕二，若山琢磨：メカニズムデザイン—資源配分制度の設計とインセンティブ，ミネルヴァ書房 (2008).

[12] Sandholm, T.: Automated mechanism design: A new application area for search algorithms. *Proc. 9th International Conference on Principles and Practice of Constraint Programming* (*CP*), pp.19–36 (2003).

[13] 杉町勇和，毛利貴之，岩崎 敦，横尾 真：混合整数計画法による自動メカニズムデザインの設計と高速化，日本オペレーションズ・リサーチ学会 2011 年秋季研究発表会 (2011).

[14] Todo, T., Li, R., Hu, X., Mouri, T., Iwasaki, A. and Yokoo, M.: Generalizing envy-freeness toward group of agents, *Proc. 22nd International Joint Conference on Artificial Intellegence* (*IJCAI*), pp.386–392 (2011).

[15] Varian, H.: Economic mechanism design for computerized agents, *Proc. 1st Usenix Workshop on Electronic Commerce*, New York (1995).

[16] 横尾 真：オークション理論の基礎—ゲーム理論と情報科学の先端領域，東京電機大学出版局 (2006).

毛利 貴之 （学生会員）

2010 年 3 月九州大学工学部電気情報工学科卒業．2012 年 3 月同大学大学院システム情報科学府修士課程修了．アルゴリズム的ゲーム理論やメカニズムデザインに関する研究に興味を持つ．第 10 回情報科学技術フォーラム（FIT 2011）船井ベストペーパー賞受賞．

杉町 勇和 （学生会員）

2011 年 3 月九州大学工学部電気情報工学科卒業．現在，同大学大学院システム情報科学府修士課程在籍中．アルゴリズム的ゲーム理論やメカニズムデザインに関する研究に興味を持つ．

東藤 大樹　（正会員）

2008 年 3 月九州大学工学部電気情報工学科単位取得退学．2010 年 3 月同大学大学院システム情報科学府修士課程修了．現在，同大学院システム情報科学府博士後期課程在籍中．2010 年 4 月より日本学術振興会特別研究員 DC1．アルゴリズム的ゲーム理論や計算論的社会選択理論，メカニズムデザインに関する研究に興味を持つ．人工知能学会学生会員．

岩崎 敦　（正会員）

2002 年神戸大学大学院自然科学研究科博士課程修了．同年より 2004 年まで NTT コミュニケーション科学基礎研究所に勤務．2004 年より九州大学大学院システム情報科学研究院助教．ゲーム理論，学習，オークション，実験経済学に関する研究に従事．博士（学術）．人工知能学会会員．

横尾 真　（フェロー）

1984 年東京大学工学部電子工学科卒業．1986 年同大学大学院修士課程修了．同年 NTT に入社．1990～1991 年ミシガン大学客員研究員．2004 年より九州大学大学院システム情報科学研究院教授．マルチエージェントシステム，制約充足問題に関する研究に従事．エージェントの合意形成メカニズム，制約充足/分散制約充足等に興味を持つ．博士（工学）．1992 年，2002 年人工知能学会論文賞，1995 年情報処理学会坂井記念特別賞，1999 年，2005 年人工知能学会全国大会優秀論文賞，2004 年 Association for Computing Machinery（ACM）Special Interest Group on Artificial Intelligence（SIGART）Autonomous Agent Research Award，2005 年ソフトウェア科学会論文賞，2006 年学士院学術奨励賞，2010 年人工知能学会業績賞，International Foundation for Autonomous Agents and Multiagent Systems influential paper award 受賞．人工知能学会，日本ソフトウェア科学会，電子情報通信学会，AAAI 各会員．