

LF-009

## 短時間近似が可能な組み合わせオークション勝者決定アルゴリズムの評価

## An Evaluation of Short-time Approximation Algorithm on Combinatorial Auctions

福田 直樹 † 伊藤 孝行 ‡  
Naoki Fukuta Takayuki Ito

## 1 はじめに

組み合わせオークションとは、入札の対象として単一の財に対してだけでなく、複数の財の組み合わせに対して入札が可能なオークションである [1]。組み合わせオークションには、すでに電子商取引で広く普及している単一財を対象としたオークションなどに置き換わる新たなオークションメカニズムとして、広く普及する可能性がある。Sandholm らによる例 [2] の他、米国 FCC による周波数帯域割り当てへの適用が検討された例も報告されている [3]。同時に、他の多くの複雑な組み合わせ最適化問題の近似にも適用できることが明らかにされつつある。たとえば、チリにおける給食の配分効率化問題への適用事例が報告されている [4]。

組み合わせオークションの勝者決定問題に対しては、最適解を高速に求めるアルゴリズムについての研究が進みつつある [5][6][7][1] が、この問題は NP-hard であることが知られており [1]、特に入札数の増加に対して爆発的に計算が複雑になる。たとえば、CASS アルゴリズム [7] では、我々が試した限りでは、1つのオークションに対する入札数がおおよそ 3000 を超えると、最適解を求めることが非常に困難になる。一方で、組み合わせオークションにおける入札数は、代替可能財を扱おうとすると、爆発的に増加してしまう。さらに、エージェント(プログラム)による入札の自動生成を考えると、入札者の持つ選好関数などからそれを満たす(人手では到底不可能な)大量の入札を作り出すことができるため、エージェントが参加可能な組み合わせオークションでは、非常に大量の入札を前提とした高速な勝者決定が求められる。

組み合わせオークションにおける高速な最適解の探索手法と、それによる組み合わせオークションの様々な有益な性質を保証することが非常に重要であることは明らかである。しかし、我々は、組み合わせオークションをアプリケーションシステムに適用する場合に、勝者決定問題に対して解の最適性を厳密に保証することが計算資源的に難しいのなら、現実的な計算時間で質の高い近似解を求めたうえで、その解に対してある一定の好ましい性質を持たせられるようにするアプローチもありえると考えている。そこで、我々はこれまでに、組み合わせオークションの勝者決定に関する新しい近似アルゴリズム [8] と、それが持つべき好ましい性質 [9] についての解析および考察を行ってきた。

本論文では、組み合わせオークションで非常に多数の入札が行われた際における、勝者決定の近似解法について、これまでに我々が提案した手法と既存の近似手法との比較を行い、特に計算時間が短時間に限られる状況下においての我々の手法の有効性を示す。

## 2 組み合わせオークションと勝者決定問題

## 2.1 組み合わせオークション

組み合わせオークションとは、入札者に対して、単一の財に対してだけでなく、複数の財の組み合わせに対して入札を行えるようにしたオークションメカニズムである [1]。この組み合わせ

オークションでは、勝者決定問題は次のように定義される [1]: 入札者の集合を  $N = 1, \dots, n$ , 財全体の集合を  $M = \{1, \dots, m\}$ .  $|M| = k$  とする。財のパンドル  $S$  は  $S \subseteq M$  を満たす任意の集合である。ここで、 $v_i(S)$  は、入札者  $i$  によって財のパンドル  $S$  になされた入札の入札額を示す。真実申告最良性が成り立つとき、 $v_i(S)$  は、入札者  $i$  が財のパンドル  $S$  に対して支払うことができる最大の金額となる。財の入札者へ割り当ては  $x_i(S) \in \{0, 1\}$  で示し、 $x_i(S) = 1$  のとき入札者  $i$  が財のパンドル  $S$  を落札できたものとする。ここで、ある財の割り当て  $x_i(S)$  が適している (feasible) とは、その財の割り当てが、ただ 1つの入札に対してのみ行われることであり、次のように表現できる:

$$\forall j \in M \sum_{i \in N} \sum_{S \ni j} x_i(S) \leq 1 \quad (1)$$

ここで、勝者決定問題とは、feasible な財の割り当て  $X \ni x_i(S)$  に対して、その落札合計額を最大化する問題

$$\max_X \sum_{i \in N, S \subseteq M} v_i(S) x_i(S) \quad (2)$$

として表せる。

## 2.2 Lehmann アルゴリズム

組み合わせオークションの勝者決定問題は計算論的に複雑度の高い問題であるため、その近似解法についての研究もなされている。勝者決定問題の近似解法としてよく知られたものに、Lehmann らによって提案されたアルゴリズム [10] がある。

Lehmann のアルゴリズムは、比較的単純な、欲張り (greedy) アルゴリズムであり、その計算量は (事前に行われるソーティングを除けば) 入札数に対して線形オーダーに近い性質を示す。ここで、ある入札を  $b = \langle a, s \rangle$  (ただし、 $s \subseteq M$  and  $a \in \mathcal{R}_+$ ) と表現することにする。また、2つの入札  $b = \langle a, s \rangle$  および  $b' = \langle a', s' \rangle$  が競合するのは、 $s \cap s' \neq \emptyset$  のときであるとする。このとき、Lehmann アルゴリズムは次のように表せる。

1. 入札は、ある基準によって事前にソートされる。文献 [10] では、直感的に表現すると、入札のリスト  $L$  を、財あたりの入札額の平均値によって降順でソートする方法が提案されている。これを、より一般化したものとして、文献 [10] では、入札のリスト  $L$  を、 $a/|s|^c$  の値によってソートする方法を提案している。ここで、 $c$  の値は  $c \geq 0$  の任意の値であり、全体の財の個数  $k$  に関連させるとよいことが指摘されている。
2. ソートされた入札のリスト  $L$  に対して、欲張りアルゴリズムにより財が割り当てられる。財の割り当ては、単純に入札リスト  $L$  を前方から見ていき、その入札の財のパンドルがまだ未割り当てであり、かつ他の (これまでに割り当てられ勝者となった) 入札と競合しないとき、その入札を勝者としてを繰り返し行う。

文献 [10] では、ソート時のパラメータ  $c$  について、 $c = 1/2$  とすると良好な結果が得られると述べられている。そこで、本論文では、以後は特に断りがない限り  $c = 1/2$  を  $c$  の標準値として話を進める。

## 2.3 Zurel らの近似法

Zurel らは、文献 [11] で、組み合わせオークションにおける勝者決定の高速な近似解法を提案している。Zurel らの手法は、

† 静岡大学, Shizuoka University

‡ 名古屋工業大学, Nagoya Institute of Technology

線形計画法に基づく近似アルゴリズムと、ランダムな逐次改善に基づく解の洗練の組み合わせとして構成される。

最初のフェーズでは、主双対内点法の一種を用いるが、このとき、それぞれの財  $i$  に対する価格幅  $p_i$  を調整するようになっている。その結果として、各入札内の各財  $i$  に対する落札比 (fractional allocation) を  $0 \leq A_i \leq 1$  の範囲で割り当て、同時に、財ごとの価格  $p_j$  が割り当てられる。そこで、各入札  $v_i$  の入札する財のバンドル  $S_j$  に対して、値  $v_i / \sum_{i \in S_j} p_j$  に基づいて降順に並べ替えが行われ、Lehmann アルゴリズムと同様な greedy な割り付けに基づいて、初期勝者集合が決定される。

初期勝者集合が得られた後は、逐次的に、ランダムな更新が試みられる。ここでは、その段階で勝者となっていない入札の中からランダムに1つの入札が選択され、先ほどの入札集合の並べ替え順序の先頭に移される。その順序に基づいて改めて greedy に勝者決定を行い、以前の勝者集合の場合よりも総落札価格が上昇した場合には、その勝者集合が新たな解となり、そのときの入札の並び順が、次のランダムな更新の際に基準として用いられる。この逐次的なランダム更新プロセスは、勝者集合が更新されなくなるまで繰り返される。本手法の詳細については、文献 [11] に詳しく述べられている。

## 2.4 Casanova

Hoos らは、文献 [12] で、汎用の random walk 型の SAT ソルバーが、組み合わせオークションの勝者決定問題の近似に対しても高い性能を発揮することを報告している。文献 [12] では、Casanova アルゴリズムという名前で、そのアルゴリズムの詳細が述べられている。Casanova アルゴリズムは、状態空間を勝者入札集合で表現し、その評価値を、落札された財の数に対する総落札価格の値 (総落札価格をそのとき落札された財の数で割ったもの) で表現する。Casanova アルゴリズムを直感的に表現すると、初期状態は勝者が空であり、そこに、もっとも評価値の高い入札を勝者として順に追加 (およびその際に衝突している入札を削除) していくことで、勝者を近似していく。Casanova アルゴリズムでは、random walk 型の探索を実現するために、もっとも評価値の高い入札が確率  $1 - wp$  で選択され、確率  $wp$  ではランダムに1つの (勝者でない) 入札が選択されるようになっている。また、勝者に加える入札の「目新しさ」を考慮し、その入札が、評価値がそのときの2番目の入札と比較して、最後に勝者集合に加えられるから間もない入札であったときには、確率  $1 - np$  の割合で2番目に評価値の高い入札を勝者として選択するようになっている。さらに、いわゆる soft restart 機能も用いており、解の質が向上しない状況が長く続いた場合か、あるいは事前に設定されたループ回数  $maxSteps$  に到達した場合に、その計算プロセス一度打ち切り、最初から再スタートさせるようになっている。Casanova アルゴリズムは、再スタート回数が事前に設定された  $maxTrials$  を越えた段階で停止し、それまでに得られた最良の解が、近似解として採用される。

## 3 Lehmann アルゴリズムの改善

### 3.1 山登り探索による解の改善

Lehmann アルゴリズムでは、財の割り当て結果が下界となる場合があるが、この下界は条件によっては (とくに財の個数  $k$  が非常に大きい場合) かなり低くなり、実際にアルゴリズムを適用した場合でも最適解に対して 50% 前後の解が出てくるなど、結果のばらつきが多く見られる。我々は、Lehmann アルゴリズムによる解を、山登り探索によって洗練することができるのではないかと考えた [8]。本節では、Lehmann アルゴリズムによって得られた財の割り当てを初期解として、その初期解を山登り探索により漸次的に洗練する手法について述べる。山登り探索では、現在の解の近傍を探索し、現在の解よりも良いものがあればそれを選んで次の解とする、というプロセスを、新しい解が見つからなくなるまで繰り返す。ここで重要となるのは、

その解の近傍をどのように定義するかである。ここで、我々は、解の近傍として、単一入札距離 (single bid distance) を用いる。

ある2つの解が単一入札距離であるとは、ある解 (勝者となる入札の集合) と、別の解との距離が、その解に含まれない入札を1つ追加することでできること (ただし、その追加による競合入札の削除と、そこで競合しない範囲での新規入札の解への追加として、複数の入札の増減を含んでもいい) である。

次のアルゴリズムは、単一入札距離での山登り探索に基づく、近似勝者決定アルゴリズムである。

入力となるのは、入札のリスト  $L$  と 初期解  $Alloc$  であり、ここでは、それぞれ Lehmann アルゴリズムで用いられた入札のリストと、Lehmann アルゴリズムで得られた解である。なお、5 から 9 行目までが、単一入札距離による近傍を求める部分である。

```

1: function LocalSearch(Alloc, L)
2:   RemainBids := L - Alloc;
3:   for each  $b \in$  RemainBids as sorted order
4:     if  $b$  conflicts Alloc then
5:       Conflicted := Alloc - consistentBids( $\{b\}$ , Alloc);
6:       NewAlloc := Alloc - Conflicted +  $\{b\}$ ;
7:       ConsBids :=
8:         consistentBids(NewAlloc, RemainBids);
9:       NewAlloc := NewAlloc + ConsBids;
10:    if price(Alloc) < price(NewAlloc) then
11:      return LocalSearch(NewAlloc, L);
12:   end for each
13:   return Alloc

```

関数 “consistentBids” は、“RemainBids” にある入札を順に辿って、内部に競合入札のない新しい解 “NewAllocn” を見つける。これは、“Alloc” に対して、新たな入札を挿入し、その入札に競合する入札を解から取り除くことで行われる。競合していた入札を解から取り除いた後は、競合を起こさずに解に追加できる入札が存在する場合がある。その場合には、競合を起こさない範囲で、リストのソート順が上位のものから可能な限りの財が落札できるように入札を解に挿入する。

具体例: 5つの財  $a, b, c, d$ , および  $e$  と、6つの入札、 $\langle 30, \{a, b, c\} \rangle, \langle 15, \{a\} \rangle, \langle 13, \{c\} \rangle, \langle 15, \{d, e\} \rangle, \langle 14, \{a, c\} \rangle, \langle 8, \{b\} \rangle$  があるとすると、これらの入札に対する Lehmann の評価値  $a/\sqrt{a}$  は、それぞれ 17.6, 15, 13, 10.7, 10, および 8 となる。ここで、Lehmann アルゴリズムによる初期解は、 $\langle 30, \{a, b, c\} \rangle, \langle 15, \{d, e\} \rangle$  であり、その総落札額は 45 である。今、解となっていない入札は  $\langle 15, \{a\} \rangle, \langle 13, \{c\} \rangle, \langle 14, \{a, c\} \rangle, \langle 8, \{b\} \rangle$  である。本アルゴリズムでは、このうち、評価値の最上位である  $\langle 15, \{a\} \rangle$  を選び、これを初期解に挿入する。 $\langle 15, \{a\} \rangle$  と競合する入札である  $\langle 30, \{a, b, c\} \rangle$  は、ここで解から取り除く。現時点での解は  $\langle 15, \{a\} \rangle, \langle 15, \{d, e\} \rangle$  である。次に、この解と競合しない範囲で、他の入札を解に挿入することを試みる。その結果、解は  $\langle 15, \{a\} \rangle, \langle 8, \{b\} \rangle, \langle 13, \{c\} \rangle, \langle 15, \{d, e\} \rangle$  となる。ここでの総落札額は 51 である。総落札額が、初期解より向上しているため、 $\langle 15, \{a\} \rangle, \langle 8, \{b\} \rangle, \langle 13, \{c\} \rangle, \langle 15, \{d, e\} \rangle$  を新たな初期解とする。本アルゴリズムは、この解の更新操作を、新たに総落札額が向上する解が見つからなくなるまで繰り返すことにより、Lehmann アルゴリズムによる解を少しずつ洗練していくことができる。

### 3.2 複数値のソート因子 'c' に対する並列探索による改善

Lehmann アルゴリズムによる近似解は、入札リストをソートするときの因子である  $c$  の値の影響を強く受ける。

Lehmann らは、 $c = 1/2$  が、理論的な解の最適性の下界をもっとも高く保証でき、かつ経験的にもよい値であることを報告し



ている．ところが、我々のこれまでの検討結果 [9] から、各オークション問題ごとの  $c$  の値による解の最適性をみると、入札数と財の数が同数であっても、試行ごとに最適となる  $c$  の値がばらつくことがある．すなわち、試行ごとに最適な  $c$  の値を見つけることで、結果を改善できる可能性があることが予想される．本節では、前節までに提案したアルゴリズムに対して、複数の  $c$  の値に対して並列に探索をさせることで、解の最適性を改善することを試みる．

本アルゴリズムでは、Lehmann アルゴリズムで用いる  $c$  の値は、 $C = \{0.0, 0.1, \dots, 1.0\}$  から適切な個数を選ぶこととした．これは、 $1/2$  を中心として、十分な数の  $c$  の値に対して探索を行うのに十分であると考えられるからである．異なる  $c$  に対する探索結果は 1 つにまとめられ、最終的な解としては、それらのうちでもっともよいものが取られる．本論文では、実行時並列度の低い状況下での制約された計算時間内の性能も考慮して、 $C = \{0, 0.5, 1\}$  の 3 つの  $c$  の値を用いることとした．

#### 4 比較実験

本論文では、比較に用いたアルゴリズムを以後は次のように表記する．“greedy( $c=0.5$ )” は、Lehmann のアルゴリズムを ( $c = 0.5$ ) で実行した結果である．“HC( $c=0.5$ )” は、 $c = 0.5$  の条件下で、山登り探索手法によって、Lehmann アルゴリズムによる結果を洗練したものである．“greedy-3” および “HC-3” で始まるものは、それぞれ、Lehmann のアルゴリズムあるいは山登り探索手法による洗練 ( $c = \{0, 0.5, 1\}$ ) の 3 つのパラメータについて行い、そのうちで最良となる結果を選んだものである．それぞれ、後ろに “-seq” がついたものは 3 つのパラメータに対して計算を単一処理スレッドとして逐次的に行った場合の結果を示し、“-para” がついたものは 3 つのパラメータに対して計算を完全に並列に行った場合の理論値を示す．それぞれについて、実験時に設定した計算打ち切り時間 (“-100ms”, “-1000ms”) をアルゴリズムの表記の末尾に付記している．“Zurel-1st” は、Zurel らのアルゴリズムで得られる初期解を示す．“Zurel” は、Zurel らのアルゴリズムが停止するまで計算を行わせ、最終的に得られた結果を示す．“casanova” で始まるものは、Hoos が提案した Casanova アルゴリズムによる結果を示し、その後ろに計算打ち切り時間 (“-10ms”, “-100ms”, “-1000ms”) を付記している．

組み合わせオークション問題の生成手法については、Leyton-Brown らの CATS [13] がよく知られている．CATS を用いた数万以上の多数の入札を含むオークションテストデータの作成は、それ自体に非常に計算時間がかかる．本論文では、入札数 20,000、財数 256 の条件で、CATS の標準パラメータを用い、現実的な時間で生成可能であった入札分散 (L2, L3, L4, L6, L7) のみに対して、CATS を用いて各入札分散ごとに 100 試行分のテストデータを準備した．なお、ここで用いた入札分散は、過去に様々な論文内で提案された入札の人工的生成方法をほぼ網羅しており、入札分散の表記は、Leyton-Brown らの文献 [13] による．各入札分散の標準パラメータや入札生成アルゴリズムの詳細については、Leyton-Brown らの文献 [13] を参照されたい．

表 1 に、各近似アルゴリズムにおける計算時間と近似解の最適性の関係を示す．実験環境には、Mac OS X 10.4.9, CPU: CoreDuo 2.0GHz, 2GBytes memory のラップトップ型コンピュータを用いている．表 1 中のそれぞれの値は、左側が近似解の最適性 (Zurel 比) を、右側の括弧内の数値がそれに要した計算時間 (ミリ秒) を、それぞれ示している．なお、入札数が 3,000 を越える場合には、CASS [7] や CPLEX 等の高速に動作することが知られている最適解探索アルゴリズムであっても、その最適解はおろか最大落札額を推定することすらきわめて困難となることが知られている [7]．本論文では、アルゴリズム間の相対的な性能比較を行うことを目的としているため、近似解の最適性を示す値として、各入札分散ごとに各アルゴリズムでの総落

札価格の平均値を計算し、Zurel らの手法による値を 1 として正規化したものを提示している．

L2 から L7 までのいずれの種類の入札分散の場合でも、Casanova アルゴリズムによる近似は、本実験で扱った計算打ち切り時間内では、他のアルゴリズムによる近似結果より大きく劣っている．これは、Casanova アルゴリズムの基盤となっている random-walk アルゴリズムが、本論文で扱うような、ごく短時間での近似には向いていないことを示している．

入札分散が L3 の場合を除き、“greedy-3” および “HC-3” の各種法は、Zurel 手法での初期解 (“Zurel-1st”) の計算時間よりも短い時間で、Zurel 手法の初期解よりも良い近似解を求めることができる．このことから、非常に限られた計算時間内での近似勝者決定には、そのオークションがごく特殊な入札分散を持つ場合を除き、我々の提案する手法が非常に有効であることがわかる．

ほぼすべての種類の入札分散で、“greedy” に関する結果よりも “HC” に関する結果のほうが近似性能が高くなっている．これは、ごく短時間であっても山登り方による解の洗練の効果があることを示している．

入札分散が L4, L6 および L7 のときには、“greedy” と “HC” のいずれの場合でも複数の  $c$  の値に対する並行探索のほうが近似解が良い結果となっている．特に、入札分散が L4 および L6 の場合には、“HC-3-seq-100ms” の近似解の質は、その内部処理が並列処理でなく逐次処理であるにもかかわらず、その 10 倍の時間をかけて単一の  $c$  の値に対して計算した “HC( $c=0.5$ )-1000ms” のものよりも良くなっている．このことから、我々の提案している、複数の  $c$  の値に対する並行探索の効果がこの状況下では非常に高いことがわかる．

入札分散が L3 のときにこの効果が出ないのは、入札分散 L3 では、その入札生成アルゴリズム自体がすべての入札に対して財の数を固定の同一値とするため、 $c$  の値を変化させても入札に対する評価値が変化しないためである．入札分散 L2 のときには、“greedy” および “HC” では、ほとんどのケースで最適解と思われる解が初期に見つかってしまうため、 $c$  の値の違いによる差が大きくは出にくかったものと思われる．

“HC-3” における処理の並列度による違いを見ると、当然ながら “HC-3-seq” よりも “HC-3-para” のほうが、同一の計算打ち切り時間ではより多くの CPU リソースを利用できるため、近似解が良い結果となっている．

#### 5 関連研究

組み合わせオークションの勝者決定問題に対しては、CASS [7] を含め、その最良解を高速に探索するためのアルゴリズムが多く検討されてきている [3]．これらの最良解探索アルゴリズムの中には、近似解探索アルゴリズムとしても転用可能なものがある．CASS は、最良解を求めるためのアルゴリズムであるが、最良解を求める過程では高速な近似解探索アルゴリズムとしても利用できる．文献 [12] および [11] において、Casanova アルゴリズムおよび Zurel らの手法は、CASS と比較して、近似解探索アルゴリズムとしては非常に高い近似性能を発揮することが報告されている．したがって、本論文では、Casanova アルゴリズムおよび Zurel らの手法を比較対象にすれば十分であると判断し、本論文での比較実験では CASS を比較対象から省略した．

Dobzinski らは、文献 [14] で、入札者に対する劣モジュラ性を前提とした状況下での、勝者決定の近似アルゴリズムを提案している．また、Lavi らは、文献 [15] で、線形計画法に基づくアルゴリズムを提案し、それが古典的な VCG (ビクレー組み合わせオークション) に拡張可能であることを示している．これらの研究は、いずれも理論的な側面からの貢献を目的としており、本論文で示すような実装上の高速性や多数入札時における近似性能の比較実験は示されていない．

表1 近似時間と近似の最適性 (20,000 入札, 256 財, Zurel=1 に正規化)

	L2		L3		L4		L6		L7	
greedy(c=0.5)	1.0002	(23.0)	0.9639	(19.0)	0.9417	(23.0)	0.9389	(23.4)	0.7403	(22.1)
greedy-3-seq	1.0003	(69.1)	0.9639	(59.2)	0.9999	(72.9)	0.9965	(67.8)	0.7541	(66.8)
greedy-3-para	1.0003	(26.4)	0.9639	(20.9)	0.9999	(28.4)	0.9965	(26.0)	0.7541	(25.5)
HC(c=0.5)-100ms	1.0004	(100)	0.9741	(100)	0.9576	(100)	0.9533	(100)	0.8260	(100)
HC-3-seq-100ms	1.0004	(100)	0.9692	(100)	1.0000	(100)	0.9966	(100)	0.8287	(100)
HC-3-para-100ms	1.0004	(100)	0.9743	(100)	1.0001	(100)	0.9969	(100)	0.9423	(100)
HC(c=0.5)-1000ms	1.0004	(1000)	0.9856	(1000)	0.9771	(1000)	0.9646	(1000)	1.0157	(1000)
HC-3-seq-1000ms	1.0004	(1000)	0.9804	(1000)	1.0003	(1000)	0.9976	(1000)	1.0086	(1000)
HC-3-para-1000ms	1.0004	(1000)	0.9856	(1000)	1.0006	(1000)	0.9987	(1000)	1.0240	(1000)
Zurel-1st	0.5710	(11040)	0.9690	(537)	0.9983	(2075)	0.9928	(1715)	0.6015	(1795)
Zurel	1.0000	(13837)	1.0000	(890)	1.0000	(4581)	1.0000	(4324)	1.0000	(3720)
casanova-10ms	0.2583	(10)	0.0069	(10)	0.0105	(10)	0.0202	(10)	0.2577	(10)
casanova-100ms	0.2583	(100)	0.0069	(100)	0.0105	(100)	0.0202	(100)	0.2577	(100)
casanova-1000ms	0.5357	(1000)	0.1208	(1000)	0.0861	(1000)	0.1486	(1000)	0.7614	(1000)

(括弧内の数値はミリ秒単位での計算時間)

Guo らは、文献 [16] で、局所探索に基づく手法で、組み合わせオークションの勝者決定の近似手法を提案している。Guo らの手法は、我々の提案する手法と Casanova アルゴリズムのちょうど中間的なものとなっている。文献 [16] で示された比較実験結果では、Guo らの手法は Casanova アルゴリズムよりも 30% 程度高い解を半分程度の時間で得られると言及されている。しかし、文献 [16] で示された実験条件は本論文で扱ったものよりも問題の大きさが一桁以上小さく、なおかつ実験試行回数も各条件に対してわずか 10 回であるため、同文献中で示された結果の信頼性には疑問がある。また、本論文中の比較実験で示した結果の通り、Casanova アルゴリズム自体が、入札数がある程度多い場合には短時間で近似の性能がそれほど高くないため、文献 [16] で示された結果はそれほど驚くべき結果ではないと考えられる。Casanova アルゴリズムを介した本研究との間接的な比較結果から、我々が提案している手法のほうがより短時間で同程度の近似解を探索できると考えられる。Guo らの手法との直接的な比較は、今後の課題である。

CPLEX は、商用のソフトウェアであり、非常に高速な線形計画法のソルバーとして知られている。ところが、組み合わせオークションの勝者決定の近似に用いる場合には、文献 [11] で、CPLEX と比較して、Zurel らの手法が少なくとも 10 から 100 倍程度は高速であることが言及されている。また、文献 [16] でも、Guo らの手法は、CPLEX と比較して、1/40 から 1/80 の計算時間で同等以上の近似解を得られることが報告されている。これらの報告結果との間接的な比較から、我々の提案している手法は、CPLEX と比較しても十分に高速であると判断できる。

## 参考文献

- [1] P. Cramton, Y. Shoham and R. Steinberg: "Combinatorial Auctions", The MIT Press (2005).
- [2] T. Sandholm, S. Suri, A. Gilpin and D. Levine: "Cabob: A fast optimal algorithm for winner determination in combinatorial auctions", *Management Science*, **51**, 3, pp. 374–390 (2005).
- [3] S. de Vries and R. V. Vohra: "Combinatorial auctions: A survey", *International Transactions in Operational Research*, **15**, 3, pp. 284–309 (2003).
- [4] R. Epstein, L. Henriquez, J. Catalan, G. Y. Weintraub, C. Martinez and F. Espejo: "A combinatorial auction improves school meals in Chile: A case of or in developing countries", *International Transactions in Operational Research*, **11**, pp. 593–612 (2004).
- [5] T. Sandholm: "An algorithm for optimal winner determination in combinatorial auctions", *Proc. of the 16th International Joint Conference on Artificial Intelligence (IJCAI'99)*, pp. 542–547 (1999).
- [6] T. Sandholm, S. Suri, A. Gilpin and D. Levine: "Winner determination in combinatorial auction generalizations", *Proc. of the 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS02)*, pp. 69–76 (2002).
- [7] Y. Fujishima, K. Leyton-Brown and Y. Shoham: "Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches", *Proc. of the 16th International Joint Conference on Artificial Intelligence (IJCAI99)*, pp. 548–553 (1999).
- [8] N. Fukuta and T. Ito: "Towards better approximation of winner determination for combinatorial auctions with large number of bids", *Proc. of The 2006 WIC/IEEE/ACM International Conference on Intelligent Agent Technology (IAT2006)*, pp. 618–621 (2006).
- [9] 福田直樹, 伊藤孝行: "組み合わせオークションにおける多数入札時での勝者決定の近似解法に関する一考察", *電子情報通信学会論文誌*, **J90-D**, 9 (2007). (掲載予定).
- [10] D. Lehmann, L. I. O'Callaghan and Y. Shoham: "Truth revelation in rapid, approximately efficient combinatorial auctions", *Journal of the ACM*, **49**, pp. 577–602 (2002).
- [11] E. Zurel and N. Nisan: "An efficient approximate allocation algorithm for combinatorial auctions", *Proc. of the Third ACM Conference on Electronic Commerce (EC2001)*, pp. 125–136 (2001).
- [12] H. H. Hoos and C. Boutilier: "Solving combinatorial auctions using stochastic local search", *Proc. of the AAAI2000* (2000).
- [13] K. Leyton-Brown, M. Pearson and Y. Shoham: "Towards a universal test suite for combinatorial auction algorithms", *Proc. of EC 2000* (2000).
- [14] S. Dobzinski and M. Schapira: "An improved approximation algorithm for combinatorial auctions with submodular bidders", *SODA '06: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, ACM Press, pp. 1064–1073 (2006).
- [15] R. Lavi and C. Swamy: "Truthful and near-optimal mechanism design via linear programming", *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pp. 595–604 (2005).
- [16] Y. Guo, A. Lim, B. Rodrigues and Y. Zhu: "A non-exact approach and experiment studies on the combinatorial auction problem", *Proc. of HICSS2005* (2005).