

組合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

福田 直樹^{†a)} 伊藤 孝行^{††}

An Analysis about Approximated Allocation Algorithms of Combinatorial Auctions with Large Numbers of Bids

Naoki FUKUTA^{†a)} and Takayuki ITO^{††}

あらまし 本論文では、10,000～100,000を超えるような多数の入札がある組合せオークションに対して、勝者決定の近似解法を提案し、その特性を評価する。提案アルゴリズムでは、従来手法の Lehmann アルゴリズムにおける入札ソート順位の決定因子に着目し、更に山登り法及びシミュレーテッドアニーリング (SA) による解の洗練を行うことで、安定して高い最適性を持った解を得る。入札数が 1000 程度の場合に、本提案手法では最適解に対して平均して 0.997 程度の総落札額が得られる解を見つけれられることを示す。更に、従来の最適解探索手法と比較して十分に高速に動作し、10,000 を超える入札数をもつオークションに対して数秒 (入札数 10,000) から数分程度 (入札数 100,000) の実用的な時間で計算が行えることを示す。本近似解法の限界点の一つとして、本近似解法を用いた組合せオークションが一般には真実申告最良とはならないことを、Monotonicity という条件が成り立たないことから示す。一方で、真実申告最良とはならないものの、本近似解法の一部では、解探索の過程におけるランダム性を排除し順序性を保存することで、近似解に対して特定の好ましい性質を満たすことを示す。

キーワード 交渉と協調, 電子商取引, 組合せオークション

1. ま え が き

組合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でなく、複数の財の組合せに対して入札が可能となるオークションである [1]。組合せオークションには、既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] のほか、米国 FCC による周波数帯域割当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合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。例えば、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]。

組合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 [1], [5]～[7] 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。例えば、CASS アルゴリズム [7] では、我々が試した限りでは、一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、組合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。更に、エージェントによる入札の自動生成ができれば、代替可能財に対する多数の入札も現実的なものとなり得る。

組合せオークションにおける高速な最適解の探索手法と、それによる組合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、組合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

[†] 静岡大学情報学部, 浜松市

Faculty of Informatics, Shizuoka University, 3-5-1 Johoku, Hamamatsu-shi, 432-8011 Japan

^{††} 名古屋工業大学大学院産業戦略工学専攻, 名古屋市

Master Course of Techno-Business Administration, Nagoya Institute of Technology, Gokiso-cho, Showa-ku, Nagoya-shi, 466-8555 Japan

a) E-mail: fukuta@cs.inf.shizuoka.ac.jp

似解を求めた上で、その解に対してある一定の好ましい性質をもたせられるようにするアプローチもあり得ると考えている。

本論文では、組合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、既存手法を更に改良するための近似解法を提案し、その特性について考察する。2. では、基本知識として組合せオークションの概要と Lehmann [8] による近似解法を紹介する。3. では、Lehmann アルゴリズムの拡張手法として、山登り法、Simulated Annealing、及び並列実装の概要と実装方式について提案する。4. では、実験と評価を示す。5. では、提案近似解法のもつ有用性と真実申告最良に関する議論を示す。6. では関連研究を示し、特に Zurel [9] らとの比較実験の詳細を示す。最後に 7. で本論文をまとめる。

2. 組合せオークションと近似解法

2.1 組合せオークション

組合せオークションとは、入札者に対して、単一の財に対してでなく、複数の財の組合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]。この組合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。財のバンドル 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 によって財のバンドル S になされた入札の入札額を示す。すなわち $v_i(S)$ とは、その入札者 i が財のバンドル S に対して支払うことができる最大の金額を意味する。財の入札者へ割当は $x_i(S) \in \{0, 1\}$ 、で示し、 $x_i(S) = 1$ のとき入札者 i が財のバンドル S を落札できたものとする。ここで、ある財の割当 $x_i(S)$ が適している (feasible) とは、その財の割当が、ただ 1 人の入札者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：

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

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

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

として表せる。

2.2 Lehmann アルゴリズム

組合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 [8] 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、二つの入札 $b = \langle a, s \rangle$ 及び $b' = \langle a', s' \rangle$ が競合するのは、 $s \cap s' \neq \emptyset$ のときであるとす。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 [8] では、直感的に表現すると、入札のリスト L を、財当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 [8] 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。財の割当は、単純に入札リスト L を前方から見ていき、その入札の財のバ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、その入札を勝者とするを繰り返す。

Lehmann アルゴリズムでは、その最悪の割当 (下界) が次のように保証されている。

[定理 1] 入札リストをソートするパラメータが $c = 1/2$ のとき、Lehmann アルゴリズムによる近似解は、最適割当に対して最悪 \sqrt{k} となる [8]。

文献 [8] では、ソート時のパラメータ c について、経験的には $c = 1/2$ 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。

具体例：財 x, y 、及び z があり、3 人の入札者 Alice, Bob、及び Charles がいるとする。Alice は財 x に対して 10 を、Bob は財 $\{y, z\}$ に対して 20 を。Charles は財 $\{x, y\}$ に対して 18 をそれぞれ入札するとする。この入札は、 $a/\sqrt{|s|}$ の値によって

降順に並べ替えられる．この値は，*Alice* の入札では $10/\sqrt{1} = 10$ ，*Bob* の入札では $20/\sqrt{2} = (\text{約})14$ ，*Charles* の入札では $18/\sqrt{2} = (\text{約})13$ となる．最終的に，入札のリストは，*Bob* の入札 $\langle 20, \{y, z\} \rangle$ ，*Charles* の入札 $\langle 18, \{x, y\} \rangle$ ，そして *Alice* の入札 $\langle 10, \{x\} \rangle$ という順番に並ぶ．Lehmann アルゴリズムでは，この順に入札を見ていく．最初に，*Bob* が $\{y, z\}$ を 20 で落札できる．次に，*Charles* の入札を見るが，この入札は *Bob* の入札と競合しているため勝者とはならず，最後に *Alice* が $\{x\}$ を 10 で落札してこのアルゴリズムは終了する．

3. Lehmann アルゴリズムの拡張手法

3.1 山登り探索による解の改善

Lehmann アルゴリズムでは，財の割当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（特に財の個数 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 [10]．本節では，Lehmann アルゴリズムによって得られた財の割当を初期解として，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．

ある二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一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  $B \in Alloc$  ( $B$  is a set of bids) then
5:       newAlloc := Alloc -  $B$  +  $b$ ;
6:       consBids :=
7:         consistentBids(newAlloc, remainBids);
8:       newAlloc := newAlloc + consBids;
9:       if price(Alloc) < price(newAlloc) then
10:        return LocalSearch(newAlloc, L);
11:   end for each
12:   return Alloc

```

関数 “consistentBids” は，“remainBids” にある入札を順にたどって，内部に競合入札のない新しい解 “newAllocation” を見つける．これは，“Allocation” 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には，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：五つの財 a, b, c, d ，及び e と，六つの入札， $\langle 30, \{a, b, c\} \rangle$ ， $\langle 15, \{a\} \rangle$ ， $\langle 13, \{c\} \rangle$ ， $\langle 15, \{d, e\} \rangle$ ， $\langle 14, \{a, c\} \rangle$ ， $\langle 8, \{b\} \rangle$ があるとする．これらの入札に対する Lehmann の評価値 $a/\sqrt{|s|}$ は，それぞれ 17.6, 15, 13, 10.7, 10, 及び 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 $\langle 15, \{a\} \rangle$ ， $\langle 13, \{c\} \rangle$ ， $\langle 14, \{a, c\} \rangle$ ，及び $\langle 8, \{b\} \rangle$ 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行くことができる。

3.2 SA (Simulated Annealing) による探索

SA (Simulated Annealing) は、山登り探索をより強化したアルゴリズムとしてよく知られている。前節で提案した山登り探索に基づくアルゴリズムは、SA に基づくアルゴリズムに比較的容易に拡張できる。本節では、SA に基づく Lehmann アルゴリズム解の洗練手法について述べる。

SA に基づく探索では、確率的なランダム動作の度合いを決定するのに、温度 t をパラメータとして用いる。SA では、 t は最初は t_0 であり、探索が進むに従って t を δt ずつ下げていく。探索は山登り法に準ずるが、 $t > 0$ のときには、ある確率関数 $p(t)$ で、解が改善されない場合でもその解を採用するという動作をする。この確率的振舞いにより、SA では探索が局所最適解に陥ることをある程度抑制できる。以下に、SA に基づくアルゴリズムの手順を示す。

入力は、ソートされた入札のリスト L と、Lehmann アルゴリズムによる初期解 *Allocation* である。

```

1:  $t = t_0$ ;
2: function SAsSearch(Allocation,  $L$ )
3:   while  $t > 0$ 
4:      $remainBids := L - Allocation$ ;
5:      $b := randomlySelectedFrom(remainBids)$ .
6:     if  $b$  conflicts  $B \in Allocation$ 
       ( $B$  is a set of bids )
7:     then
8:        $newAllocation := Allocation - B + b$ ;
9:        $consBids :=$ 
10:        consistentBids( $newAllocation, remainBids$ );
11:        $newAllocation := newAllocation + consBids$ ;
12:     if price(Allocation) < price( $newAllocation$ ) then
13:        $Allocation := newAllocation$ ;
14:     else if  $p(t) == 1$  then
15:        $t := t - \delta t$ ;
16:        $Allocation := newAllocation$ ;
17:   return Allocation

```

3.3 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、入札リストをソーティング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、 $c = 1/2$ が経験的によい値であることを報告している。図 1 は、 c を 0 から 2 まで変化

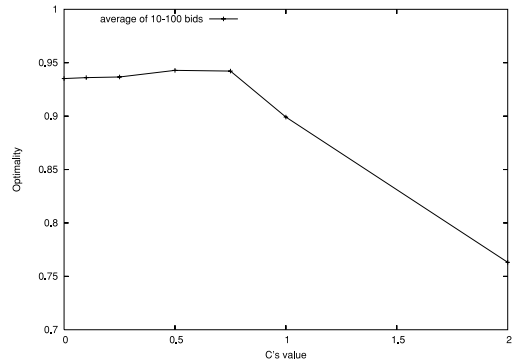


図 1 c の値と最適性の関係

Fig.1 The relation between c 's value and the optimality.

させたときの Lehmann アルゴリズムによる解の最適性を示している。ここでは、財の数は 30 であり、入札数は 10 から 100 までそれぞれ 100 回の試行を行ったものの平均値を示している。図 1 から、Lehmann らの主張通り、平均的にはおおむね $c = 1/2$ のあたりが比較的良好な値を示していることが分かる。各試行ごとの 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 の値はばらついている。このことから、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一つにまとめられ、最終的な解としては、それらのうちで最もよいものがとられる。

本アルゴリズムの手順は次に示すとおりである。本質的に行っていることは、複数の c の値に対して同一の探索アルゴリズムを並列に走らせているだけである。入力は、ソートされていない入札のリスト L と、探索対象とする c の値のリスト C である。

```

1: Function LSForMultiCValues( $L, C$ )
2:    $maxAlloc := \emptyset$ 
3:   while  $|C| > 0$ 
4:      $c$  is selected from  $C$ ;

```

```

5:    $C := C - c;$ 
6:    $Allocation := LocalSearch(L, c);$ 
7:   if  $price(maxAlloc) < price(Alloc)$  then
8:      $maxAlloc := Alloc;$ 
9:   return  $maxAlloc$ 

```

ここでの関数 $LocalSearch(L, c)$ は、ソート因子 c を用いた Lehmann アルゴリズムによる初期解探索後に、山登り探索を行う。Lehmann アルゴリズムによる初期解は山登り探索を行うより前に得られているため、計算速度と解の最適性のバランスをとって、山登り探索を一定の時間制限を付けて途中で打ち切ったり、あるいは Lehmann アルゴリズムによる初期解のみを採用するようにすることも可能である。この場合でも、複数の c に対する解を同時に求めておくことが有効に機能することが期待できる。

SA サーチに関しては、そのランダムな振舞いをする特性を考慮して、次に示す、 c の値をランダムに選択後に探索を行う動作を繰り返し行うアルゴリズムを用意した。

入力となるのは、 L と C である。 L は、入札のリストである。 C は、 c のとり得る値のリストである（ただし、以下では C はグローバル変数として与えられている）。

```

1: Function RandomRestartedSAForMultiCValue( $L$ )
2:    $maxAlloc := \emptyset$ 
3:    $noUpdateCount = 0$ 
4:   while  $noUpdateCount < k$ 
5:      $c$  is randomly selected from  $C$ ;
6:      $Allocation := LehmannAlgorithm(L, c);$ 
7:      $SAAlloc := SASearch(Allocation, L);$ 
8:     if  $price(maxAlloc) < price(SAAlloc)$ 
9:       then
10:         $maxAlloc := SAAlloc;$ 
11:         $noUpdateCount := 0;$ 
12:     else  $noUpdateCount := noUpdateCount + 1;$ 
13:   end while
14:   return  $maxAlloc$ 

```

SA は、ランダム探索アルゴリズムの一種であるので、同一の入力と探索パラメータを与えた場合でも、探索ごとに異なる結果を示すことがある。この特性を考慮し、与えられた各 c の値に対して並列に探索を行うのではなく、ランダムに c を選んで探索を行う操作を、結果が向上しなくなるまで繰り返すようにした。この繰り返しは、最大 k 回解が改善されなければ計算

を打ち切るようにしている。本論文では、 $k = 5$ としている。

3.4 実装

本論文で示したアルゴリズムでは、複数の c の値に対して解の探索を行う部分について、その処理の分散・並列化が比較的容易に実現できる。本論文で示したアルゴリズムは、分散実行を実現する部分の実装にはモバイルエージェントプログラミング環境 $MiLog$ [11] を使い、Lehmann アルゴリズム及び山登り探索等の高速な処理が必要な部分には C 言語による実装を行っている。

4. 実験と評価

4.1 実験条件

前章で提案したアルゴリズムに対し、我々は次の条件で実験を行い、その性能を評価した。

財、入札、入札の対象となる財のバンドルとその入札額は、次の条件でランダムに生成している。一つの入札に含まれる財の個数は、一様分布 $Uniform(1, 10)$ に基づいて、ランダムに選択される。一つの入札に含まれる財のバンドルは、そこでランダムに選択された個数分だけ、一様分布に従って重複がないように、財全体の集合から選択される。各財ごとに基準価格 (Y) を設定しており、その値はオークションごとに一様分布 $Uniform(100, 10100)$ に基づいて決められる。このとき、ある入札の入札額 (X) は、 $X = Z + (Z \times j) \times R$ で与えられる。ここで、 Z は、その入札に含まれる各財に対する基準価格 Y を足し合わせたもので、 j は価格の分散を決める定数パラメータ、 R は、標準ガウス分布（平均 0 分散 1.0）で得られる正規乱数である。なお、本実験では、定数パラメータ $j = 0.1$ としている。なお、本実験での入札のランダム生成では、代替可能財は含めていない。

本実験では非常に大量の入札があるオークションを扱う性質上、そこで起きる現象を現実的に計算可能な範囲内で観測するため、一つのオークションにおける財の数を 30 としている。また、各実験条件に対して、100 回の試行を行い、その平均値をとっている。本実験で用いる解の最適性 (optimality) は、総落札価格を基準とし、それが最適解と一致する場合に 1、全く落札がなかった場合を 0 として、正規化している。本実験では、最適解を求めるために、高速な最適解探索手法の一つである CASS [7] を用いている。

本実験においては、性能に影響を及ぼす部分のプロ

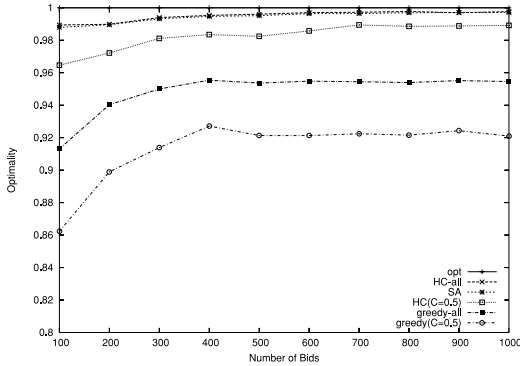


図 2 中規模 (1000 個の入札まで) オークションでの山登り法の最適性

Fig. 2 The optimality of our hill-climbing algorithm in middle-size bids. (up to 1000 bids)

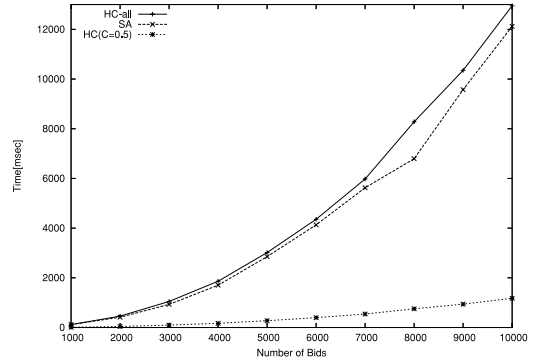


図 3 大規模 (1000 個の入札以上) オークションでの山登り法の計算時間

Fig. 3 The computational time of our hill-climbing algorithm in large-size bids. (greater than 1000 bids)

グラムはすべて C 言語で記述している．実験環境としては，Mac OS X 10.4.6, CPU: CoreDuo 2.0 GHz, 2 GBytes memory のラップトップ型コンピュータを用いている．

4.2 実験結果

本論文では，大きく分けて三つの実験を行っている．各実験では，それぞれ用いたアルゴリズムを次のように表記している．“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．“greedy-all” は，Lehmann のアルゴリズムを ($0 < c < 1$) の範囲で複数回行い，そのうちで最良のものを選んだものである^(注1)．“HC($c=0.5$)” は， $c = 0.5$ の条件下で，山登り探索によって，Lehmann アルゴリズムによる結果を洗練したものである．“HC-all” は，山登り探索による洗練を ($0 < c < 1$) の範囲で行いその最良のものを選んだものである．“SA” は，本論文で述べた，シミュレーテッドアニーリング法を適用した結果である．

図 2 は，各アルゴリズムの基本的な性能を，解の最適性の点から比較したものである．横軸が，一つのオークション当りの入札数であり，100 から 1000 まで変化させている．縦軸が，解の最適性の平均値であり，その算出にあたり，CASS を用いて求めた最適値を用いている．

図 2 から分かるとおり，“HC-all” と “SA” が最も良い値を示しており，その最適性は入札数が 1000 に近づくにつれ大きくなっている．入札数が 1000 あたりでは，解の最適性はおよそ 0.997 である．その次の良い結果が出ているのは，“HC($c = 0.5$)” である．

“HC($c = 0.5$)” では，入札数が増えるに従って解の最適性も上がっているが，“HC-all” や “SA” ほどは上がらず，およそ 0.98 で頭打ちになっている．残りの二つである “greedy($c = 0.5$)” と “greedy-all” については，入札数が 300 前後で解の最適性は上がらなくなっており，その値はおよそ 0.92 と 0.95 である．

図 2 から，本論文で提案している改善手法の一つである，入札をソートする際の因子である c の値を複数とって調べることが，解の最適性を改善することに対して有効に機能していることが読み取れる，“HC-all” は “HC($c = 0.5$)” と比較して解の最適性に優位性が安定して認められる．また，“greedy-all” でも，“greedy($c = 0.5$)” と比較して解の最適性に優位性が安定して認められる．それらの差は，それぞれ 0.1 と 0.4 であり，“HC-all” と “HC($c = 0.5$)” は解の最適性が極めて高い値での比較である点を考えると，十分に効果があったと考えられる．

また，図 2 で示した実験からは，“HC-all” と “SA” の解の最適性に関する差はほとんど見られなかった．“SA” が一般には探索問題に対して極めて優秀な性質を示すことが知られていることから，“HC-all” も，この実験条件では十分な性能を発揮していると考えられる．

図 3 は，最終的な近似解が見つかるまでの平均実行時間を 1 オークション当りの入札数ごとにグラフにしたものである．なお，CASS による最適解探索の時間

(注1): このメカニズムは非常に単純であるが，Lehmann らの論文では言及されていない．

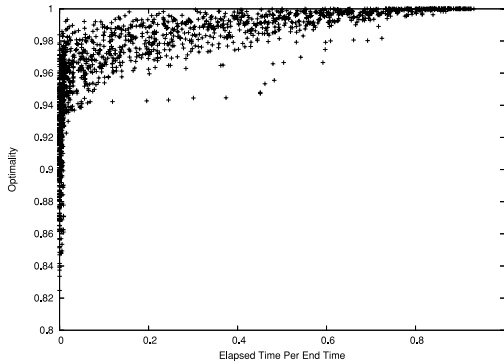


図 4 10,000 個の入札での山登り法の計算時間と解の質の関係

Fig. 4 The relationship between computational time and quality of our hill-climbing algorithm in 10000 bids.

は、入札数が 1000 以下の段階で既に 200,000 msec を超えていたため、図 3 には含めていない。また、図 3 で示した結果は単一処理スレッドによる逐次実行結果であり、並列化・分散化による性能向上分は含まれない。このため、“HC-all” の実行時間は、“HC($c=0.5$)” のおよそ 11 倍になっている。“HC-all” の実行時間は、入札数が 10,000 のときでも 13 秒以内であり、“SA” の結果は“HC-all” よりもわずかに少ない程度となっている。本論文では紙面の都合でグラフは割愛しているが、入札数が 100,000 の場合でも、計算時間は“HC($c=0.5$)”、“HC-all”、“SA” で、それぞれおよそ 55 秒、607 秒、472 秒であった。

図 4 では、“HC-all” について、1 オークション当りの入札数が 10,000 のときの、実行時間内での途中解の品質と時間との関係を示している。すなわち、全体の実行時間を 1 としたときに、実行時間の早い段階でどの程度の最適性をもつ解候補が見つかったかを示している。“HC-all” は、基本的に、漸時的アルゴリズム (anytime algorithm) の一種であるので、計算途中であってもそれまでに見つかった途中解 (漸時解) を提示することができる。図 4 より、ほとんどの場合では、全体の実行時間の 10~20 パーセント程度の中に、0.95 程度の最適性をもった解が見つけられていることが分かる。このことから、“HC-all” では、計算時間に制約を加えて途中で計算を打ち切った場合でも、ある程度満足できる質の結果が得られることが期待できる。

図 5 は、1 オークション当りの入札数が 1,000 から 10,000 までの状況下での、各アルゴリズムで得られ

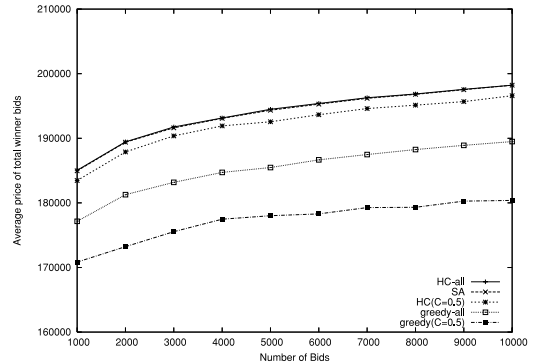


図 5 総落札額の平均値

Fig. 5 The average price of total winners bids.

た解による総落札額の平均値を示している。入札数が 1,000 を超える状況では、最適解を求めることは現実的な時間内では困難なため、ここでは、総落札額の平均を比較することで、各アルゴリズムの性能を比較している。入札数を 10,000 まで増やした場合でも、各アルゴリズム間の性能の差は安定しており、入札数 1000 までと同様の傾向を示している。なお、本論文では見やすさを確保するため図からは割愛しているが、入札数を 100,000 まで増やした場合でも、各アルゴリズム間の性能の差は同様であった。

5. 議 論

5.1 近似解法の限界と満たすべき性質

本節では、近似的勝者決定法が、どのような性質をもてば、実世界のオークション参加者にとって受け入れられ得るか、という点について議論する。

一般に、近似的勝者決定法では、最適解において勝者となるはずの入札者が、近似解において勝者になることができないという現象が当然起こり得る。これは、極端な場合には、オークションの総落札価格が最適値から 0.01 パーセント以下しか離れていないのに、近似解による落札者が最適解による落札者とほとんど一致しないという状況にもなる。ここで、本来落札者となるはずだったものと全く異なる者に財を割り当ててよいのか、というオークションメカニズムの根幹にかかわる問題が生じる。

また、我々の日常世界では、オークション終了後に他の入札者 (特に落札者) の入札価格を知ることができるチャンスがある。最適解法により落札者になり得た入札者が、近似解法により落札者となれなかった事

実が、ごく少数の他者の入札額をオークション終了後に知ること、その入札者自身に分かってしまう場合もある。このことから、単に落札者が近似解法により大きく異なってしまうだけにとどまらず、その事実が、オークション参加者自身にも明確に分かってしまうことにより、特に本来落札者になり得たオークション参加者から強い不満が出ることが予想される。これは、オークション主催者がオークション参加者からの信用を失ってしまう現象を引き起こし得る。

以上の近似勝者決定法がもたらす現象により、近似的勝者決定法がいかに必要とする計算資源の小ささにおいて優れているとしても、実際にはオークション参加者にとって受け入れられにくいのではないかという指摘がある。

我々は、近似的勝者決定法が実世界のオークション参加者に受け入れられ得るためには、以下の(1)、及び(2)を、少なくともある水準で満たす必要があると考える。

(1) 参加する入札者自体も最適解を求めるのが困難である。そのため、入札者自身が得られるごく限られた情報からは最適解を容易に探すことができず、近似手法で得られた解が最適解とどのように異なるのかがオークション参加者には分からない。

(2) 近似解法がもし漸時的計算アルゴリズムであった場合には(1)の性質が計算途中のあらゆる近似解について成り立つ。

我々は(1)の条件を満たすための一つのサブゴールとして、以下に述べる(3)及び(4)の条件が成り立つことが有益であると考え。

(3) 同一の入札(バンドル)について、より低い入札値を提出している入札者が勝つことはない。

(4) 自分の入札しているバンドルを含むバンドル(スーパーセット)に、自分の入札値より小さい入札値を提出している他の入札者が勝つことはない。

(3)は(4)の条件を弱めたものである。本論文では(4)あるいは(3)の状況の起こらない近似勝者決定により行われるオークションを、Admissibleであると呼ぶことにする。

[命題1] (SAを除く)本アルゴリズムで得られた解では、以下の好ましくない状況は起こらない。

(1) (漸時解での Weak Admissible の不成立) アルゴリズム停止以前に得られた解(漸時解)において、同一のバンドルに入札している他の人が、自分より低い入札値で落札している。

(2) (停止解での Admissible の不成立) アルゴリズム停止時に得られた解(停止解)において、自分が入札しているバンドルを含むようなバンドルに入札している他の人が、自分と同じか低い入札値で落札している。

以下の証明では、本論文で提案する近似勝者決定アルゴリズムについて、ランダム性をもつ探索を行うSAを除けば、Admissibleに関連する条件が成立することを示す。具体的には、漸時的計算アルゴリズムとしては、Weak Admissibleを満たし、最終解のみを用いる近似勝者決定アルゴリズムとしては、Admissibleを満たすことを証明する。

(証明) 漸時解に関して、あるバンドル X に対する入札 $b_i(X)$ 及び $b_j(X)$ があり、その落札価格 $v_i(X), v_j(X)$ が $v_i(X) > v_j(X)$ を満たすとき、必ず $b_i(X)$ が落札されることを示す。

Lehmann アルゴリズムでは、ソート順が c の値にかかわらず $b_i(X)$ の方が $b_j(X)$ よりも前になるため、 $b_i(X)$ が勝者にならず $b_j(X)$ が勝者になることはない。すなわち、山登り法による解の洗練の素となる初期解では、必ず Weak Admissible が成り立つ。

次に、山登り法での勝者の更新時での証明を示す。

ある勝者集合 W_n が、ある時点での勝者の集合であると仮定する。

勝者の集合 W_n を更新するきっかけとなる権利をもつ入札 $b_i(X), b_j(X)$ があったとき、その権利を行使することのできる順番は、Lehmann アルゴリズム適用時の入札のソート順位による。ここで、 $v_i(X) > v_j(X)$ であるので、入札 $b_i(X)$ が既に W_n に含まれていない場合には、入札 $b_i(X)$ が常に $b_j(X)$ より優先される。

まず、入札 $b_i(X)$ が既に W_n に含まれていないと仮定する。このとき、入札 $b_i(X)$ により解の更新が可能であった場合、本アルゴリズムでは、 $b_j(X)$ が勝者集合 W_n を更新するチャンスは与えられず、 $b_i(X)$ を含んだ勝者の集合 W_{n+1} が新たな勝者集合となる。

次に、入札 $b_i(X)$ が既に W_n に含まれている場合を考える。このとき、入札 $b_i(X)$ と $b_j(X)$ は同一のバンドルに対する入札なので、 $b_j(X)$ により W_n を更新した結果できる勝者の集合 W'_n は、単純に、 W_n にある、 $b_i(X)$ を $b_j(X)$ に置き換えたものになる。ここで、 $v_i(X) > v_j(X)$ であるので、 W'_n の総落札価格は、 W_n より必ず低くなる。すなわち、 W'_n は、 W_n を更新する解にはなり得ない。

したがって、初期解から最終解(停止解)までのいか

なる解 W_n において, Weak Admissible が成り立つ. 停止解での Admissible に関して, あるバンドル X 及び Y , 及び $b_i(X)$ 及び $b_j(Y)$ があり, $X \subset Y$ かつ $v_i(X) > v_j(Y)$ であるとする. このとき, 必ず $b_i(X)$ が落札されることを示す.

Lehmann アルゴリズム適用時にはソート順が c の値にかかわらず $b_i(X)$ の方が $b_j(Y)$ よりも前になるため, $b_i(X)$ が勝者にならず $b_j(Y)$ が勝者になることはない.

次に, 山登り法での勝者の更新時には, これを背理法で証明する. ある $b_j(Y)$ を含む勝者集合 W_{fin} が最終的な勝者の集合であると仮定する.

勝者の集合 W_{fin} に入札 $b_i(X)$ を挿入した場合, $X \subset Y$ より, $b_i(X)$ は, $b_j(Y)$ とのみ競合する状態である. したがって, W_{fin} の他の入札はこの操作によって, W_{fin} から削除されることはない. すなわち, $b_i(X)$ と $b_j(Y)$ が入れ換わるのみである. ここで, $v_i(X) > v_j(Y)$ であるので, $b_j(Y)$ を $b_i(X)$ で置き換えた勝者の集合 W'_{fin} の方が総価格は高くなる. したがって, W_{fin} は本アルゴリズムで得られる最終的な勝者の集合ではない. これは仮定に矛盾する.

したがって, 停止解 W_{fin} では, Admissible が成り立つ. (以上, 証明終了)

近似勝者決定法において Admissible を成り立たせる方法には大きく二つある. 一つは, 近似解法によって勝者を求める以前にフィルタプログラムによって非支配的な入札をすべて排除してしまい, そこで得られる近似解が必ず Admissible となるように保証する方法である. もう一つは, 我々が提案している近似解法のように, アルゴリズム自身に Admissible を保証する仕組みを組み込むことである. 故に, 他の近似勝者決定法においてもフィルタプログラムを用いることにより Admissible とすることは可能である. アルゴリズム自身に Admissible を保証する仕組みを組み込むことの利点は, フィルタプログラムの計算オーダが $o(n^2)$ であるのに対し, その計算オーダより低いオーダで (初期の) 解を高速に求められる可能性がある点である. 本論文で提案する単純な山登り法アルゴリズム HC($C=0.5$) の場合, 初期の Lehmann 解における計算オーダは $o(n \log n)$ であり, これは $o(n^2)$ よりも小さい. 更に, たいいていの場合, 非支配的な入札はソート順位で下位となるため, 最終的な近似解が求められる直前に参照されるのみで, 計算時間にほとんど影響を及ぼさない. これらの利点についての詳細な

解析は, 本論文の範囲を超えるため, 本論文では割愛する.

5.2 真実申告最良性の不成立について

Lehmann [8] が示すとおり, 真実申告最良な (Truthful) オークションプロトコルが満たすべき最小限の必要十分条件は, Exactness, Monotonicity, Critical, 及び Participation という性質が成り立つことである. 本近似解法を用いたプロトコルは, たとえ単一バンドル選好を仮定したとしても, Monotonicity が成立しないため, 一般には真実申告最良なプロトコルにはならないことを, 本節で示す.

四つの条件のうち, Exactness は, 単一バンドル選好な入札者が自分の望むバンドルを (正確に) 得られるか若しくは何も得られないかのどちらかであることを意味する. すなわち入札者が望まない財を落札するようなことがないことをいう. Monotonicity は, 入札者 j がバンドル s に入札額 v を申告して s を落札できる時, バンドル $s' \subseteq s$ 及び入札額 $v' \geq v$ を申告してもバンドル s' を落札できることを意味する. Critical は, 入札者 i の支払額がその入札額に依存しないこと, 及び入札者 i の支払額は, その入札額より小さい入札額ではその入札が勝てないぎりぎりの値に等しいことを意味する. Participation は, 参加者がオークションに参加することによって負の効用を得ることはないことを意味する. 真実申告最良なプロトコルを実現するためには, 以上の四つの条件を満たす必要があるが, このうち, 勝者決定に関連する部分は, Exactness 及び Monotonicity の二つである.

本論文で提案する近似解法は, そのいずれもが, 決定された勝者に対して Feasible な財の割当が行われることは自明である. よって, Exactness は常に満たされる. したがって, 本提案近似解法を用いたプロトコルでは, Exactness, Critical, 及び Participation を満たすことは, 原理的に可能である. Critical については, 価格設定法を Lehmann [8] の方法に準じることで満たすことができると考えられる. しかし, Monotonicity は, たとえ SA によるランダム探索要素を排除したとしても, 山登り法による解の洗練を加えるだけで, 満たすことができなくなる. 以下に, 山登り法による解の洗練を行うことで, Monotonicity を満たすことができないことを, 具体例で示す.

入札者 i の入札を以下とする.

$$\text{入札 } s = \langle 10000, \{a, b, c, x1\} \rangle, \gamma_s = 2500$$

入札 $s' = \langle 10001, \{a, b, c\} \rangle, \gamma_{s'} = 3334$

$\gamma = a/|s|^c$ が, Lehmann アルゴリズム時にソートする際の基準値である. 本反例では, 簡単のために $c = 1$ を仮定する. ここで Monotonicity を満たすには, s が落札されるなら, s の代わりに s' を申告したとしても s' が必ず落札されなければならない,

他の入札が以下だとする.

入札 $A = \langle 10000, \{x1, x2\} \rangle, \gamma_A = 5000$

入札 $B = \langle 12000, \{x2, x3, x4, x5\} \rangle,$

$\gamma_B = 3000$

入札 $C = \langle 11000, \{x3, x10, x11, x12, x13\} \rangle,$

$\gamma_C = 2200$

入札 $D = \langle 13000,$

$\{x10, x11, x12, x13, x14, x15, x16\} \rangle,$

$\gamma_D = 1857$

入札 $E = \langle 12002, \{a, x2\} \rangle, \gamma_E = 6001$

入札 $F = \langle 11020, \{b, x11\} \rangle, \gamma_F = 5510$

入札 $G = \langle 11010, \{c, x10\} \rangle, \gamma_G = 5505$

もし, 初期の入札集合が $\{A, B, C, D, E, F, G, s\}$ のとき, 勝者の集合は $\{B, D, s\}$ となる. 一方, 初期の入札集合が $\{A, B, C, D, E, F, G, s'\}$ のとき, 勝者の集合は $\{E, F, G\}$ となり Monotonicity は満たされない. (以上, 証明終り)

6. 関連研究

組合せオークションにおける高速な勝者決定アルゴリズムに関する研究は数多く報告されている [3]. 特に著名なものに, CABOB [2] と CASS [7] がある.

一方で, 勝者決定の近似解法については, 本論文でも既に述べた Lehmann [8] らのアルゴリズムがある. Dobzinski [12] らは, 劣モジュラ (submodular) な入札者を対象とした近似解法を提案している. Lavi [13] らは, フラクショナルなドメインにおいて統計的に真実申告最良になる近似オークションを扱っている. 以上の研究では, 既にある組合せオークションを近似的にかつ高速に解くことに主眼を置いているのではなく, むしろ, 組合せオークションと似た頑健なメカニズムをもつような新たなオークションメカニズムを提案することを目的として, 理論的な側面に注力している点で, 本研究とは異なった取組みである.

Zurel と Nisan [9], Hoos [14] 及び Guo [15] からも, 組合せオークションの近似解法について, 本論文と似た視点から提案を行っている. 根本的に異なっている点は, 本プロトコルでは, 初期解として Lehmann [8] アルゴリズムによる解を用いている点である. 特に, Zurel らの手法は, 初期解を求めてからそれを洗練するという 2 段階の方法をとっている点で類似しているが, Zurel らの手法では初期解の探索に線形計画法の近似解法を用いている点で我々の手法とは異なっている. 我々の手法では, あくまでも Lehmann のアルゴリズムに準じる範囲内で, 入札のソート順位を変更する因子である c の値を変更するアプローチを取っているが, Zurel らのアプローチでは, 彼らの提案する線形計画法の近似解法により入札をソートする際の優先順位を決定している点が最も異なる点である. 文献 [9] では, この線形計画法の近似解法の計算オーダは, 入札数 n に対して最悪 $o(n^3)$ であると述べられている. Lehmann アルゴリズムでの初期解の割当の計算オーダは $o(n)$ であり, 入札のソートにかかる計算オーダも最悪でたかだか $o(n^2)$ であるため, 求める初期解の精度にもよるが, 初期解を求めるまでの速度は Lehmann アルゴリズムに基づく我々の手法の方が高速であろうと予想される. 現在, 文献 [9] で示された実験プログラムをもとに, Zurel らの手法との比較を試みている. 後述のとおり, 解の最適性については, 限られた実験範囲についてのみであるが, 我々の提案手法の方が若干高いことが, 現時点では確かめられている. 実行速度に関しては, 実装におけるコーディングテクニック及び詳細な内部パラメータの設定方法によって結果の変動があるため, 公平な比較を行うべく, 実験準備を進めている.

組合せオークション問題の生成手法については, Layton-Brown らの CATS [16] がよく知られている. 本論文では, 非常に多数の入札をもつオークション問題を効率良く生成するために, CATS ではなく独自のアルゴリズムによる問題生成を行っている. 本論文で提案した手法を, Layton-Brown らのテストデータ (財の数 40 ~ 400, 入札数 50 ~ 2000, 分散 9 種類, 各 800 試行分) に対して, 別途本手法を適用してみたところ, 図 6 に示す結果となった^(注2). 入札の生成方法 (モデル) により多少のばらつきはあったが, おおむ

(注2): Ar, Ma, Re, Sc はそれぞれ, Arbitrary, Matching, Regions, Scheduling に対応する. モデルの呼び名は, 文献 [16] による.

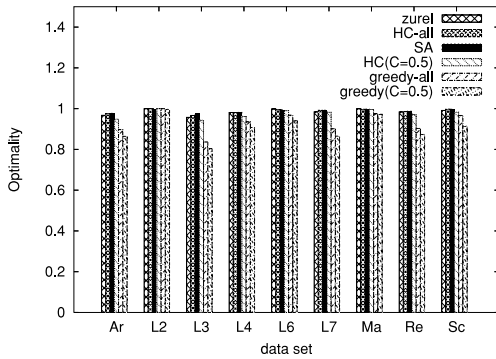


図 6 CATS 共通テストセットによる最適性の比較

Fig. 6 Optimality for CATS Common Testset. (var-size)

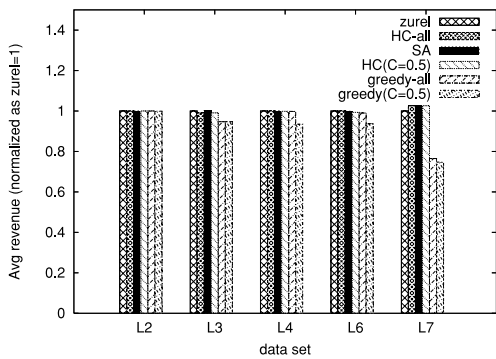


図 7 CATS テストセットによる最適性の比較 (10,000 入札, 256 財)

Fig. 7 Optimality for CATS Testset. (10,000 bids, 256 items)

ね、本論文で提示したのと同様の傾向が見られた。特に、L7 の入札生成モデルにおいて、greedy による結果が低いにもかかわらず、Zurel らの手法と比較して、HC-all 及び SA の解の最適性が良好な結果となっている。

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ため、CATS での本論文で示したような多数入札に対する実験は今後の課題の一つである。現時点では、入札数 10,000 について、標準的なパラメータにより現実的な時間で生成可能であったデータセットを用いた比較評価を行っている (図 7)。なお、入札数が 10,000 となると、最適解を求めることは極めて困難となるため、図 7 では、各モデルに対して単純に総落札価格の平均値を計算し、Zurel らの手法による値を 1 として正規化してある。入札数が 10,000 と増

えた場合でも、図 6 とほぼ同様の傾向が出ている。

計算時間に関しては、Zurel らの手法を用いた場合の方が比較的短時間で済む場合が多いが、我々の手法、Zurel の手法とともに、十分に許容できる程度の短時間で終了しているため、特別な比較は行っていない。ただし、モデルが paths の場合に、Zurel らの手法と比較して明らかに本論文で提案した手法の方が実行速度が遅い場合があることが分かっている^(注3)。この paths というモデルは、経路探索問題という局所最適解の非常に多い種類の問題を組み合わせオークションに射影したものである。入札数を 10,000 まで増やした結果、提案手法にとって極めて不利なテストデータとなった可能性がある。この点から、提案手法では、一般の組合せ最適化問題の高速化には必ずしも貢献しない可能性が考えられる。Zurel らの手法では、線形計画法による近似に近いメカニズムが含まれており、この種の問題についても高速に動作したのと考えられる。Zurel らの手法との、より精密な計算時間の比較は、今後の課題である。

7. む す び

本論文では、多数の入札をもつ組合せオークションの勝者決定問題に対する近似解法を提案し、その特性を評価した。提案した近似解法では、Lehmann アルゴリズムにおけるソート決定因子 c に着目し、更に山登り法による解の洗練を行うことにより、従来提案されていた Lehmann アルゴリズムと比較して、安定して高い最適性をもった解が得られることが分かった。本論文で行った評価実験の結果より、本実験条件では、入札数が 1000 程度で、最適解に対して平均して 0.997 程度の総落札額が得られる解を見つげられることを示した。更に、従来の最適解探索手法と比較して十分に高速に動作し、10,000 を超える入札数をもつオークションに対して実用的な時間で計算が行えることを示した。本提案近似解法は、計算途中で随時途中解を提示することができ、計算終了までの時間のおよそ 1 割程度の時点で、96 パーセント程度の性能の解を提示することが可能であることを示した。

本提案手法の特異な性質として、Admissible という性質を解が満たすことを示した。また、本手法の限界の一つを示すものとして、Lehmann アルゴリズム

(注3): この理由により、計算が完了したテストデータ数が少ないため、図 7 からは省略している。

のもつ真実申告最良 (Truthful) であるという特性は、山登り法による洗練の結果保証できなくなってしまうことを示した。真実申告最良である特性を保ちつつ Lehmann 解を洗練する手法を提案することは、今後の課題の一つである。

文 献

- [1] P. Cramton, Y. Shoham, and R. Steinberg, *Combinatorial Auctions*, 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*, vol.51, no.3, pp.374–390, March 2005.
- [3] S. de Vries and R.V. Vohra, "Combinatorial auctions: A survey," *International Trans. in Operational Research*, vol.15, no.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 Trans. in Operational Research*, vol.11, pp.593–612, 2004.
- [5] T. Sandholm, "An algorithm for optimal winner determination in combinatorial auctions," *Proc. 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. 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. 16th International Joint Conference on Artificial Intelligence (IJCAI99)*, pp.548–553, 1999.
- [8] D. Lehmann, L.I. O'Callaghan, and Y. Shoham, "Truth revelation in rapid, approximately efficient combinatorial auctions," *J. ACM*, vol.49, pp.577–602, 2002.
- [9] E. Zurel and N. Nisan, "An efficient approximate allocation algorithm for combinatorial auctions," *Proc. Third ACM Conference on Electronic Commerce (EC2001)*, pp.125–136, 2001.
- [10] N. Fukuta and T. Ito, "Towards better approximation of winner determination for combinatorial auctions with large number of bids," *Proc. 2006 WIC/IEEE/ACM International Conference on Intelligent Agent Technology (IAT2006)*, pp.618–621, 2006.
- [11] N. Fukuta, T. Ito, and T. Shintani, "A logic-based framework for mobile intelligent information agents," *Poster Proc. of the Tenth International World Wide Web Conference(WWW10)*, pp.58–59, May 2001.
- [12] S. Dobzinski and M. Schapira, "An improved approx-

imation algorithm for combinatorial auctions with submodular bidders," *SODA '06: Proc. seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pp.1064–1073, 2006.

- [13] 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.
- [14] H.H. Hoos and C. Boutilier, "Solving combinatorial auctions using stochastic local search," *Proc. AAAI2000*, pp.22–29, 2000.
- [15] Y. Guo, A. Lim, B. Rodrigues, and Y. Zhu, "A non-exact approach and experiment studies on the combinatorial auction problem," *Proc. HICSS2005*, p.82.1, 2005.
- [16] K. Leyton-Brown, M. Pearson, and Y. Shoham, "Towards a universal test suite for combinatorial auction algorithms," *Proc. EC 2000*, pp.66–76, 2000.

(平成 18 年 11 月 30 日受付)



福田 直樹 (正員)

1997 名工大・工・知能情報システム卒。1999 同大学院工学研究科電気情報工学専攻博士前期課程了。2002 同大学院博士後期課程了。同年 4 月より静岡大学情報学部情報科学科助手。博士 (工学)。セマンティック Web, 及びエージェント間交渉の最適化に興味をもつ。IEEE-CS, ACM, 情報処理学会, ソフトウェア科学会, 人工知能学会, 情報システム学会各会員。



伊藤 孝行 (正員)

平 12 名古屋工業大学大学院工学研究科博士後期課程了。博士 (工学)。平 11~13 にかけて日本学術振興会特別研究員 (DC2 及び PD)。平 12~13 にかけて南カリフォルニア大学 Information Sciences Institute (USC/ISI) 客員研究員。平 13 北陸先端科学技術大学院大学知識科学教育研究センター助教授。平 17~18 にかけて米国ハーバード大学客員研究員及び、米國マサチューセッツ工科大学 Sload School of Management 客員研究員。平 18 より名古屋工業大学大学院産業戦略工学専攻助教授, 現在に至る。2006 International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS2006) 最優秀論文賞受賞。2005 日本ソフトウェア科学会論文賞受賞。平 16 年度 IPA 未踏ソフトウェア創造事業スーパークリエータ認定。第 66 回情報処理学会全国大会優秀論文賞及び奨励賞受賞。マルチエージェントシステム, 計算論的メカニズムデザイン, オークション理論, 電子商取引に興味をもつ。AAAI, ACM, 情報処理学会, 日本ソフトウェア科学会, 人工知能学会, 計測自動制御学会各会員。