短時間再割り当てを考慮した組合せオークション勝者決定の高速近似手法

福田 直樹　伊藤 孝行

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場合がある．我々は，このような場合には，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，組合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と，それが持つべき好ましい性質についての解析および考察を行ってきた．しかしながら，これまでの我々のアプローチ，および関連研究のアプローチでは，いずれも，個々のオークションを独立した問題としてとらえ，オークション同士の関係を利用した勝者決定の近似の効率化の可能性については検討してこなかった．ユビキタス環境下での資源配分問題などでは，短時間で状況が変化し，それに伴って資源の再割り当ての必要が頻繁に生じるような状況があると考えられる．本論文では，組合せオークションが何度も繰り返され，そのオークション間の入札の差分が小さい場合に，直前のオークションにおける近似勝者を効果的に再利用する手法について述べる．

In this paper, we propose enhanced approximation algorithms of combinatorial auction that are suitable for the purpose of periodical reallocation of items. Our algorithms are designed to effectively reuse the last solutions to speed up initial approximation performance. We present experimental results that show our proposed algorithms outperform existing algorithms in some aspects when the existing bids are not deleted. Also, we propose an enhanced algorithm that effectively avoids undesirable reuse of last solutions in the algorithm. This is especially effective when some existing bids are deleted from the last cycle.

1 はじめに

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なオークションである [2]．組合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 [21] 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．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 [19] [20] [5] [2] が，この問題は NP-hard であることが知られており [2]，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 [5] 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．さら

Towards Approximated Short Cycle Reallocation on Combinatorial Auctions.
Naoki Fukuta, 静岡大学情報学部, Faculty of Informatics, Shizuoka University.
Takayuki Ito, 名古屋工業大学大学院産業戦略専攻・情報工学科/マサチューセッツ工科大学スローン経営大学院, School of Techno-Business Administration/Dept. of Computer Science, Nagoya Institute of Technology/Sloan School of Management, Massachusetts Institute of Technology.
コンピュータソフトウェア, Vol.25, No.4 (2008), pp.208–225.
[研究論文] 2007 年 11 月 30 日受付.

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合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，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場合もある．我々は，そのような場合には，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，組合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 [10] についての解析 [7] [6] および考察 [9] を行ってきた．

しかしながら，これまでの我々のアプローチ，および関連研究のアプローチでは，いずれも，個々のオークションを独立した問題としてとらえ，オークション同士の関係を利用した勝者決定の近似の効率化の可能性については検討してこなかった．ユビキタス環境下での資源配分問題などでは，短時間で状況が変化し，それに伴って資源の再割り当ての必要が頻繁に生じるような状況があると考えられる．本論文では，組合せオークションが何度も繰り返され，そのオークション間の入札の差分が小さい場合に，直前のオークションにおける近似勝者を効果的に再利用する手法について述べる．

以降の本論文の構成は，次のようになっている．2 節では，本論文の基礎となる組合せオークションと勝者決定問題に関する定義とこれまでの研究内容を概観する．3 節では，我々が本論文で新しく提案するアルゴリズムの詳細を述べる．4 節では，提案アルゴリズムの有効性を評価し，その結果を示す．5 節で，本提案手法および本論文での評価手法における限界について議論する．6 節で，関連研究との違いを示す．最後に，7 節で本論文をまとめ，今後の課題について述べる．

2 組合せオークションと勝者決定問題

2.1 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 [14] 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に近い性質を示す．ここで，ある入札を $b =< a, s >$(ただし，$s \subseteq M$ かつ $a \in \mathcal{R}_+$) と表現することにする．また，2 つの入札 $b =< a, s >$ および $b' =< a', s' >$ 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 [14] 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 [14] 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ことを繰り返し行う．

文献 [14] では，ソート時のパラメータ c について，$c = 1/2$ とすると理論上の下界を適切に保証でき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[†1].

2.2 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．我々はこれまでに，文献 [8] などで，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を提案してきている．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.

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．

function LocalSearch($Alloc$, L)
　$RemainBids$:= $L \cap \overline{Alloc}$;
　for each $b \in RemainBids$ as sorted order
　　if b conflicts $Alloc$ **then**
　　$Conflicted$:=$Alloc \cap \overline{consistentBids(\{b\}, Alloc)}$;
　　　$NewAlloc$:= $(Alloc \cap \overline{Conflicted}) \cup \{b\}$;
　　　$ConsBids$:=
　　　　consistentBids($NewAlloc, RemainBids$);
　　　$NewAlloc$:=$NewAlloc \cup ConsBids$;
　　if $price(Alloc) < price(NewAlloc)$ **then**
　　　return LocalSearch($NewAlloc$,L);
　end for each
　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 “NewAlloc” 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．

2.3 複数値のソート因子 c に対する並列探索

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．我々は，山登り法による解の改善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手法を提案してきている [10], [8].

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での性能も考慮して，$C = \{0,\ 0.5,\ 1\}$ の 3 つの c の値を用いることとした．

3 近似アルゴリズムの拡張

3.1 直前の近似結果の部分的再利用による高速化

定期的に何度も資源の再割り当てを行うようなシナリオを考えた場合，その割り当て毎に全く異なる入札集合を扱うことはむしろ稀であり，直前の入札集合からわずかに変更されたものを対象にする場合が多くなる状況も考えられる．しかしながら，これまでに用いられてきた勝者決定アルゴリズムでは，たとえ入札集合が似ている組合せオークションであっても，それらを独立した別個の問題として扱うようになっていた．そのため，たとえわずかな入札の変更が加えられただけであっても，勝者を改めて最初から決定し直す必要があった．以下で，直前に勝者近似で用いられた入札集合に対する勝者近似結果を再利用し，近似を高速化することが可能なアルゴリズムを提案する．以下のアルゴリズムでは，対象とするオークションの入札集合のみでなく，直前の入札集合 ($LastBids$) とそのときの勝者集合 ($LastWinners$) を与えることで，その差分を計算し，勝者近似をより短時間で行えるようにする．なお，$b(x)$ および $v(x)$ は，入札 x の対象財バンドルおよび入札価格をそれぞれ示す.

†1 文献 [21] では，c の値は 0.8 から 1.0 のとき良好な結果が得られたと報告している．

Function PartialReallocationA(
$\quad\quad LastBids, LastWinners, CurrentBids)$
$\quad AddedBids :=$
$\quad\quad CurrentBids \cap (\overline{LastBids \cap CurrentBids});$
$\quad DeletedBids :=$
$\quad\quad LastBids \cap (\overline{LastBids \cap CurrentBids});$
$\quad Winners := LastWinners;$
$\quad$ **foreach** $d \in DeletedBids$
$\quad\quad$ **if** $d \in Winners$
$\quad\quad\quad$ **then** $Winners := Winners \cap \overline{\{d\}};$
$\quad$ **foreach** $a \in AddedBids$
$\quad\quad$ **foreach** $w \in Winners$
$\quad\quad\quad$ **if** $b(w) = b(a)$ and $v(w) < v(a)$
$\quad\quad\quad\quad$ **then** $Winners := (Winners \cap \overline{\{w\}}) \cup \{a\};$
$\quad Winners :=$
$\quad\quad$ LocalSearch($Winners, CurrentBids$);
$\quad$ **return** $Winners$

本アルゴリズムでは，最初に，直前の入札集合と比較して「削除された」と考えられる入札を，勝者集合から削除する．次に,「新たに加えられた」入札について，それを必要に応じて勝者集合と置き換えていく．このときの置き換えでは，新たに加えられた入札がすでに勝者となっている入札とまったく同一の財の集合に対して入札を行っており，かつその入札額がすでに勝者となっている入札より大きい場合にのみ，置き換えが適用される．なお，入札の改変 (modification) は，入札の削除と新規入札の追加の 2 つのオペレーションの組合せに分解されたものとして扱われる．

3.2 近似結果の再利用による性能劣化への対処

一般的には，類似した問題の答えを再利用することは，最適化問題を解く上では必ずしも有利には働かない．実際に，類似した問題の答えを利用することで，むしろ性能が劣化する場合が，しばしば生じる．このような性能劣化をできるかぎり少ない計算オーバヘッドで避ける方法を考える．

我々がここで採用する方法は，単純に,「初期解導出時に，再利用による結果が単純な greedy 割り当てより劣るのであれば，それを捨て，初期解を greedy 割り当てに置き換える」というものである．初期解導出後の動作は，我々がこれまでに提案した山登り法に基づくアルゴリズムと同様である．これまでの実験結果により，greedy 割り当ての計算オーバヘッドは非常に小さいことがわかっており，我々が先に示した部分的再利用アルゴリズムにおける計算オーバヘッドは十分軽微なものであれば，仮に近似結果の再利用がうまくいかなかった場合でも，その計算オーバヘッドが全体の結果に悪影響を及ぼさないと考えられる．以下が，そのアルゴリズムの記述である．

Function PartialReallocationX(
$\quad\quad LastBids, LastWinners, CurrentBids)$
$\quad AddedBids :=$
$\quad\quad CurrentBids \cap (\overline{LastBids \cap CurrentBids});$
$\quad DeletedBids :=$
$\quad\quad LastBids \cap (\overline{LastBids \cap CurrentBids});$
$\quad Winners := LastWinners;$
$\quad$ **foreach** $d \in DeletedBids$
$\quad\quad$ **if** $d \in Winners$
$\quad\quad\quad$ **then** $Winners := Winners \cap \overline{\{d\}};$
$\quad$ **foreach** $a \in AddedBids$
$\quad\quad$ **foreach** $w \in Winners$
$\quad\quad\quad$ **if** $b(w) = b(a)$ and $v(w) < v(a)$
$\quad\quad\quad\quad$ **then** $Winners := (Winners \cap \overline{\{w\}}) \cup \{a\};$
$\quad GreedyWinners :=$
$\quad\quad$ GreedySearch($CurrentBids$);
$\quad$ **if** $price(Winners) \le price(GreedyWinners)$
$\quad\quad$ **then** $Winners := GreedyWinners;$
$\quad Winners :=$
$\quad\quad$ LocalSearch($Winners, CurrentBids$);
$\quad$ **return** $Winners$

4 比較実験

4.1 実験条件と表記

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を $c = \{0,\ 0.5,\ 1\}$ 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．また，本論文で提案する 2 つのアルゴリズム (`PartialReallocationA`, `PartialReallocationX`) を用い，後述の方法で近似

解を再利用した結果を “AHC-3”, および “XHC-3” で始まる名前で表記している．なお，“-3” の表記からわかるとおり，いずれも，$c = \{0,\ 0.5,\ 1\}$ の 3 つのパラメータについて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 [26] 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 [12] 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．

実験環境には， Mac OS X 10.4, 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．

4.2 評価方法

組合せオークション問題の生成手法については，Leyton-Brown らの CATS [15] 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 [15] 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 [15] を参照されたい．

CATS の問題生成手法では，オークションにおける入札は静的なものとして扱われているため，本論文で述べるような，少量の入札差分をもつ一連のオークションの集まりとしては問題が構成されない．そこで，本論文では，手順 1 に示す方法で，CATS で生成した問題を，少量の入札差分をもつ一連のオークションの集まりとして再構成した．

手順 1　CATS で生成したそれぞれのオークション問題について，その入札集合を，入札生成順序に沿って k 等分する．この k 等分した入札集合を用いて，1 秒毎に入札集合を変更する操作を合計 k 回行う．ここで，各入札集合は，k 等分されたものの先頭から 1 つずつ順に「削除」印をずらしていき,「削除」印のない入札集合のみを集めたものを，その回のオークションの入札集合とする．たとえば，最初の 1 秒間における入札集合は，k 等分されたうちの先頭の 1 つに「削除」印が付され，それを除いた $k-1$ 個の入札部分集合の和で入札集合が構成される．その 1 秒後には，さきほど入札集合に含まれていなかった最初の 1 ブロック分の入札部分集合が入札集合に追加され，そのかわりに，先頭から 2 ブロック目の入札部分集合に「削除」印が付され，入札集合から削除される．これを最終ブロックの入札部分集合に「削除」印が付されるまで合計 k 回繰り返し，最後の $k+1$ 回目のみ，すべての入札部分集合からなる入札集合を対象として勝者決定が行われる．

この手順では，従来我々が提案，および比較対象としてきた手法では，入札集合の更新のたびに，あらためて勝者決定を最初からやり直すことになる．本論文で提案する手法では，入札集合の更新の際に，直前の入札集合に対する近似勝者を再利用しようとする．ここで重要な点は，最後の 1 回の試行ではすべての入札を対象に勝者決定が行われるため，その結果は，これまでに我々が比較実験に用いてきた結果と直接的に比較することができる点である．

本評価実験では，この手順 1 を用い，2 つの側面から提案アルゴリズムの評価を行う．手順 1 は，全 $k+1$ 回の勝者決定の試行から構成される．本評価実験では，そのうちの，最後 ($k+1$ 回目) の試行と，初回と最後を除く (2 から k 回目までの) 試行に着目し，それぞれ，最終回試行，中途回試行と呼ぶことにする．

表 1 近似時間と近似の最適性 (20,000 入札,256 財,k=10, 解の最適性は 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)
AHC-3-seq-100ms	1.0004	(100)	0.9690	(100)	1.0006	(100)	0.9974	(100)	1.0225	(100)
XHC-3-seq-100ms	1.0004	(100)	0.9813	(100)	1.0005	(100)	0.9987	(100)	1.0217	(100)
HC-3-para-100ms	1.0004	(100)	0.9743	(100)	1.0001	(100)	0.9969	(100)	0.9423	(100)
AHC-3-para-100ms	1.0004	(100)	0.9741	(100)	1.0006	(100)	0.9977	(100)	1.0249	(100)
XHC-3-para-100ms	1.0004	(100)	0.9820	(100)	1.0006	(100)	0.9988	(100)	1.0249	(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)
AHC-3-seq-1000ms	1.0004	(1000)	0.9795	(1000)	1.0007	(1000)	0.9982	(1000)	1.0266	(1000)
XHC-3-seq-1000ms	1.0004	(1000)	0.9830	(1000)	1.0006	(1000)	0.9991	(1000)	1.0266	(1000)
HC-3-para-1000ms	1.0004	(1000)	0.9856	(1000)	1.0006	(1000)	0.9987	(1000)	1.0240	(1000)
AHC-3-para-1000ms	1.0004	(1000)	0.9847	(1000)	1.0008	(1000)	0.9990	(1000)	1.0272	(1000)
XHC-3-para-1000ms	1.0004	(1000)	0.9853	(1000)	1.0008	(1000)	0.9996	(1000)	1.0272	(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)

最終回試行では，直前の入札の入れ替えで入札の削除がないという条件のため，本提案アルゴリズムによる性能の改善が比較的起きやすい状況である．ただし，その前に k 回の試行を経ているため，中途回試行時に極端な性能劣化が起きている場合には，その影響が出ることになる．これらの点に着目して，最終回試行における性能から，本提案アルゴリズムによる性能向上がどの程度見られるかを，他のアルゴリズムによる結果との比較により評価を行う．

中途回試行では，直前の入札の入れ替え時に入札の削除と追加が同時に起きるため，最終回試行と比較して，本提案アルゴリズムにとってはシビアな条件設定になる．中途回試行では，本提案アルゴリズムがHC-3 アルゴリズムと比較してシビアな条件下での性能劣化をどの程度抑えられるか，また，その条件下でもなお性能の向上が見られる場面があるか，という観点から評価を行う．

4.3 最終回試行における性能の評価

表 1 に，$k = 10$ における，本手順に基づいた最後の 1 回の試行時での，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まとめる．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 [5] 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 [5]．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．

ほとんどの入札分散について，“Zurel-1st” では 1 秒以上の時間がかかっているにもかかわらず，その近似性能は “greedy-3-seq” に劣っている．我々がこれまでに提案してきた “HC-3” では，入札分散 L3 を除けば，その計算にかかる時間が “Zurel-1st” や “Zurel” と比較して十分小さいにもかかわらず，そ

表 3　近似時間と近似の最適性 (100,000 入札,256 財,k=10, 解の最適性は Zurel=1 に正規化)

	L2		L3		L4		L6		L7	
HC-3-para-100ms	1.1098	(100)	0.9836	(100)	1.0003	(100)	1.0009	(100)	0.8688	(100)
AHC-3-para-100ms	1.1098	(100)	0.9836	(100)	1.0003	(100)	1.0009	(100)	0.9941	(100)
XHC-3-para-100ms	1.1098	(100)	0.9880	(100)	1.0003	(100)	1.0010	(100)	0.9939	(100)
HC-3-seq-1000ms	1.1098	(1000)	0.9859	(1000)	1.0003	(1000)	1.0009	(1000)	0.9395	(1000)
AHC-3-seq-1000ms	1.1098	(1000)	0.9859	(1000)	1.0003	(1000)	1.0009	(1000)	0.9943	(1000)
XHC-3-seq-1000ms	1.1098	(1000)	0.9884	(1000)	1.0003	(1000)	1.0010	(1000)	0.9941	(1000)
HC-3-para-1000ms	1.1098	(1000)	0.9880	(1000)	1.0003	(1000)	1.0010	(1000)	0.9814	(1000)
AHC-3-para-1000ms	1.1098	(1000)	0.9880	(1000)	1.0003	(1000)	1.0010	(1000)	0.9991	(1000)
XHC-3-para-1000ms	1.1098	(1000)	0.9889	(1000)	1.0003	(1000)	1.0011	(1000)	0.9990	(1000)
zurel-1st	0.8971	(74943)	0.9827	(2257)	0.9998	(5345)	0.9987	(4707)	0.7086	(8688)
Zurel	1.0000	(91100)	1.0000	(6036)	1.0000	(30568)	1.0000	(44255)	1.0000	(17691)

(括弧内の数値はミリ秒単位での計算時間)

表 2　近似時間と近似の最適性 (表 1 の全入札分散の平均)

	average	
greedy(c=0.5)	0.9170	(22.1)
greedy-3-seq	0.9429	(67.2)
greedy-3-para	0.9429	(25.4)
HC(c=0.5)-100ms	0.9423	(100)
HC-3-seq-100ms	0.9590	(100)
AHC-3-seq-100ms	0.9980	(100)
XHC-3-seq-100ms	1.0005	(100)
HC-3-para-100ms	0.9828	(100)
AHC-3-para-100ms	0.9995	(100)
XHC-3-para-100ms	1.0013	(100)
HC(c=0.5)-1000ms	0.9887	(1000)
HC-3-seq-1000ms	0.9975	(1000)
AHC-3-seq-1000ms	1.0011	(1000)
XHC-3-seq-1000ms	1.0019	(1000)
HC-3-para-1000ms	1.0019	(1000)
AHC-3-para-1000ms	1.0024	(1000)
XHC-3-para-1000ms	1.0027	(1000)
Zurel-1st	0.8265	(3433)
Zurel	1.0000	(5470)
casanova-10ms	0.0632	(10)
casanova-100ms	0.1107	(100)
casanova-1000ms	0.3305	(1000)

(括弧内の数値はミリ秒単位での計算時間)

の近似性能は同程度となっている．本論文で提案している “XHC-3-seq-100msec” は，その計算時間が “HC-3-seq-1000ms” の 1/10 であるにもかかわらず，近似解の再利用によって，その近似性能が “HC-3-seq-1000ms” をほとんどの場合で上回っている．

表 2 に，表 1 の全入札分散での平均のみをまとめたもの示す．我々がこれまでに提案してきた手法 “HC-3” と比較して，“XHC-3” は，計算時間がおよそ 1/10 であっても，ほぼ同程度かそれ以上の品質の解を導出できている．また，“Zurel” の解の品質を実質十分な品質と考えたとき，それと同等の品質の解を，“Zurel” が平均で 5.4 秒程度かけて導出しているのに対し，“XHC-3” では，実行の並列性を問わず 0.1 秒での近似により得ることができている．

表 4　近似時間と近似の最適性 (表 3 の全入札分散の平均)

	average	
HC-3-para-100ms	0.9927	(100)
AHC-3-para-100ms	1.0177	(100)
XHC-3-para-100ms	1.0186	(100)
HC-3-seq-1000ms	1.0073	(1000)
AHC-3-seq-1000ms	1.0182	(1000)
XHC-3-seq-1000ms	1.0187	(1000)
HC-3-para-1000ms	1.0161	(1000)
AHC-3-para-1000ms	1.0197	(1000)
XHC-3-para-1000ms	1.0198	(1000)
zurel-1st	0.9174	(19188)
Zurel	1.0000	(37930)

(括弧内の数値はミリ秒単位での計算時間)

表 3 に，表 1 と同条件で，1 オークションあたりの入札数のみを 20,000 から 100,000 に増加させたときの計測結果を示す．“HC-3” と “XHC-3” との傾向の差は入札数 20,000 のときとほぼ同様である．“XHC-3” と “Zurel-1st” および “Zurel” との比較では，入札数の増加に対して，提案手法である “XHC-3” の計算時間面での有効性が明確になっている．“XHC-3-para-100ms” は，“Zurel-1st” に対し，計算時間では 22〜86 倍小さく，かつその最適性はいずれの場合も “Zurel-1st” よりもすぐれている．

表 5 近似時間と近似の最適性 (20,000 入札,256 財,k=2,5,10,20,40 時の比較．解の最適性は Zurel=1 に正規化)

	k=2		k=5		k=10		k=20		k=40	
HC-3-para-100ms	0.9828	(100)	0.9828	(100)	0.9828	(100)	0.9828	(100)	0.9828	(100)
AHC-3-para-100ms	0.9952	(100)	0.9979	(100)	0.9995	(100)	1.0003	(100)	1.0009	(100)
XHC-3-para-100ms	0.9952	(100)	0.9998	(100)	1.0013	(100)	1.0021	(100)	1.0028	(100)
HC-3-para-1000ms	1.0019	(1000)	1.0019	(1000)	1.0019	(1000)	1.0019	(1000)	1.0019	(1000)
AHC-3-para-1000ms	1.0019	(1000)	1.0021	(1000)	1.0024	(1000)	1.0026	(1000)	1.0027	(1000)
XHC-3-para-1000ms	1.0019	(1000)	1.0025	(1000)	1.0027	(1000)	1.0031	(1000)	1.0035	(1000)

(括弧内の数値はミリ秒単位での計算時間)

表 4 に，表 3 に示した結果の全入札分散での平均のみをまとめたものを示す．入札数 20,000 のときと同様に，“Zurel” の解の品質を実質十分な品質と考えたとき，それと同等の品質の解を，“HC-3” では 0.1 秒では得られず，“Zurel” が平均で 37.9 秒程度かけて導出しているのに対し，“XHC-3” では，0.1 秒での近似により得ることができている．

表 5 に，入札数 20,000 において，手順 1 における k の値を 2, 5, 10, 20, 40 と変化させたときの結果を，全入札分散での平均で示す．なお，ここでは k の値の変更により影響を受けるアルゴリズムは “AHC-3” と “XHC-3” のみであるが，比較を行いやすくするために，“HC-3” の結果も並記してある．結果は，他の表と同様に，“Zurel” の結果を 1 として正規化してある．表 5 から，計算時間が 0.1 秒の場合には “XHC-3” は “HC-3” と比較して $k = 2$ から高い結果が得られており，“Zurel” と同等の品質の解は，$k = 10$ かそれ以上のときで得られている．すなわち，計算時間に対する制約が 0.1 秒という状況下では，入札の入れ替えが非常に大きい $k = 2$ の状況下でも “XHC-3” を利用する利点があり，$k = 10$ に相当する入札の入れ替え程度以下の入札の変化であれば，その時間制約内に “Zurel” と同等の十分な品質の解を得られる．また，$k = 20$ かそれより緩やかな入札の入れ替えしか起きない状況では，“HC-3” と比較して 10 倍程度以上の時間性能の向上が得られる．以上はいずれも本実験条件でのみいえることであるが，一般的に考えて，入札の入れ替えのバースト性が少なくとも $k = 2$ 程度以下に納まっていれば，特に時間制約の厳しい環境下では本提案アルゴリズムを用いるメリットがあり，なおかつ本提案アルゴリズムを用いる際のデメリットはほとんどないと考えられる．

4.4 中途回試行での性能の評価

図 1 に，k=10 での，入札数 20,000 における，3 つの手法 “HC-3-para”, “AHC-3-para”, および “XHC-3-para” における，手順 1 試行時の時間変化に伴う近似性能の変化を示す．ここでは，各入札分散ごとの違いがわかるように，入札分散ごとに分けて値をプロットしている．手順 1 による繰り返し入札更新では，1 秒ごとの区間が各オークション問題に対応するため，手順 1 を行う途中で「総落札額の理論最大値」も入札の更新毎に変化する．そこで，図 1 では，各区間内での勝者近似性能の時間的な変化を相互に比較可能とするために，各区間内で “HC-3-para-1000msec” で得られた近似結果を 1 として，値を正規化してある．たとえば，2100msec 時点での値が 0.95 であったとすると，その意味は，2000msec から 3000msec までの区間に “HC-3-para-1000msec” で得られた近似結果 (総落札額) の 95 パーセントの合計落札額を持つ勝者を，その入札集合に対する近似計算の開始から 100msec 後に発見することができたことを示している．

入札分散が L7 のときには，“AHC-3” および “XHC-3” の結果が，“HC-3” と比較して良くなっている．しかしながら，他の入札分散 (L3,L4,L6) では，最終区間での近似結果を除けば，“AHC-3” の結果は “HC-3” の結果に大きく劣る場合がほとんどである．これは，本論文で示した手順 1 による入札集合の変化により，“AHC-3” が，近似勝者の「適切でない再利用」を行う場合が多くなったため，山登り方による解の改善のための初期値が非常に悪い点に定められたことによると考えられる．これに対して，“XHC-3” では，

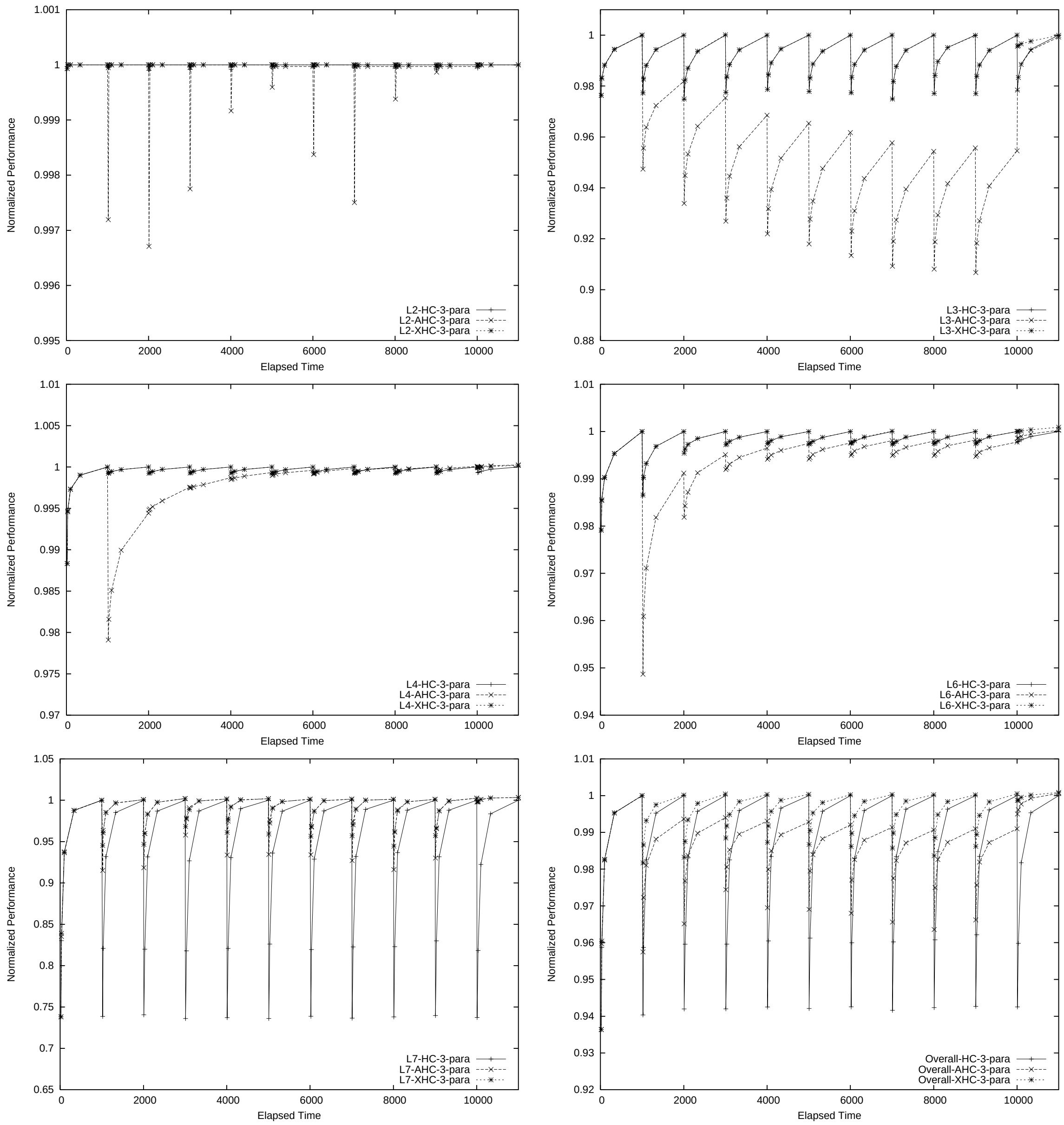

図 1　k=10 における，手順 1 による繰り返し入札更新時における近似性能の時間的変化 (試行全体)

“AHC-3” に見られるような解の改悪の様子は見られない．“XHC-3” は，入札分散 L7 では “AHC-3” と同程度の性能向上を得られながら，他の入札分散でも少なくとも “HC-3” と同程度の性能を維持していることから，“XHC-3” の手法は，近似解の再利用を適切に行いながら，性能の低下を避けるという動作が効果的に動作していることがわかる．また，入札分散全体の平均 (図中では “Overall-” と表記) では，解の再利用が原理的に生じない初回の試行を除き，いずれの区間でも “XHC-3” が “HC-3” を上回っている．

図 2 に，図 1 と同条件で，中途回試行区間 (2〜k 回試行) のみを取り出し，その平均をとったものを示す．すなわち，図 2 は，中途回試行区間における各試行の平均的な時間性能を，入札分散ごとに示した

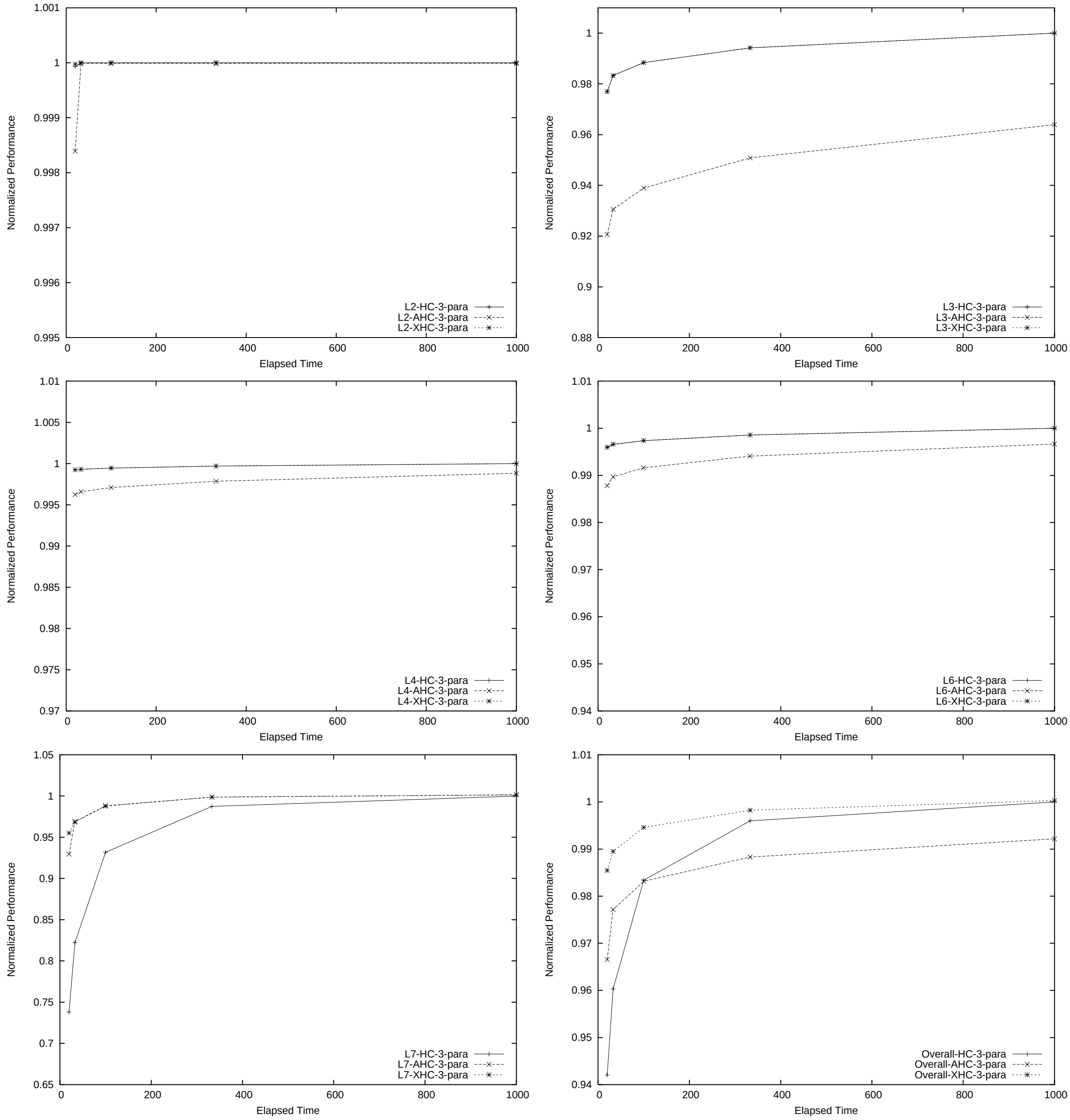

図 2　k=10 における，中途回試行での近似性能

ものである．先に述べた現象は，図 2 からはさらに明確に読み取れる．入札分散 L2,L3,L4, および L6 では，“HC-3” と “XHC-3” の結果はほぼ同じであるが，“AHC-3” の結果のみが，これらより劣った結果となっている．入札分散 L7 では，“AHC-3” と “XHC-3” がほぼ同等の結果であるが，“HC-3” のみがこれら 2 つよりも劣った結果となっている．入札分散全体の平均 (図中では “Overall-” と表記) では，“XHC-3” は “HC-3” よりも全体的に高い結果となっており，たとえば “HC-3” では 100 ミリ秒で得られる結果よりも高い結果を ‘XHC-3” では初期解の段階 (およそ 20 ミリ秒程度) で得られており，また “HC-3” が 333 ミリ秒で得られる以上の結果を “XHC-3” では 100 ミリ秒の段階で得られている．この結果から，この実験条

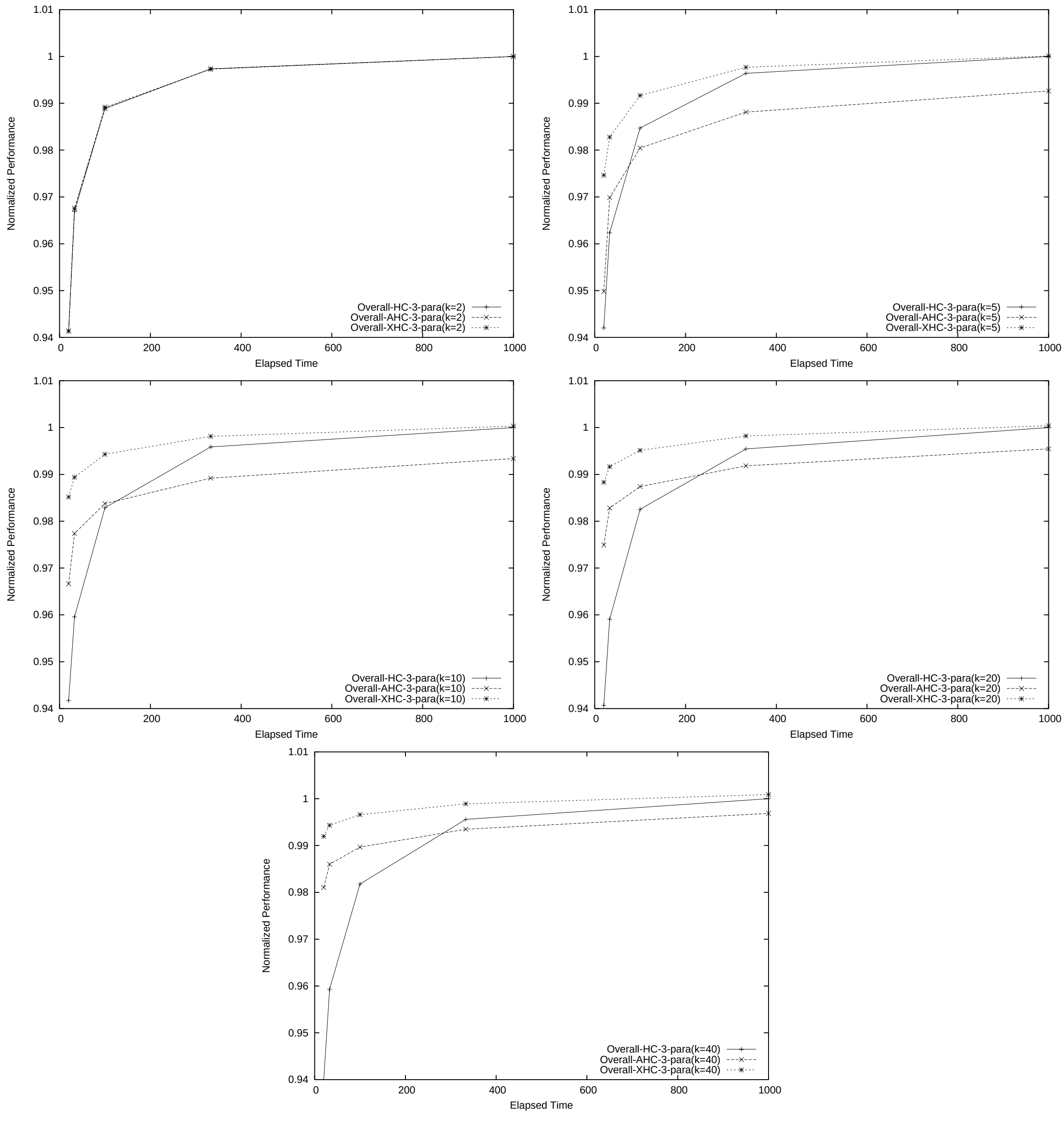

図 3　k=2,5,10,20,40 における，中途回試行での近似性能の変化 (全入札分散での平均)

件下では，中途回試行の区間であっても，“XHC-3” は “HC-3” の 3〜5 倍程度の時間性能を平均的に発揮しているといえる．

図 3 に，図 1 と同条件で，k の値を 2,5,10,20,40 と変化させたときの中途回試行区間での時間性能の平均を示す．ただし，図 3 では，k の値の変更に伴って 1 手順中での中途回試行の回数が変化することを考慮し，$k = 2$ のときを除き，2〜5 回目までの中途回試行の平均を取っている[†2]．これは，1 手順あたりの試行回数が増えたときに，全中途回試行区間での平均を取ると，解の再利用が連続的に起きる結果として k が大きいほど有利になる (平均として求めた値が本来

†2　$k = 2$ のときは，中途回試行が 1 回しかないため，その結果をそのまま掲載している．

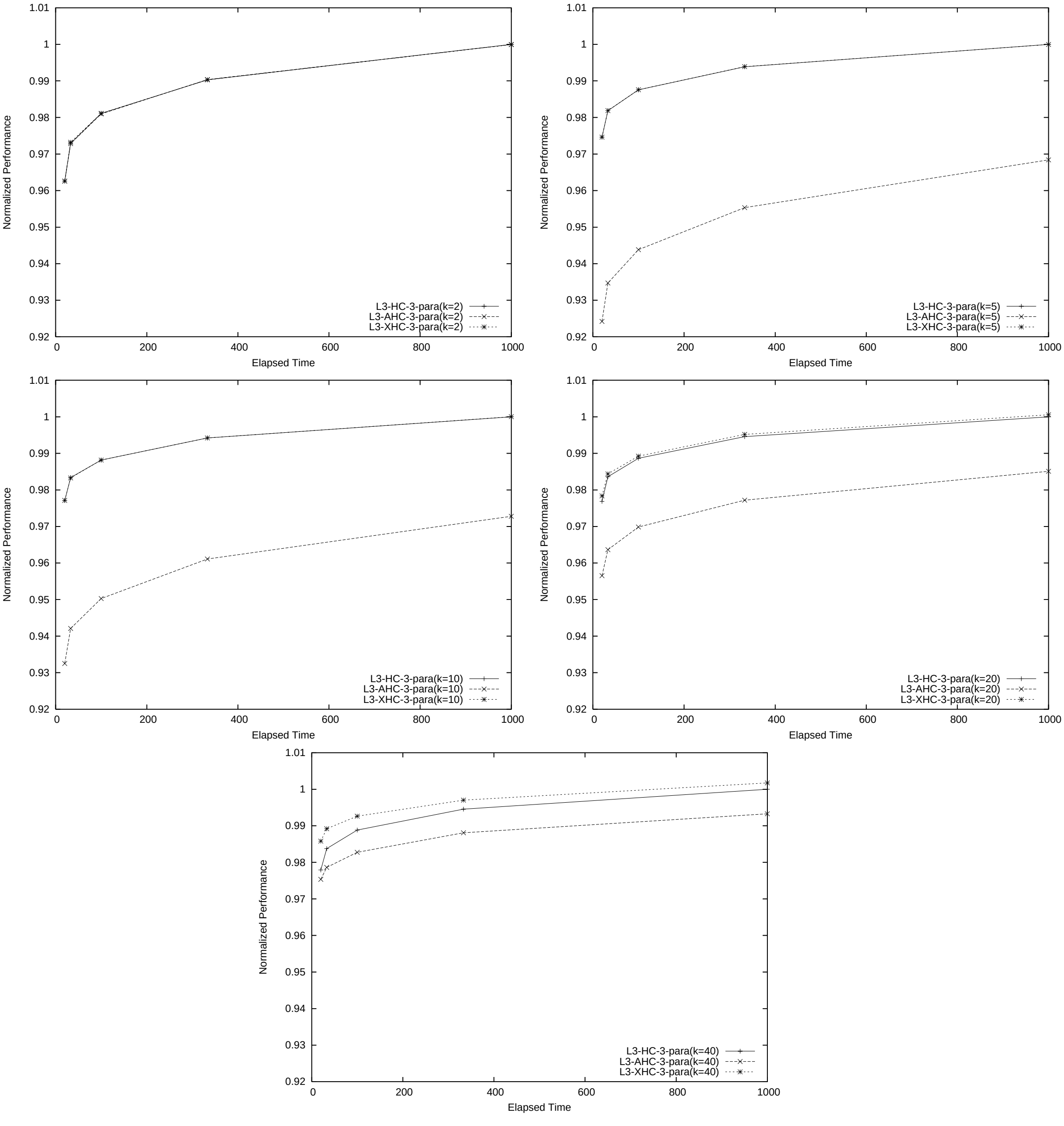

図 4　k=2,5,10,20,40 における，中途回試行での近似性能の変化 (入札分散 L3)

の性能以上に上がってしまう) ことを考慮し，できるだけ公平な条件でこれらの結果を比較できるようにするためである．

図 3 より，$k = 2$ の場合では “HC-3” と “XHC-3” は同等であるが，$k = 5$ 以上の場合では，“HC-3” の結果を “XHC-3” が上回っている．この実験条件下では，効果の差はあるが，$k = 2$ の場合を除けば中途回試行のような条件下でも本提案アルゴリズムを利用するメリットがあると考えられる．

図 4 に，図 3 と同条件で，特に入札分散 L3 の結果のみを取り出した結果を示す．これは，中途回試行の場面で入札分散 L7 以外での本提案手法によるメリットがどの程度得られるのかを知るのに役立つ．図 2 に示された結果から，全入札分散の平均値に対して入札

分散 L7 の結果の影響が非常に大きいことがわかる．また，同様に，図 2 の入札分散 L2,L3,L4, および L6 では，“XHC-3” と “HC-3” の結果に変わりがない．図 4 では，$k = 20$ のときに “XHC-3” が “HC-3” をわずかに上回っており，$k = 40$ では，その差が明確になっていることがわかる．このことから，この実験条件下では，入札分散 L3 でも $k = 20$ 以上のときに本提案アルゴリズムを用いるメリットがあると考えられる．なお，紙面の都合で割愛するが，入札分散 L4, および L6 についても，L3 と同様の傾向が見られた．入札分散 L2 では k の値にかかわらず “HC-3” と “XHC-3” の差は確認できなかったが，これは，入札分散 L2 では，初期解の段階ですでにほぼ最適解に到達しており，これ以上の解の改善の余地がなかったことが原因と考えられる．

5 議論

5.1 問題の特性と提案手法の有効性との関連

本論文で示した評価実験では，オークション問題を生成するための入札分散の性質の違いにより，提案アルゴリズムの有効性に差が出ることが示されている．特に，入札分散が L7 のときに，本提案アルゴリズムによる性能の改善の度合いが著しく，逆に，入札分散 L2,L4,L6 では，単純に解の最適性の値だけを見れば，“greedy-3” の結果からの大きな改善が見られないように見える．本節では，入札分散の違いによるオークション問題の構造の違いを示し，本提案アルゴリズムの有効性との関連について議論する．

入札分散 L7 によって生成されるオークション問題は，少数の入札の組合せにより (準) 最適な勝者が構成されるが，Lehmann アルゴリズムで用いられる入札の重み値 ($a/|S|^c$) が高い特定の入札が勝者決定にとって支配的とはならず，むしろ入札間の組合せが重要となるような問題構造を持っている．すなわち，Lehmann アルゴリズムに代表される greedy な財の割り当て手法が有効に働きにくい．このため，落札額が高額になるような入札の組合せは，“HC-3” のような手法では見つけるまでに時間がかかる．それらが再利用できれば，計算時間の節約につながる可能性が高い．このため，入札分散 L7 では，従来手法ではよい結果が得られにくく，本提案アルゴリズムの有効性が発揮されやすかったと考えられる．

入札分散 L2 によって生成されるオークション問題では，1 つの入札が非常に多数の財のバンドルに対してのものであり，勝者となる入札がほとんどの場合で 1 つとなるような問題構造を持っている．このため，greedy な財の割り当てにより，ほぼ最適解が得られてしまう．同時に，Zurel アルゴリズムが内部で行っている個々の財に対する評価値の推定がほとんど勝者決定には反映されず，なおかつその推定における誤差率が下がらないため，特に “Zurel-1st”(Zurel の初期解導出) に大きな時間がかかる．このため，このオークション問題の構造はきわめて Zurel アルゴリズムにとって不利にできているといえる．本論文ではこの点を加味し，実験結果として全入札分散における平均だけでなく，各入札分散ごとのデータも可能な限り掲載している．

入札分散 L3,L4,L6 によって生成されるオークション問題では，(準) 最適な勝者が比較的多くの入札から構成されるため，特に入札数が多数になった場合には，“greedy-3” で良好な結果が短時間で得られる．一方で，greedy な割り当てで得られた解を改善する速度はきわめて遅い，という問題の構造を持っている．このため，絶対的な解の最適性は “greedy-3” のものから劇的には向上しないが，解の改善の速度については，“HC-3” と比較した場合に本提案アルゴリズムによる十分な改善が見られる．対象とするアプリケーションや使用する価格付け方法に依存するため一概には言えないが，総落札価格としてみた場合の 0.1 パーセントの改善は，オークション主催者の立場から見れば小さく見えるが，特定の出品者からみた場合にはその影響は決して軽微ではなく，その改善に意味を見いだせる可能性がある．そのような場面には，本提案アルゴリズムは有効に機能すると考えられる．

以上をまとめると，次のようになる．(1)greedy 割り当てで最適解に達してしまう特殊な状況を除き，greedy 割り当てで最適性の高い解を得られる状況でも，その後の解の改善について，最適性の改善の余地は相対的に小さいものの，従来手法と比較して解の改善速度の向上は観測できた．(2) greedy 割り当てで

最適性の比較的低い解が得られる特定の状況下での効果が観測できた．

なお，同一の入札分散によって生成されたオークション問題内での，本提案アルゴリズムの有効性とgreedy 割り当て結果の性能の関連については，まだ明確な相関を見いだすには至っていない．すなわち，本アルゴリズムが一般的に特定の性質の問題に対して有効に働くか否かは，そのオークション問題の背景にある入札分散モデルが既知であれば予測できる場合もあるが，個別の問題ごとの特性に左右され，具体的に適用してみるまでは本提案アルゴリズムが有効に働くかどうかは明確にはわからない．

本提案アルゴリズムでは，得られる解の効率性がLehmann アルゴリズムの greedy 割り当ての理論的下界以上となることが保証されるものの，その結果が"HC-3" より高いことは保証されない．たとえば，計算時間 1 秒後の近似解の性能について，実験結果上では "HC-3" よりも "XHC-3" のほうが平均値が高いが，この関係が常に成り立つことは，対象となるオークション問題の性質にかかわらず，理論的には保証されない．

5.2 実問題への適用の可能性

本論文では，対象とする問題として，組合せオークションが何度も繰り返され，かつ，そのオークション間の差分が小さい状況を想定することを，1 節で述べた．本節では，この想定に合致し，なおかつ本提案アルゴリズムが有効に機能するような具体的な資源配分問題としてどのようなものが考えられるかについて述べる．

組合せオークションによって最適化が可能な問題としては，資源配分問題の中でも，特に「効用に基づく資源配分問題[22]」があげられる．効用に基づく資源配分問題では，資源配分は配分の効率性を指標として最適性が決められる．また，一般的に，無償廃棄仮説 (free disposal assumption) が適用される．このような性質を持つ実世界の問題としては，無線のチャネル割り当て問題が著名である．近年では，OFDMA(Orthogonal Frequency Division Multiple Access) 技術により，同一無線帯域内に複数のチャネルを作成し同時に利用することが可能となったため，従来の物理的な周波数帯域の制約を超えて，複雑な資源配分問題を解決する必要が出てきている[25]．また，無線規格の 802.11n のように，複数の無線チャネルを同時に利用することで通信速度や接続性を高める技術も一般化しつつあり，異なる性質 (転送速度，到達距離，セキュリティレベルなど) を持った周波数帯を複数同時に利用するような技術も出てきている[18]．このような環境下では，組合せオークションにおける財バンドルが扱うような，複数チャネルの同時割り当てが必要となってきており，この点は組合せオークションの想定する状況と非常に親和性が高い．また，これらの無線技術は，携帯電話や PDA などユーザが通常持ち運ぶ機器で使われる場面が想定されている．このため，ユーザの移動に伴う機器の物理的位置の移動に伴い，無線チャネル割り当ての変更が頻繁に行われることが想定される．

本論文で評価に用いた「手順 1」では，たとえば次のような状況をアプリケーションとして想定している．

シナリオ 1　ある建物内には，共有できない資源 (無線チャネルやセンサーなど) が 200 程度あり，それらを，その建物内にいる複数のユーザたちに効率的に割り当てたい．ここで，ユーザたちは，事前にそれぞれの資源の組合せに対する効用を入札としてシステムに登録しておき，その後は，システムへは単に，その建物への入場と退場のみを通知する．システムは，ユーザの入場・退場の通知を受け，そのユーザに対応する入札の有効・無効を切り替え，その都度，建物内にいるユーザに対する資源割り当てを変更する．

ここで，建物の入り口の大きさは限られているので，同時に入場・退場するユーザの数は必然的に限られることになる．また，ユーザの入場・退場する速度よりも勝者決定を含めたオークションによる資源割り当ての速度が相当に速ければ，ユーザの入場・退場するたびに資源の再割り当てを行うことができ，そのときの資源割り当ての変更の前後でのオークション内の入札の差分も非常に小さくなる．

シナリオ 1 では，ユーザに割り当てた資源を他のユーザに再割り当てすることを妨げる要素を考慮し

ていない．すなわち，あるユーザに割り当てた資源を，次の瞬間に別のユーザに問題なく再割り当てできることを仮定している．一般に，無線チャネルのように物理的な占有を伴いにくい資源であれば，この仮定は大きな問題にならない†3．すなわち，状況 (入札) の変化ごとに組合せオークションを行い，その都度すべての資源を再割り当てするという，最も単純な組合せオークションの適用方法を想定している．

一方で，資源割り当て問題では，資源をある一定期間占有するような状況を想定する必要がある場合についても，すでに広く検討されている．たとえば，この種の問題としては，飛行場の滑走路・誘導路・ゲートの割り当て問題があり，組合せオークションの適用による最適化が検討されてきた [17]．仮に資源の一定期間占有を考慮してシナリオ 1 を拡張するとすれば，シナリオ 1 での資源割当期間を，次の資源割り当て更新時までとせずに一定期間とし，ある程度先までの資源割り当てのタイムスケジュールを組合せオークションにより決定する方法が考えられる．この場合，資源の割り当ての更新サイクルは資源の占有時間よりずっと短く取ることができ，資源の再割り当てが行えなくなるのは，その時点ですでに占有されている資源のみであり，今後の資源割り当て予定については，状況の変化に応じて再割り当てが行える可能性が高い．ただし，シナリオ 1 と比較して，入札を事前にシステムに登録しておくことが難しくなるなどの問題は，別途生じる．

また，特定の資源を継続的に使い続けたほうが利便性が高いような状況を扱うには，入札価格に対して，その直前の資源割り当て状況が反映されるようにすればよい．たとえば，あるユーザが資源 A,B,C をすでに利用しているとき，その次のオークションへの入札としては，それらの資源を含む入札について，より高価な入札額を設定すれば，それらの資源の継続利用の可能性が高まる．この入札額の決定モデルは，効用に基づく入札額の決定となんら矛盾しておらず，入札額を決定するユーザ (あるいはエージェント) 側の負担はあるものの，システム側で資源配分方法において特別の配慮を行う必要はないと考えられる．

本論文で現時点では想定していない問題の種類には，たとえば，a) 将来起きる未知の入札・出品財の変化の予測に基づく現時点での最適な資源割り当て問題，および，b) 割り当ての効率性よりも他の尺度を優先するような資源割り当て問題がある．

問題 a) では，たとえば資源の占有時間の不確実性の考慮 [23] がその例としてあげられる．Grid 環境におけるプロセスへの CPU 資源の割り当て問題では，そのプロセスがどの程度の時間で終了するかを完全に予測することができないため，実際よりも CPU 資源の占有時間が前後する場合がある．このため，ある程度の余裕を持った資源の配分や，まもなく空くことが予想される資源を考慮した投機的な資源配分が有効となる場合がある．また，シーケンシャルオークション (sequencial auction) [1] の拡張によって同様の問題を直接的に扱おうというアプローチもある [13]．

問題 b) では，たとえば，無線チャネル割り当てにおける帯域割り当ての「公平性」を基準とする場合や，セキュリティ・プライバシ上の配慮から，資源の割り当てに複雑な制約が別途適用される場合などがある [24]．これらの種類の問題には，単純な組合せオークションの適用は難しく，組合せオークション以外の手法で対応する必要があると考えられる．組合せオークションが適用困難な種類の資源割り当て問題へは，当然ながら，本提案手法を適用できない．

以上は，本論文で想定するアプリケーションの例を述べたものである．「共通部分を多く含み差分の小さいような組合せオークション勝者決定問題を短時間に解く」という性質をうまく利用できるアプリケーションであれば，本提案アルゴリズムの利点が生かせる可能性がある．一方で，本提案アルゴリズムが適用できるような資源割り当て問題を含むアプリケーションは現時点では限定的であり，本提案アルゴリズム自身が，オークション問題間の差分が小さくなることを保証するものではない†4．

†3 著者らは，異なる無線チャネル間で 1 つの継続的な通信セッションを切断することなく引き継ぐ技術はすでに開発されており，通信セッションの保護のために特定の無線チャネルを占有し続ける必要はすでになくなりつつあると考えている．

†4 複数ラウンドオークションにおける入札の更新数の上限を理論的に保証したものには，たとえば[13] がある．

本論文では，特に勝者決定問題について焦点を絞ってアルゴリズムを提案・評価している．実際には，組合せオークションは勝者決定問題以外にも価格付けなど多くの解決すべき問題があり，本論文でそのすべてを解決しているわけではない．本論文では評価を行っていないが，本アルゴリズムを競り上げ型の組合せオークションの勝者近似手法として用いることは原理的に可能であり，その場合には，価格付け問題に対して一定の貢献が期待できる．しかしながら，オークションの出品者が複数の場合には，組合せ交換問題 (Combinatorial Exchange) [16] となり，出品者に対する報酬の分割方法を別途検討する必要がある．実際のアプリケーションに適用する際には，勝者決定以外の問題に対しての解決方法を本論文の提案手法とは別に用意する必要がある．

6 関連研究

組合せオークションの勝者決定問題に対しては，CASS [5] 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物がある．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 [26] 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．

Guo らは，文献 [11] で，局所探索に基づく手法で，組合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 [11] 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 [11] 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の近似の性能がそれほど高くないため，文献 [11] 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，我々が従来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 らの手法には，本論文で述べた近似勝者の再利用手法は含まれていない．

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，組合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 [26] 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 [11] 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，我々の従来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．

これらの関連研究は，基本的に offline アルゴリズムなので，本論文で提案するような，部分的な近似解の再利用は行われない．

7 おわりに

本論文では，組合せオークションの近似的勝者決定における，部分的な近似解の再利用により，オークションへの入札の変化に対しての再計算を効率化する手法を提案し，その特性を解析した．

本論文では，提案手法の有効性を評価するための方法として手順 1 を定義し，この手順に従ってアルゴリズムの比較評価を行った．手順 1 では，CATS により生成された 1 つのオークション問題を複数に分割し，そのうちで勝者決定に用いない部分を相互にずらすことで「差分の少ないオークション問題」を作成している．このオークション問題の作成方法について

は，既存のオークション問題作成プログラムおよびその問題への評価結果を流用でき，かつ，少ない操作で多数の「差分の少ないオークション問題」列を作成できるという利点がある†5.

一方で，手順 1 によるオークション問題の生成では，特定の入札の出入りを差分として扱えるのみであり，たとえば，1) あるオークションの敗者が次のオークションで自身の入札額を増加させる，あるいは 2) あるオークションで特定の財が落札され，その財に関する入札が次のオークションからは削除される，といった状況を再現するものではない．すなわち，手順 1 で生成されるオークション問題列は，現実的なオークション問題列のすべての性質を網羅するようにはできていない．従って，本論文で示した評価結果はあくまでもこの手順 1 の条件に対するものであり，現実世界の多様なオークション問題列の性質ごとにその結果が異なってくることが考えられる．現実的には，1) の状況は競い上げオークションをモデル化した際には必然的に発生し，2) の状況をモデル化したものとしては，シーケンシャルオークションに関する研究で扱われている例がある．これらの要素を盛り込んだオークション問題列生成手法の効率的な実装とその評価への利用は，今後の課題である．

本論文では，オークション問題を生成するための入札の生成モデルである入札分散として，CATS の L2,L3,L4,L6,L7 の 5 つを用いた．これらは人工的な入札生成モデルではあるものの，これまでに勝者決定問題の性能評価について検討されてきた入札生成モデルのうち重要なものをほぼ網羅しており，少なくとも現時点で標準とされてきたオークション問題の生成方法をある程度カバーしていると考えられる．ただし，1 つのオークション問題中に複数の種類の入札分散が混合される状況は評価の対象としておらず，そのような混合的な入札分散によるオークション問題に対する評価は今後の課題である．

†5　現実的には，CATS 等によるオークション問題の生成には，提案手法等による勝者決定近似にかかる時間よりもはるかに多くの時間がかかるため，(非支配的入札を含まない) オークション問題列を短時間に準備できることは，評価実験そのものを現実的な時間内に終えるために非常に重要な点である．

一般的な組合せ最適化問題のいくつかは，組合せオークション問題に射影することが可能である．本論文では，そのような一般の組合せ最適化問題の射影をオークション問題列としては用いていない．したがって，本論文で示した評価結果は，一般の組合せ最適化問題についてそのまま拡張できるものではない．

本論文で示した評価実験では，オークション間の差分を規定するパラメータとして k を用い，$k = 2, 5, 10, 20, 40$ の範囲での評価結果を示した．評価結果から，$k > 40$ の状況でも本提案アルゴリズムが有効に機能することが期待できる．オークション問題間の差分の変化率はオークション問題列内では (最終回試行の直前を除き) 一定となっているため，たとえば比較的大きな差分と小さな差分が交互に現れるような，差分の変化率が一定でない状況下での本提案アルゴリズムの性能がどのようになるかは，本評価実験の結果からは導出されない．現実問題として，オークションへの入札の変化が，burst 的に起きるのか，定常的に起きるのか，その頻度や変化の入札全体に対する割合がどの程度かは，対象とするアプリケーションによって異なる．具体的な特定のアプリケーションと実装を想定した際の，入札の変化に対するアルゴリズムのロバスト性の評価方法の洗練は，今後の課題である．

ユビキタス環境での資源配分問題では，配分の効率性以外に，より多くの人が利用できる，特定の人が資源を占有し続けないなど，長期的なスパンでの割り当ての公平性の考慮が必要となる．本提案手法のこれらの目的への貢献の可能性の検討は，今後の課題である．

参考文献

[1] Boutiler, C., Goldszmidt, M. and Sabata, B.: Sequential auctions for the allocation of resources with complementarities, in *Proc. of International Joint Conference on Artificial Intelligence(IJCAI1999)*, 1999, pp. 527–534.

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

[3] de Vries, S. and Vohra, R. V.: Combinatorial Auctions: A Survey, *International Transactions in Operational Research*, Vol. 15, No. 3(2003), pp. 284–309.

[4] Epstein, R., Henriquez, L., Catalan, J., Weintraub, G. Y., Martinez, C. and Espejo, F.: A Combinatorial Auction Improves School Meals in Chile : A Case of OR in Developing Counteries, *International Transactions in Operational Research*, Vol. 11(2004), pp. 593–612.

[5] Fujishima, Y., Leyton-Brown, K. and Shoham, Y.: Taming the Computational Complexity of Combinatorial Auctions: Optimal and Approximate Approarches, in *Proc. of the 16th International Joint Conference on Artificial Intelligence* (*IJCAI99*), 1999, pp. 548–553.

[6] Fukuta, N. and Ito, T.: Periodical Resource Allocation Using Approximated Combinatorial Auctions, in *Proc. of The 2007 WIC/IEEE/ACM International Conference on Intelligent Agent Technology*(*IAT2007*), 2007, pp. 434–441.

[7] Fukuta, N. and Ito, T.: Short-Time Approximation on Combinatorial Auctions – A Comparison on Approximated Winner Determination Algorithms, in *Proc. of The 3rd International Workshop on Data Engineering Issues in E-Commerce and Services*(*DEECS2007*), June 2007, pp. 42–55.

[8] Fukuta, N. and Ito, T.: Towards Better Approximation of Winner Determination for Combinatorial Auctions with Large Number of Bids, in *Proc. of The 2006 WIC/IEEE/ACM International Conference on Intelligent Agent Technology*(*IAT2006*), 2006, pp. 618–621.

[9] Fukuta, N. and Ito, T.: Toward A Large Scale E-Market : A Greedy and Local Search based Winner Determination, in *Proc. of The 20th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*(*IEA/AIE2007*), June 2007, pp. 354–363. (poster presentation).

[10] 福田直樹, 伊藤孝行: 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, 電子情報通信学会論文誌, Vol. J90-D, No. 9(2007), pp. 2324–2335.

[11] Guo, Y., Lim, A., Rodrigues, B. and Zhu, Y.: A Non-exact Approach and Experiment Studies on the Combinatorial Auction Problem, in *Proc. of HICSS2005*, 2005, pp. 82.1.

[12] Hoos, H. H. and Boutilier, C.: Solving Combinatorial Auctions using Stochastic Local Search, in *Proc. of the AAAI2000*, 2000, pp. 22–29.

[13] Koenig, S., Tovey, C., Zheng, X. and Sungur, I.: Sequential Bundle-Bid Single-Sale Auction Algorithms for Decentralized Control, in *Proc. of International Joint Conference on Artificial Intelligence*(*IJCAI2007*), 2007, pp. 1359–1365.

[14] Lehmann, D., O'Callaghan, L. I. and Shoham, Y.: Truth revelation in rapid, approximately efficient combinatorial auctions, *Journal of the ACM*, Vol. 49(2002), pp. 577–602.

[15] Leyton-Brown, K., Pearson, M. and Shoham, Y.: Towards a Universal Test Suite for Combinatorial Auction Algorithms, in *Proc. of EC 2000*, 2000, pp. 66–76.

[16] Parkes, D. C., Cavallo, R., Elprin, N., Juda, A., Lahaie, S., Lubin, B., Michael, L., Shneidman, J. and Sultan, H.: ICE: An Iterative Combinatorial Exchange, in *The Proc. 6th ACM Conf. on Electronic Commerce* (*EC'05*), 2005.

[17] Rassenti, S. J., Smith, V. L. and Bulfin, R. L.: A Combinatorial Auction Mechanism for Airport Time Slot Allocation, *Bell Journal of Economics*, Vol. 13(1982), pp. 402–417.

[18] Salem, N. B., Buttyan, L., Hubaux, J.-P. and Jakobsson, M.: Node Cooperation in Hybrid Ad Hoc Networks, *IEEE Transactions on Mobile Computing*, Vol. 5, No. 4 (2006), pp. 365–376.

[19] Sandholm, T.: An Algorithm for Optimal Winnr Determination in Combinatorial Auctions, in *Proc. of the 16th International Joint Conference on Artificial Intelligence*(*IJCAI'99*), 1999, pp. 542–547.

[20] Sandholm, T., Suri, S., Gilpin, A. and Levine, D.: Winner Determination in Combinatorial Auction Generalizations, in *Proc. of the 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems* (*AAMAS02*), 2002, pp. 69–76.

[21] Sandholm, T., Suri, S., Gilpin, A. and Levine, D.: CABOB: A Fast Optimal Algorithm for Winner Determination in Combinatorial Auctions, *Management Science*, Vol. 51, No. 3(2005), pp. 374–390.

[22] Thomadakis, M. E. and Liu, J.-C.: On the efficient scheduling of non-periodic tasks in hard real-time systems, in *Proc. of IEEE Real-Time Systems Symp.*, 1999, pp. 148–151.

[23] Xiao, L., Chen, S. and Zhang, X.: Adaptive Memory Allocations in Clusters to Handle Unexpectedly Large Data-Intensive Jobs, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 15, No. 7 (2004), pp. 577–592.

[24] Xie, T. and Qin, X.: Security-Aware Resource Allocation for Real-Time Parallel Jobs on Homogeneous and Heterogeneous Clusters, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 19, No. 5 (2008), pp. 682–697. (available at http://csdl.computer.org/dl/trans/td/5555/01/l07240.pdf).

[25] Yang, J. and Manivannan, D.: An Efficient Fault-Tolerant Distributed Channel Allocation Algorithm for Cellular Networks, *IEEE Transactions on Mobile Computing*, Vol. 4, No. 6 (2005), pp. 578–587.

[26] Zurel, E. and Nisan, N.: An efficient approximate allocation algorithm for combinatorial auctions, in *Proc. of the Third ACM Conference on Electronic Commerce* (*EC2001*), 2001, pp. 125–136.