

組合せオークションの勝者決定問題に対する 並列分枝限定解法

水戸 将弥 田頭 茂明 藤田 聡

次世代のオークションとして注目を集める組合せオークションの勝者決定問題は NP 困難な問題である。問題の厳密解は重要な意義をもつが、分枝限定法では高速かつ高精度な上界の計算法は知られておらず短時間に解ける問題の規模が限定されてしまう。本研究では分枝限定法では対応できない問題を並列分枝限定解法を用いて高速に解くことを目的とし、1) データ構造の軽量化による分枝限定法の高速度化、2) P2P 通信モデルを採用した並列化をおこなった。また、提案手法のプロトタイプシステムを構築し評価をおこなった。結果として提案手法は、従来のマスター・ワーカモデルと比較して安定して良い性能を得ることができた。

A Parallel Branch and Bound Method for Solving Winner Determination Problem in Combinatorial Auctions

Masaya MITO Shigeaki TAGASHIRA Satoshi FUJITA

Combinatorial auctions have received significant attention in recent years as new type auctions. The winner determination problem in the combinatorial auction is known to be NP-hard. In this paper, we propose a parallel branch and bound system for solving the problem in a short computation time. The basic idea of the proposed system is to design an efficient data structure of the search tree, and to adopt a peer-to-peer style for parallel processing. The result of experiments implies that the proposed system can provide stable performance for several instances compared with the traditional master-worker style system.

1 はじめに

近年のネットワーク技術の発展に伴い、企業や消費者による電子決済やオンラインショッピングなどの電子商取引の利用が拡大している。Yahoo!オークションや eBay のようなオンラインオークションシステムはその代表的なもので、われわれにとってもオークションが身近な存在になってきている。このような状況の中、財の組合せに対する入札を各入札者に許す組合せオークションが、財の間の代替性や補完性

を表現できる新しいタイプのオークションとして高い注目を集めている [2, 5, 8]。組合せオークションシステムを用いて取引するためにはオークションの勝者となる入札を決定する必要がある。しかしながら、組合せオークションの勝者決定問題（以下、WDP）は NP 困難であることが知られている。

実際の商取引を想定した場合、“品物を落札するのに十分な入札値を提示していたにもかかわらず落札できなかった”などのトラブルを未然に防ぐためにも WDP の厳密解を求めることは非常に重要な意義を持つと考えられる。WDP に対する厳密解法には sandholm らの CABOB [5, 7] などがあるが、最も代表的な厳密解法として分枝限定法 [3, 8] があげられる。WDP では高速に精度の高い上界を求める手法が知られておらず、探索木の浅いところでの効率の良い枝刈

広島大学大学院工学研究科情報工学専攻
〒739-8527 東広島市鏡山 1-4-1
Graduate School of Engineering, Hiroshima University
Kagamiyama 1-4-1, Higashi-Hiroshima, 739-8527 Japan
{mito,shigeaki,fujita}@se.hiroshima-u.ac.jp

りが困難である。そのため、あらゆる問題の厳密解を求めることができないためにオークションの適用範囲を制限してしまう。本研究ではまずデータ構造の軽量化による分枝限定法の高速度をおこなう。その後 P2P 通信モデルで局所的な負荷分散をおこなう並列分枝限定解法を提案し、それを利用した組合せオークションシステム CAP (Combinatorial Auction system using rapid Parallel branch and bound method) の構築を目的とする。

以下では 2 節で問題のモデル化と従来の分枝限定法の説明をおこない、3 節で分枝限定法の高速度と並列化の方針を述べ、4 節で並列分枝限定法を実験的に評価し、5 節でまとめる。

2 WDP に対する分枝限定解法

2.1 WDP のモデル化

組合せオークションでは競売人によって複数種類の財 $S = \{x_1, x_2, \dots, x_m\}$ が同時に売りに出される。入札者は財の部分集合に対する入札が許されており、競売人は提示された入札集合の中から競売人の利益が最大となるような入札の部分集合を選択する。選択された入札の入札者を勝者と呼ぶ。本研究では入札者が提示できる入札数を制限せず、また複数の入札に関して勝者となれるものと仮定する。この仮定により一般性を失うことなく入札と入札者を切り離して考えることができることに注意されたい¹。

競売人に提示された入札集合を $B = \{B_1, B_2, \dots, B_n\}$ とする。各入札 $B_i \in B$ は順序対 $\langle S_i, v_i \rangle$ で、空でない S の部分集合 S_i をビットセットとよび、整数 v_i を入札値とよぶことにする。 S_i のサイズを S_i に含まれる財の個数とし $|S_i|$ と記述する。また、 B の部分集合 B' はその要素である任意の入札が空でない共通集合を持たなければ実行可能で、入札 $B_i (i \in B)$ は $B' \cup \{B_i\}$ が実行可能ならば、 $B' (\subseteq B)$ に対して実行可能である。一方、 $B_i \cap B_j \neq \emptyset$ であるような入札 B_j が選択された入札集合 B' に加えられたとき、入札 B_i は入札 B_j によって排除されたという。部分集合 $B' (\subseteq B)$ の利益は B' に含まれる入札の入札値の和で $r(B')$ と記述する。WDP は与えられた入札集合 B から、売り手の利益を最大化するような実行可能な入札集合 B' を構成する問題である。

¹この仮定により、 S に代替的な財が存在した場合、効果的な割り当てがおこなえないことがある。この問題は代替的な入札がともに勝者とならないようにするために代替的な入札が共有する仮財を新たに導入することで解決できる [3]。

WDP に対しては厳密解法を中心として過去さまざまな研究がなされており [5, 7] その代表的な解法として分枝限定法 [3, 8] があげられる。また並列分枝限定解法に関して非常に多くの研究がなされている [1, 4, 9] が、WDP に対して並列分枝限定解法を用いた例は存在しない。

2.2 分枝限定法

WDP に対する分枝限定法では 2 分木を構築しながら左子から深さ優先探索をおこなう。探索木の根は入札が何も割当てられていない状態に対応し、各節点の左子はある 1 つの入札が選択された状態に、右子はある 1 つの入札が選択されなかった状態にそれぞれ対応する。各節点の深さを根から節点までのパスに含まれる枝の数とし、木の深さを葉の深さの最大値とする。このとき木の深さは最大で入札の数 n に等しく、節点の数は $O(2^n)$ である。実行可能解は根から葉までのパス上の節点で選択された入札の集合に対応している。

各節点での枝刈りのための上界値には自明な上界を利用する。自明な上界の計算には実行可能な入札の集合 B^+ に対する予測値を用いる。ここで入札集合 B^+ は選択された入札集合 B' に対して実行可能な入札集合から選択されなかった入札集合 B'' を取り除いたものである。予測値 $h(B^+)$ は以下のように定義される：

$$h(B^+) \stackrel{\text{def}}{=} \sum_{x \in S^+} \left\{ \max_{S_j \ni x, B_j \in B^+} \frac{v_j}{|S_j|} \right\}, \quad (1)$$

ここで S^+ は B^+ に含まれる財の集合である。ある節点が入札 B_i に関する子を持つ場合、入札 B_i を選択した状態に対応する左子の上界値は $r(B') + v_i + h(B^+ - B_i^-)$ によって計算され、選択しない状態に対応する右子の上界値は $r(B') + h(B^+ - \{B_i\})$ によって計算される。ここで入札集合 B_i^- は入札 B_i とそれによって排除される入札からなる集合である。

3 CAP

CAP では分枝限定法の性能改善とその効率の良い並列化によって分枝限定法では対応できない問題を並列分枝限定解法を用いて高速に解くことを目的としている。並列分枝限定法の通信モデルにはマスター・ワーカー・モデル (以下、MW) と P2P モデル (以下、P2P) がある。CAP の運用方法については現在検討

中であるが、オークションの出品者・入札者の協力を得て並列計算をおこなうか、オークションの開催者があらかじめ用意しておいたプロセッサで並列計算をおこなうことを想定している。そのため不均一な環境に対しても動的な負荷分散が可能と考えられる P2P を CAP では採用する。本研究では他のプロセッサの負荷を調べることはおこなわず、“できるだけ粒度の粗い部分問題を株分けする”という非常に局所的な方法による負荷分散方式を提案する。

3.1 節点のデータ量の削減

分枝限定法の性能は、1)木の展開の速度、2)上界計算の速度、3)上界の精度、4)下界(暫定解)の更新速度、に大きな影響を受ける。WDP では自明な上界から精度を上げるためには大きなオーバーヘッドがかかってしまうため、本研究では特に木の展開の速度に注目することにする。下界の更新速度にかかわる入札を順序付ける評価関数 [6] には上界の高速な計算を助けるために、正規化された入札値 $\tilde{v}_i \stackrel{\text{def}}{=} \frac{v_i}{|S_i|}$ を採用し評価値について非増加順に順序付けるものとする。

並列化を前提としたとき、プロセッサどうしで部分問題の通信をおこなう必要があるため、各節点を部分問題に対応させ柔軟な部分問題のやり取りを実現する方法も考えられる。しかしながら、その方法では節点の内部に各入札ごとの排除された回数など問題のサイズに影響を受けるデータを節点に保持しなければならず、木を展開するコストが大きくなってしまい非常に多くの節点を探索する大規模問題に対応できないという欠点も考えられる。そこで、本研究では図 1 のように、探索木の各節点には対応する入札とそれが選択されているか否か、それまでに選択された入札値の和といった問題のサイズに依存しない情報のみを保存することにする。

3.2 通信モデル

分枝限定法を並列化する場合、マスターが最初に分割した部分問題をワーカーに株分けする MW と任意のプロセッサ間で問題を通信する P2P の 2 つの通信モデルが候補となる。両方のモデルに共通する問題通信の注意点として、通信のオーバーヘッドよりも探索時間の方が短いような粒度の細かい問題を通信してはならないという点があげられる。本研究では問題の粒度を探索しなければならない節点の数と

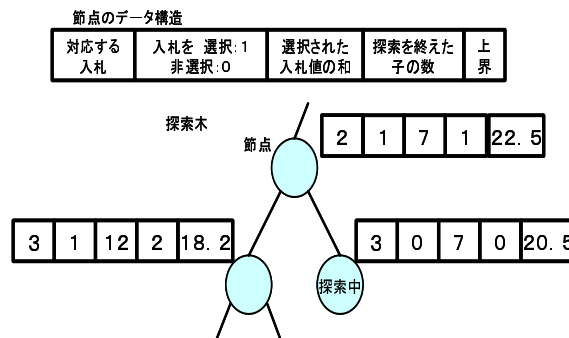


図 1: 探索木のデータ構造.

考える。

双方のモデルで大きく異なるのは問題分割の方法で、MW ではマスターがあらかじめ問題を分割しておきワーカーから要求を受けた時に分割しておいた問題を渡すのに対し、P2P ではプロセッサが任意のプロセッサに問題要求を出しそれを受けたプロセッサが探索している問題を株分けする。負荷分散は通信のオーバーヘッドを抑えつつプロセッサのアイドル時間を減らすことが重要となる。そのためマスターが最後の部分問題を配布し終えたあと全てのワーカーが同時に探索を終了するような部分問題の粒度が理想的であると考えられる。これらから、MW では最初に分割する部分問題の粒度が重要で粗すぎても細かすぎても並列分枝限定法の性能に悪影響をおよぼすと考えられる。一方、P2P では 1 台のプロセッサが非常に粗い粒度の問題を解いており、それ以外のプロセッサが探索を終了していたとしても探索を終了したプロセッサが探索中のプロセッサに問題を株分けしてもらうことができる。そのため P2P では問題の粒度が非常に粗くても負荷分散をおこなうことができると考えられる。

一般に部分問題を株分けする木の深さによって粒度を調節すると考えられるが部分問題の探索空間を予測することは困難である。そのうえ不均一な環境ではたとえ問題の粒度が同じでもプロセッサの性能によって探索時間に差が出てしまう。そのため不均一な環境に対応するためには MW のマスターはワーカーの性能や将来ワーカーで発生する負荷などを考慮したうえで粒度の調節をおこなわなければならない。そこで CAP では粒度が粗くても良い P2P を通信モデルとして採用することにする。

3.3 問題配布戦略

CAP は分散メモリ環境を対象とし、任意のプロセッサに任意のタイミングで問題要求できる P2P モデルを採用する。“できるだけ粗い粒度の部分問題を株分けする”というシンプルなアイデアで他のプロセッサの負荷を調べることはおこなわない非常に局所的な負荷分散方式を提案する。

プロセッサの集合を $\mathcal{P} = \{p_1, p_2, \dots, p_k\}$ とし、問題を読み込むプロセッサ p_1 を CAP とする。CAP はまず、最初に与えられた問題をできるだけ粗い粒度に問題を分割し配布する。具体的に CAP は $2^d - k$ 台に深さ d で株分けした部分問題を割当て、残りのプロセッサに深さ $d - 1$ で株分けした部分問題を割当てる。ここで $d = \lceil \log_2 k \rceil$ である。

各プロセッサは自身に割当てられた部分問題の探索を終えたとき、任意のプロセッサに部分問題の株分け要求を出す。要求を受けたプロセッサはもし探索中ならば部分問題の株分けをおこなう。部分問題を株分けする手法として以下の3つを提案する。

AE(Always Empty method) 各プロセッサは株分けの要求を受けた時点で探索している節点から下の部分問題を株分けし、要求元のプロセッサに渡す。

IDCA(Immediate Delete subproblem of Candidate Area method) IDCA ではプロセッサは Candidate Area (以下、CA) に $l (\geq 1)$ 個の部分問題を蓄えておく。図 2(a) のように各プロセッサは左子を展開し探索するときに、その兄弟である右子を部分問題として蓄えることを繰り返し、株分けの要求を受けたら CA 中の最も浅い部分問題を株分けする。IDCA では 2(b) のように問題を株分けするとその問題を CA から削除し新たな部分問題を蓄え常に株分けできる問題を l 個蓄える。また、探索中に蓄えておいた部分問題に対応する右子に到達したならば、蓄えておいた部分問題を削除し、新たな問題を蓄える。

RCA(Remain subproblem of Candidate Area method) IDCA では常に部分問題を蓄えているため粒度の細かい問題をやり取りしてしまうおそれがある。そこで RCA では図 2(c) のように株分けした問題を CA からすぐには削除せず、株分けした部分問題に対応する節点に到達した時点で CA から削除する。RCA では探索中にもかかわらず問題要求に応えられない可能性も大きくなる。しかしながら、あ

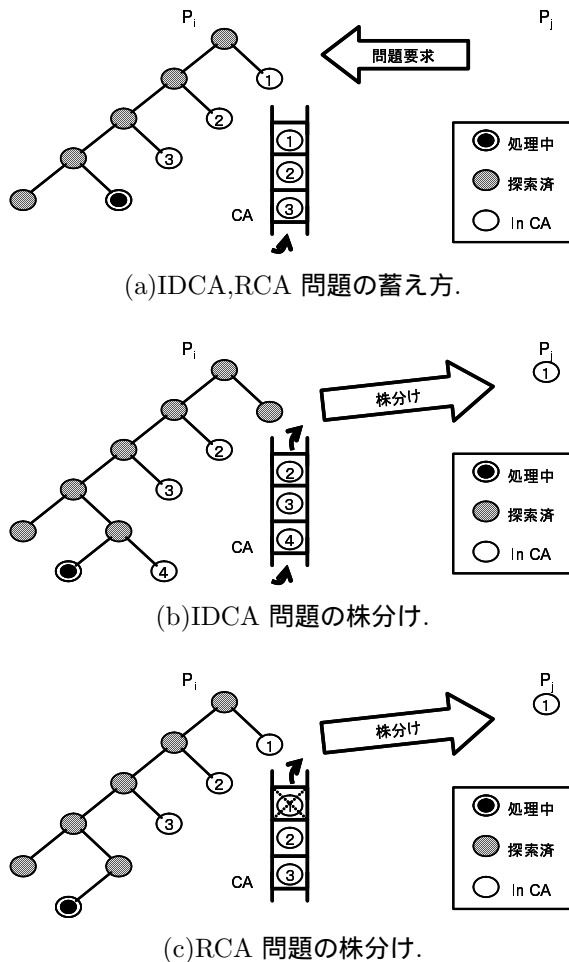


図 2: IDCA と RCA、 $l = 3$ 。

えて細かい粒度の問題を渡さないことによって粗い粒度の部分問題を持つプロセッサとの間での負荷分散が期待できる。

4 実験的評価

4.1 準備

提案した手法の評価のため以下の環境で実験をおこなった：CPU: 500MHz Celeron; Memory: 256MB; OS: Linux kernel 2.4.9; 通信環境: 100base-TX; CAP は mpich を利用して C 言語で実装。実験には文献 [8] で評価に用いられている 2 つの問題インスタンスのクラスを用いた。

Random: 各入札 B_i は S から k_i 個の財を選択し入札値 v_i をつけられる。ここで k_i は $\{1, 2, \dots, m'\}$ からのランダムな値で $m' \leq m$ をみたし、 v_i は

表 1: 節点のデータ量を軽減した分枝限定法の実行時間 (節点が部分問題に対応)(s).

suite	uni200_1	uni200_2	uni200_3
BB	470 (2175)	224 (1035)	925 (4313)
suite	rand1000_1	rand1000_2	rand1000_3
BB	36 (47)	18 (22)	265 (225)

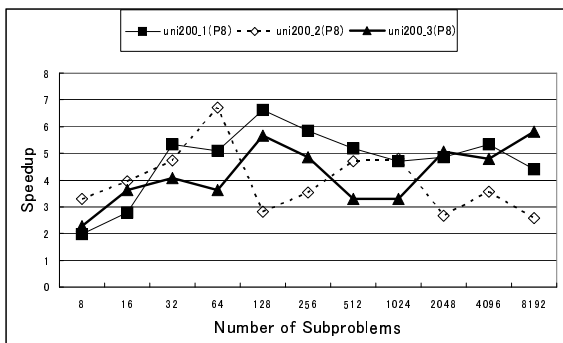


図 3: 部分問題の粒度と MW の性能.

$\{1, 2, \dots, m'\}$ から引かれるランダムな値である。

Uniform: 各入札のサイズを定数 k に固定することにより Random を変形したもの。今回の実験では $k = 3$ としている。

以降では Random の問題インスタンスは rand 入札数_問題 ID、Uniform の問題インスタンスは uni 入札数_問題 ID のように記述する。並列分枝限定法の性能は以下で定義される Speedup によって評価する：

$$Speedup = \frac{1 \text{ 台のプロセッサでの実行時間}}{k \text{ 台のプロセッサでの実行時間}}$$

なお、各インスタンスに対する 1 台のプロセッサでの実行時間は表 1 のとおりである。rand1000_3 を除くインスタンスに対してデータ構造の軽量化の効果が現れている。

4.2 性能評価

部分問題の粒度と MW の性能 MW ではマスターが静的に分割する問題の粒度が実行時間に大きな影響を及ぼす。Uniform の 3 つの問題インスタンスに対する分割する部分問題の数と Speedup の関係を図 3 に示す。この図からインスタンスによって適切な粒度は異なることがわかる、uni200_2 に対してプロセッサの数が 8 の場合では、部分問題を 64 に分けたとき 6.735 倍の Speedup を実現しているのに部分問題を

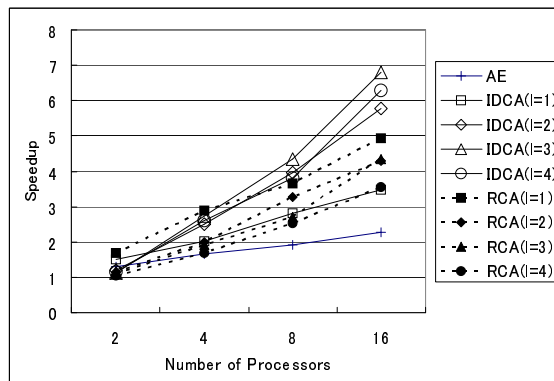


図 4: 問題を蓄える数 l と P2P の性能.

128 にわけたとき 2.82 倍 Speedup しか実現できていないように、問題を分割する木の深さを 1 変えただけで性能が大きく変化する場合も存在している。これらのことから同じクラスのインスタンスでも最適な粒度は異なり、粒度が MW の性能に大きな影響をあたえていることがわかる。

プロセッサが蓄える問題数と性能 uni200 の 3 問に対する P2P での問題配布戦略 AE, IDCA, RCA のプロセッサ数と Speedup の関係を表 4 に示す。IDCA は $l = 3$ 、RCA は $l = 1$ で良い性能を示していることがわかる。これは、探索の多くの時間は木の深いところを探索しているため IDCA はある程度の数の問題を蓄えておかなければ CA の中が粒度の細かい問題ばかりになってしまうからだと考えられる。一方で RCA が $l = 1$ で良い性能を示すのは粒度が粗い問題のみを株分けしているためと考えられる。

MW vs P2P MW と IDCA, RCA を利用した P2P モデルの比較をおこなう。プロセッサ数 8 のときの分割する部分問題の数を 64、128 に設定した MW と IDCA、RCA の各問題インスタンスに対する性能を図 5 に示す。適切な粒度に問題を分割した場合、MW は効率の良い負荷分散をおこなえらる。そのためどのインスタンスに対してもいずれかの粒度の MW が最も良い性能を示している。しかしながら多くのインスタンスで IDCA は 2 種類の MW の粒度の間の性能を示している。図 3 と照らし合わせてみると MW の 13 種類の粒度設定のうち uni200_1 に対しては 11 種類、uni200_2 に対しては 5 種類、uni200_3 に対しては 8 種類の粒度設定よりも高速な Speedup が実現できており、平均的に良い性能を示していることがわかる。

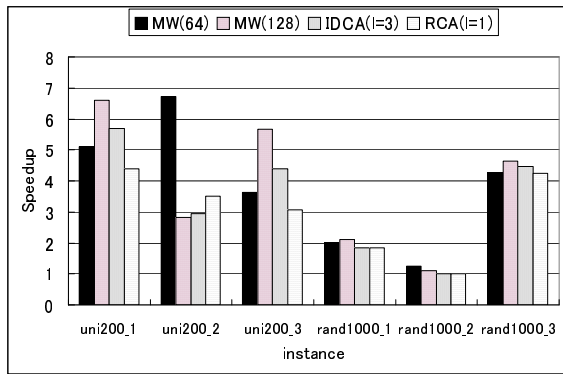


図 5: MW と P2P の性能比較.

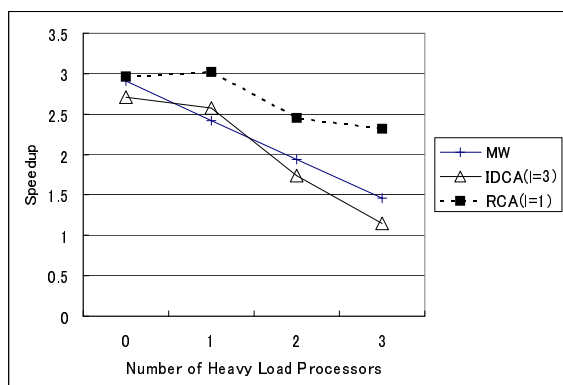


図 6: 不均一な環境での性能.

不均一な環境での頑健性 P2P は動的に負荷分散をおこなっているため MW と比較して不均一な環境でも安定した性能を示すと考えられる。そこで負荷をかけるプログラムを動かし CPU を 50% しか使えない状態の計算機の数を変化させ実験をおこなった。プロセッサ数 4 台、高負荷の計算機 0~3 台の uni200_1 に対する結果を図 6 に示す。高負荷の計算機はマスターもしくはリーダー以外のプロセッサからランダムに選んだ。不均一な環境での評価はより詳しくおこなう必要があると考えられるが、図 6 から RCA が CAP で期待している不均一な環境に対する頑健性を持っているのではないかと期待できる。

5 おわりに

本研究では CAP 構築のために IDCA や RCA といった他のプロセッサの情報を利用しない負荷分散方式を用いた並列分枝限定解法を提案し実験的に評価した。その結果、均一な環境での適切な粒度設定の MW に最良性能では劣るが CAP はパラメータの設

定をおこなわなくても平均的に良い性能を示し、不均一な環境に対する頑健性もあるのではないかという予測を裏付けるデータも得られた。今後の課題として、不均一な環境を主としたより詳細な評価および、問題を要求するプロセッサの決定法や他のプロセッサの過去の探索履歴を利用することによる効率的な枝刈りなどによる CAP の高速化があげられる。

参考文献

- [1] A. de Bruin, G.A.P. Kindervater, and H.W.J.M. Trienekens. Towards an abstract parallel branch and bound machine. *Solving Combinatorial Optimization Problems in Parallel*, LNCS, 1054: 145–170, 1996.
- [2] S. de Vries, R. Vohra. Combinatorial auctions: A survey. *to appear in INFORMS Journal on Computing*, 15, 2003.
- [3] Y. Fujishima, K. Leyton-Brown, Y. Shoham. Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. In *Proc. IJCAI'99*, pages 548–553, 1999.
- [4] D. Henrich. Local load balancing for data-parallel branch-and-bound. *EUROSIM*, 227–234, 1994.
- [5] M. H. Rothkopf, A. Pekeć, and R. M. Harstad. Computationally manageable combinatorial auctions. *Management Science*, 44(8): 1131–1147, 1998.
- [6] M. Mito, and S. Fujita. On Heuristics for Solving Winner Determination Problem in Combinatorial Auctions. *to appear in IAT'03*.
- [7] T. Sandholm, S. Suri, A. Gilpin, and D. Levine. CABOB: A fast optimal algorithm for combinatorial auctions. In *Proc. IJCAI'01*, pages 1102–1108, 2001.
- [8] T. Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135(1-2): 1–54, 2002.
- [9] Y. Shinano, K. Harada, and R. Hirabayashi. Control schemes in a generalized utility for parallel branch-and-bound algorithms. In *Proc. IPSP'97*, pages 621–627, 1997.