A Q-Learning Based Adaptive Bidding Strategy in Combinatorial Auctions

Xin Sui
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, Hong Kong, China
xsui@cse.cuhk.edu.hk

Ho-Fung Leung
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, Hong Kong, China
lhf@cse.cuhk.edu.hk

ABSTRACT

Combinatorial auctions, where bidders are allowed to put bids on bundle of items, are the subject of increasing research in recent years. Combinatorial auctions can lead to better social efficiencies than tractional auctions in the resource allocation problem when bidders have complementarities and substitutabilities among items. Although many works have been conducted on combinatorial auctions, most of them focus on the winner determination problem and the auction design. A large unexplored area of research in combinatorial auctions is the bidding strategies. In this paper, we propose a Q-learning based adaptive bidding strategy for combinatorial auctions in static markets. The bidder employing this strategy can transit among different states, gradually converge to the optimal one, and obtain a high utility in the long-term run. Experiment results show that the Q-learning based adaptive strategy performs fairly well when compared to the optimal strategy and outperforms the random strategy and our previous adaptive strategy in different market environments, even without any prior knowledge.

Categories and Subject Descriptors

K.4 [**Computers and Society**]: Electronic Commerce; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence–Intelligent agents, Multi-agent systems

General Terms

Economics, Algorithm

Keywords

Bidding Strategy, Adaptive , Q-Learning, Combinatorial Auctions

1. INTRODUCTION

ICEC '09, August 12-15, 2009, Taipei, Taiwan

With the increasing demands of computational resources for complex scientific computing problems, such as astronomical calculation and biological calculation, the utilization of computing power provided by centralized and distributed infrastructures is receiving more and more attention from computer scientists and researchers in recent years. Internet is a case example of such infrastructure where different users can use the provided computational resources to perform their own tasks [12]. This kind of resource allocation problem, that is, how to allocate these resources among a group of users, is becoming an important issue. Compared with other approaches for resource allocation[8][20], Internet Auction has a significant advantage that it allocates resources to bidders who value them most and gets the efficient allocation from the view of economics [9].

Among all types of auction, combinatorial auctions, where bidders can bid on combination of items, called "packages", rather than just individual items[10], has received much attention from researchers in both computer science and economics [11]. Combinatorial auctions can lead to more economical allocations of resources than traditional single-item auctions when bidders have complementarities and substitutabilities among them. Such expressiveness will result in an improvement on efficiency, which has been demonstrated in many applications [23][26][3].

There has been a lot of works on combinatorial auctions in the last decade. The most two widely studied problems are the winner determination and the auction design. Winner determination problem is to compute the optimal allocation of resources among bidders and is proved to be NP-hard [25]. Many works have been conducted to solve this problem, including finding optimal solutions and approximate solutions [27][13][34][5][15]. Auction design involves the designing of different protocols for combinatorial auctions, such as single-round versus multi-rounds, sealed-bid versus open-bid and false-name-proof [22][10][33][24][4]. In addition, the computational mechanism design for combinatorial auctions, that is to design a computational feasible while truthful mechanism for combinatorial auctions, is also a vibrant research area [19][21][18][2][14].

Another area of research on combinatorial auctions is the design of bidding strategies. As combinatorial auctions are incorporated with the first-price sealed-bid auction protocol in many applications [10], we are especially interested in bidding strategies in this kind of auction. Unlike the Vickrey Auction, where bidding truthfully is the dominant strategy, no dominant strategy exists in the first-price sealed-bid com-

binatorial auctions [17], which makes the design of bidding strategies a nontrivial problem. Although there are some previous works addressing this problem[7][1][28][29], those proposed strategies are not adaptive, which means that the results are built on some posterior analysis and are very restrictive to certain circumstances.

Based on the principles of Q-learning [31][32], we propose a Q-learning based adaptive bidding strategy in this paper. The bidder adopting this strategy can transit among different states according to his bidding history, and thus perceive and respond to the market. The bidder does not need to have prior knowledge about the markets and the markets are not restricted to certain types. In other words, our strategy generalizes well. Furthermore, the proposed Q-Learning based adaptive strategy is not restricted to combinatorial auctions. Actually, such a strategy can be used in any repeated auctions, which will be further discussed in the conclusion section. Through simulations, we show that the Q-learning based adaptive strategy performs well and generates high utilities in different markets when compared with the random strategy and our previous adaptive strategy. We also compare the performances of the Q-learning based adaptive strategy and the previous adaptive strategy in detail. In addition, we also show that the bidder using this strategy can converge quickly to the optimal state in different markets, and thus be capable of learning and adapting, even without any prior knowledge.

This paper is structured as follows. In Section 2 we present some related work on bidding strategies in combinatorial auctions. In Section 3 we present the combinatorial auction model and the Q-learning model. In Section 4, we describe the proposed Q-learning based adaptive bidding strategy. In Section 5, we show some experiment results and make some discussions, and finally, In Section 6, we conclude this paper and highlights some possible future work.

2. RELATED WORK

In first-price sealed-bid combinatorial auctions, a bidder has exponential number of bundles to bid on. The problem of deciding which bundles to choose and how much to bid for them are referred to as the bundle strategy and the price strategy respectively. In this section, we present a survey on both strategies in combinatorial auctions.

The work of Berhault et al. [7] focus on the bundle strategy in single-round combinatorial auctions. They use combinatorial auctions to allocate unexplored terrains to robots distributed in a large field and propose four bidding strategies: Three-Combination, Smart-Combination, Greedy and Graph-Cut. Through experiments they show that combinatorial auctions achieve better efficiencies than single-item auctions and generate good results compared to optimal centralized allocations. They also show the influences of bundle strategies on team performances, where the Graph-Cut strategy clearly outperforms the other three.

An et al. [1] also study the bundle strategy in single-round combinatorial auctions. They propose two bundle strategies: Internal-Based and Competition-Based. Bidders using the former bid on bundles for which they have higher valuations, while bidders using the latter bid on bundles for which they have higher ratios of valuations to their competitors' according to their prior estimations. Simulation results show that wise bidders using these two strategies outperform naive bidders, who only submit single-item bids. They also analyze the impact of these two strategies on the auctioneer's revenues in combinatorial auctions.

Schwind et al. [28] attempt to solve the computational resource allocation problem with the multi-round combinatorial auctions. They study the situation where bidders use virtual currencies, which are obtained by selling idle resources, to get accesses to computational resources needed for accomplishing their own tasks. They propose price strategies for two kinds of bidders: impatient bidders and quantity maximizing bidders. Experiment results show that for the first kind of bidders, it is better to bid with high prices to get quick accesses to the resources, while the second kind of bidders had better bid with low prices and keep on waiting for resources.

In our previous work [29], we also use the multi-round combinatorial auctions to distribute computational resources among a group of users. We propose an adaptive bidding strategy for bidders in static markets where the ratios of supplies and demands are kept constant during the whole procedure of the auction. The bidder adopting this strategy can adjust his profit margin constantly according to his bidding history, and finally converge to the optimal one even without prior knowledge. Through simulations, we show that the adaptive strategy outperforms other strategies and generate high utilities when compared with the optimal strategy in several static markets.

3. PRELIMINARIES

3.1 Combinatorial Auctions

In this paper, we consider a scenario where the first-price sealed-bid combinatorial auctions are employed to distribute computational resources among a group of users. Suppose m type of resources are provided by a resource manager (auctioneer) to n users (bidders). For each type of resource $j \in \{1, 2, ..., m\}$ or M, the capacity c_j denotes the total number of units that are available.

At any time during the auction, each user $i \in \{1, 2, ..., n\}$ may need certain types of resources to perform his current task, and for each type of resource j, the maximum number of units that he can require for is d_j. He can submit a sealed-bid $b_i = (T, p_i(T))$, where $T = (t_1, t_2, ..., t_m)$ is a resource bundle, with t_j being the number of units that resource j is requested by i and satisfying $0 \leq t_j \leq d_j$, $\forall j \in M$, and $p_i(T)$ is a positive number denoting the price i will pay for getting T.

After receiving bids from all users, the resource manager solves the winner determination problem, which is given by:

$$\begin{aligned} \max & \sum_{i=1}^{n} \sum_{T \subseteq M} p_i(T) x_i(T) \\ s.t. \quad & \sum_{i=1}^{n} \sum_{T \subseteq M, T \ni j} x_i(T) \leq 1 \ \ \forall j \in M \\ & x_i(T) \in \{0, 1\} \end{aligned} \tag{1}$$

where $x_i(T) = 1$ if bidder i is allocated T.

Each winning user i pays $p_i(T)$, gets accesses to the resources, performs his own task, and returns the access control back to the resource manager. We refer to the process from the beginning of bids submission to the end of access control return as a *round* of a combinatorial auction. Because the resources are reusable, the combinatorial auction

can be repeated for multiple rounds before closed by the resource manager.

We list some assumptions used in this paper. First, we assume that the information available to each bidder is his past bidding information only, e.g., his bidding bundles, bidding prices and bidding results of wins or loses. Any information of other bidders is not disseminated. Second, we assume that the combinatorial auction market is static. A combinatorial auction market is said to be static if ratios of supplies and demands for different types of resources are kept constant during whole process of the auction. Finally, we only consider the simplest case that each bidder only submit a single bid per round, which means that no bidding language is used. The winner of the previous round submits a new bid, while each loser continues to submit the lost bid. However, a same bid will be dropped if it has been submitted for τ consecutive rounds, which means that the bidder has a limited patience on waiting.

3.2 Q-Learning

Q-learning [31][32] is a reinforcement learning [16][30] used for solving tasks modeled by Markov Decision Processes. It works by learning an action-value function that gives the expected utility of taking a given action in a given state and following a fixed policy thereafter. The most two significant strengthes of the Q-learning are that it can compare the expected utility of the available actions without modeling the environment and it can be used on-line. Q-Learning is well suited for solving sequential decision problems, where the utilities of actions depends on a sequence of decisions made and there exists uncertainty about the dynamics of the environment.

In the Q-learning framework, the environment which the agent interacts with, is a finite-state, discrete-time, stochastic dynamic system. The interaction between the agent and the environment at time t consists of the following sequence:

- The agent senses its state $s_t \in S$.
- Based on the state s_t, the agent choose an action $a_t \in A$.
- With probability of $Pr_{s_t,s_t^*}(a_t)$, the agent transmits to a new state of $s_t^* \in S$.
- The environment gets a reward $r(s_t, a_t)$ as the consequence of agent choosing a_t at s_t.
- The reward $r(s_t, a_t)$ is passed back to the agent and the process is repeated.

The objective of the agent is to determine an optimal policy π^*, that will maximize the total expected discounted reward, which is given by:

$$V^{\pi}(s) = E\{\sum_{t=0}^{\infty} \beta^t r(s_t, \pi(s_t)) | s_0 = s\} \tag{2}$$

where E is the expectation operator, $0 \leq \beta < 1$ is a discounted factor, and π is a policy $S \rightarrow A$. $V^{\pi}(s)$ is often called the value function of state s.

Recall $Pr_{s_t,s_t^*}(a_t)$, equation 2 can be rewritten as:

$$\begin{aligned} V^{\pi}(s) =& E\{r(s_0, \pi(s_0)) | s_0 = s\} + \\ & E\{\sum_{t=1}^{\infty} \beta^t r(s_t, \pi(s_t)) | s_0 = s\} \\ =& R(s, \pi(s)) + \beta \sum_{s^*} Pr_{s,s^*}(\pi(s)) V^{\pi}(s^*) \end{aligned} \tag{3}$$

where $R(s, \pi(s)) = E\{r(s, \pi(s))\}$ is the mean of $r(s, \pi(s))$.

Equation 3 indicates that the value function of state s can be represented in terms of the expected immediate reward of the current action and the value function of the next state.

According to Bellman's optimality criterion [6], there is always an optimal policy π^* that satisfies equation 3. The objective is to find out the optimal policy without prior knowledge about $R(s, \pi(s))$ and $Pr_{s,s^*}(\pi(s))$. For a policy π, a Q value is defined as:

$$Q^{\pi}(s, a) = R(s, a) + \beta \sum_{s^*} Pr_{s,s^*}(a) V^{\pi}(s^*) \tag{4}$$

which is the expected discounted reward for executing action a at state s and then following policy π thereafter.

Let

$$\begin{aligned} Q^*(s, a) &= Q^{\pi *}(s, a) \\ &= R(s, a) + \beta \sum_{s'} Pr_{s,s^*}(a) V^{\pi^*}(s^*) \end{aligned} \tag{5}$$

Then we can get

$$V^*(s) = \max_a [Q^*(s, a)] \tag{6}$$

Thus the optimal value function V^* can be obtained from $Q^*(s, a)$, and in turn $Q^*(s, a)$ may be expressed as:

$$\begin{aligned} & Q^*(s, a) \\ & = R(s, a) + \beta \sum_{s'} Pr_{s,s^*}(a) [\max_{a^*} Q^*(s^*, a^*)] \end{aligned} \tag{7}$$

where s^* is the new state reached with probability of $Pr_{s,s^*}(a)$ when doing action a at state s.

The Q-learning process tries to find $Q^*(s, a)$ in a recursive way using $(s, a, s^*, R(s, a))$, and the rule is:

$$Q(s, a) = (1 - \alpha) \cdot Q(s, a) + \alpha \cdot [r(s, a) + \beta \cdot \mathbb{M}] \tag{8}$$

where $\mathbb{M} = \max_{a^*} Q(s^*, a^*)$ and $0 \leq \alpha < 1$ is the learning rate.

4. THE Q-LEARNING BASED ADAPTIVE BIDDING STRATEGY

As described in the above section, each winner need to pay the price he has bid to get resources, and each loser pays nothing. The utility of bidder i is computed as follows:

$$u_i(T) = \begin{cases} v_i(T) - p_i(T) & \text{if } i \text{ wins} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where $u_i(T)$, $vi(R)$ and $p_i(T)$ are his valuation, his winning utility and his bidding price for bundle T respectively.

When putting a bid on a resource bundle, a rational bidder will use a positive value which is less than his valuation for that bundle, otherwise he will get a negative utility when winning. That is to say, if the valuation of bidder i for bundle T is $v_i(T)$, then his bidding price $p_i(T)$ is $p_i(T) = (1 - pm_i) \times v_i(T)$, where $0 < pm_i < 1$. We refer to pm_i here as bidder i's profit margin. Combined with equation 9, the utility of bidder i is hence:

$$u_i(T) = \begin{cases} pm_i \times v_i(T) & \text{if } i \text{ wins} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

Now, there is a dilemma faced by a bidder on deciding what profit margin to use when bidding for resource bundles: bidding with a low profit margin will increase his winning

probability, but decreases his winning utility at the same time; bidding with a high profit margin will lead to a high winning utility, but under a high risk of losing. If a bidder is able to get some prior knowledge about the market environment, e.g., the ratio of supplies and demands, he may probably use these information to help his decision. For example, a bidder who has prior knowledge about the market that the there are more supplies of resources than demands will use a high profit margin when bidding. This is because in this kind of market, competitions for resources among bidders are not severe and bidding with a high profit margin will lead to a high winning utility while the probability of winning is almost unaffected. However, by our assumption, the information available to each bidder is very limited. Our aim here is to design an adaptive bidding strategy based on the principle of Q-learning such that the bidder adopting it can perceive and responds to the market in a timely manner even with limited information, and this is the main contribution of this work.

4.1 Basic Concepts

We first introduce some basic concepts used in the Q-learning based adaptive bidding strategy.

DEFINITION 1. A *bidding record* of a bid b for bidder i is a tuple $br_b = (T_b, v_i(T_b), pm_b, wait_b, win_b)$, where T_b is the requested bundle in b, $v_i(T_b)$ is i's valuation for T_b, $wait_b$ is the number of rounds the bidder has kept on waiting before b is accepted or dropped, and win_b is an integer of 0 or 1 indicating the bidding result for b, that win_b equals to 1 if b is finally accepted, otherwise 0.

From the definition, we can see that the maximum value for $wait_b$ is τ and the minimum value for it is 0. In the former case, the bidder keeps on waiting for τ rounds and finally dropped the bid, and in the latter case, the bid b is accepted at the first round when submitted by the bidder. In addition, the value of win_b can also be inferred from the value of $wait_b$, that $win_b = 0$ if and only if $wait_b = \tau$.

DEFINITION 2. A *bidding history* of a bidder, denoted as cbh^ρ, is the sequence of the most recent ρ bidding records. However, we say that it is *consistent* if and only if all these ρ bidding records share the same profit margin.

Suppose $\rho > 1$, every time when a bid is accepted or dropped, the bidding history is updated that the oldest bidding record is removed from the bidding history and the newest one is inserted into the bidding history. However, the bidding history is said to be consistent only when the all the containing bidding records use the same profit margin. If a bidder uses a fixed profit margin for all bidding records, then each history is consistent; if he never uses the same profit margin for two consecutive bidding records, then none of his bidding history is consistent.

DEFINITION 3. A *state* of a bidder, denoted by s, is the profit margin currently used by this bidder.

During the auction, the bidder can change his profit margin by either increasing or decreasing, which will trigger the transition of his state.

DEFINITION 4. A *action* of a bidder at the state of s, denoted by a, is a non-zero real number, by which his state will transit from s to $s^* = s + a$ for the following rounds before the next transition, where the new state s^* satisfies that $0 < s^* < 1$.

From the definition, we can see that every time when an action a is made at the state of s, the bidder will transit to a new state s^* because of the non-zero property of a, which means the bidder will use a new profit margin for the following rounds before the next transition, with the constraint that $0 < s^* < 1$.

DEFINITION 5. The *reward* that a bidder receives from the environment when making an action of a at the state of s, denoted as $r(s, a)$, is defined as:

$$r(s,a) = s^* \times \frac{\sum_{br_b \in cbh^{\rho^*}} win_b}{\sum_{br_b \in cbh^{\rho^*}} (win_b + wait_b)} \tag{11}$$

where s^* is the new state when choosing the action of a at the state of s, and cbh^{ρ^*} is the consistent bidding history formed when the bidder remains at state s^*.

4.2 The Core Algorithm

Based on the basic concepts defined above, we will describe the core algorithm of the Q-learning based adaptive strategy. The main idea is that every time when a consistent bidding history is formed, the bidder computers the reward, updates the Q-values, and chooses an action according to the updated Q-values. By doing is iteratively, the bidder will transit among different states, and finally approach to the optimal state, which maximizes the bidder's accumulated utility in the long term run. We first give a notation and a definition.

NOTATION 1. We say that state s is θ close to state s^*, denoted as $s <_\theta s^*$, if and only if $|s - s^*| < \theta$.

DEFINITION 6. The Q-value of the state set-action pair (L, a), where $L \subseteq S$, is defined as the average Q-values of pair (s, a), where $s \in L$. That is:

$$Q(L,a) = \frac{1}{|L|} \sum_{s \in L} Q(s,a) \tag{12}$$

The adaptive strategy is illustrated in Algorithm 1. We use s to denote the bidder's current state, which is obtained by doing action a' at state s', and use s^* to denote the next state if action a' is carried on s. We also use r and r' to denote the reward obtained by the bidder when reaching the state of s' and s respectively. In addition, we use a variable of $V_{Q(s,a)}$ to indicate the number of times that the state-action pair (s, a) has been visited, which is initialized with 0 at the beginning of the algorithm.

The adaptive strategy is illustrated in Algorithm 1.

At the beginning, state set S and action set A are initialized with $\{s_{ini}\}$ and $\{+\theta, -\theta\}$ respectively, and then some variables used in the algorithm are also initialized (line 1 and 2). During the process the auction, the bidder remains at the current state of s, and transit to a new state every time when 1) a new consistent bidding history with length ρ is formed and 2) θ is greater than the threshold value of ϵ. The bidder first computes the reward of the previous state-action pair $r(s', a')$ according to equation 11 (line 6), then

Algorithm 1 The Core Algorithm

1: $S \leftarrow \{s_{ini}\}$, $A \leftarrow \{+\theta, -\theta\}$
2: $r' = 0$, $s' = s = s_{ini}$, $Q(s, +\theta) > 0$.
3: **while** auction does not finish **do**
4: Keep at the state of s
5: **if** a new cbh^{ρ} is formed and $\theta > \epsilon$ **then**
6: Set $s' = s$ and compute $r = r(s', a')$.
7: Update $Q(s', a')$ with equation 13 and $Q(s', a')$++.
8: $q = Q(s, a')$, update $Q(s, a')$ with equation 14 and $Q(s, a')$++.
9: **if** Decrease$\theta() = true$ **then**
10: $\theta = \theta / \gamma$
11: **if** $\theta \notin A$ **then**
12: $A \leftarrow A \bigcup \{+\theta, -\theta\}$
13: **end if**
14: **end if**
15: **if** $V_{Q(s,a')} > 0$ and $V_{Q(s,-a')} > 0$ **then**
16: $a = arg\max_{a^* \in A, |a^*| = \theta} Q(s, a^*)$
17: **else if** $V_{Q(s,a')} = 0$ and $V_{Q(s,-a')} = 0$ **then**
18: $a = arg\max_{a^* \in A, |a^*| = \theta \cdot \gamma} Q(s, a^*)/\gamma$
19: **else**
20: **if** $r \geq r'$ **then**
21: $a = a'$
22: **else**
23: $a = -a'$
24: **end if**
25: **end if**
26: $Q(s, a') = q$ and $Q(s, a') - -$.
27: $s = s + a$, $r' = r$
28: **if** $s \notin S$ **then**
29: $S \leftarrow S \bigcup \{s\}$
30: **end if**
31: FillUpQValues ().
32: **end if**
33: **end while**

updates the Q-values for pairs (s', a') and (s, a') using the following equations (line 7 and 8):

$$Q(s', a') = \begin{cases} (1-\alpha) \cdot Q(s', a') + \alpha \cdot [r(s', a') + \beta \cdot \mathbb{M}] & \text{if } V_{Q(s',a')} > 0 \\ r(s', a') & \text{otherwise} \end{cases} \tag{13}$$

where $\mathbb{M} = \max_a Q(L, a)$ for $L = \{s^{\#} | s^{\#} <_{\theta} s\}$ and

$$Q(s, a') = \begin{cases} (1-\alpha) \cdot Q(s, a') + \alpha \cdot [r(s', a') + \beta \cdot \mathbb{M}] & \text{if } V_{Q(s,a')} > 0 \\ r(s', a') & \text{otherwise} \end{cases} \tag{14}$$

where $\mathbb{M} = \max_a Q(L, a)$ for for $L = \{s^{\#} | s^{\#} <_{\theta} s^*\}$ and $s^* = s + a'$.

Note that here we save the value of $Q(s, a')$ to a variable of q before updating. This is because that actually, $r(s', a')$ should be used to update $Q(s', a')$ rather than $Q(s, a')$. Updating $Q(s, a')$ with $r(s', a')$ means that we transcendentally believe that doing action a' at the state of s will bring the bidder the same reward as that of doing action a' at the state of s'. The updated $Q(s, a')$ will be used for deciding the action at the state of s, after which it is restored to original value of q (line 26).

Then the bidder check the decreasing condition for θ and decrease its value if necessary (line 9 to 14), and choose an action according to the following rules (line 15 to 25): I) if both state-action pairs of (s, a') and $(s, -a')$ have been visited before, the bidder will choose the action with the higher Q-value; II) if neither of them have been visited before, which means that θ has just been decreased, the bidder will first choose the action assuming that decrease of θ does not happen, and then decrease the chosen action by γ; III) otherwise, if $r \geq r'$, which means that doing action a' has led to an increase on reward, then we continue this action; else if $r < r'$, which means that doing action a' has led to a decrease on reward, then we opposite oppose this action.

After that, the bidder transits to the new state according to the selected action (line 27), updates the state set S if necessary (line 28 to 30) and call the function of FillUpQ-Values (line 31), by which a transition of state has finished. Such a transition will be repeated until θ is smaller than a threshold ϵ, after which the bidder will remain at that state for all subsequent rounds until the auction finishes.

Next, we will introduce the two functions of Decreaseθ and FillUpQValues in detail.

4.3 Function of Decreaseθ

The value of θ is decreased to make sure that the bidder's state can be more approached to the optimal state.

DEFINITION 7. The *state history*, which is denoted as sh, is a sequence of λ real numbers, in which the kth element, sh^k, is the bidder's kth most recent state.

DEFINITION 8. The θ *history*, denoted as θh, is a sequence of λ real numbers, in which the kth element θh^k, is the action used when when the bidder transits from sh^{k-1} to sh^k.

NOTATION 2. We say that $s \Rightarrow \pi$ if 1) $s < \pi$ and the next action of the bidder $a > 0$ or 2) $s > \pi$ and the next action of the bidder $a < 0$.

The function of Decreaseθ is given in Algorithm 2. At first, we compute the mean value of the elements in sh (line 1), then for each element we check whether the distance between sh^k and $mean$ is no more than θh^k and use a 0 or 1 variable ω^k to indicate the result (line 2 to 7). On deciding whether to decrease θ, we check three conditions (line 8): the first one checks whether at least ϕ elements in sh are close to $mean$ in terms of the action chosen then, by which we regard $mean$ as an approximation of the optimal state, and the second and the third ones together guarantee that the optimal state can be further approached if θ is decrease. If all conditions hold, $true$ is returned.

Algorithm 2 Function: DecreaseΘ

1: Compute $mean = \frac{1}{\lambda} \sum_{k=1}^{\lambda} sh^k$.
2: **for** $k = 0$ to λ **do**
3: $\omega^k = 0$
4: **if** $|sh^k - mean| \leq \theta h^k$ **then**
5: $\omega^k = 1$
6: **end if**
7: **end for**
8: **if** $\sum_{k=1}^{\lambda} \omega^k \geq \phi$ and $\omega^1 = 1$ and $s \Rightarrow mean$ **then**
9: return $true$
10: **end if**

4.4 Function of FillUpQValues

As the name denotes, we fill up the Q-values for some state-action pairs in this function according to others values

in the Q-matrix. This is because according to our definition of reward in equation 11, for two state-action pairs of (s_1, a_1) and (s_2, a_2), their rewards, and also their Q-values if combined with the definition in equation 7, should be the same if $s_1 + a_1 = s_2 + a_2$ when ρ^* is infinitely large. In addition, the Q-values for some pairs should be close to those for some others, e.g. the Q-values for state-action pairs of $(0.8, 0.04)$ and $(0.75, 0.1)$ should be close to each other, although $0.8 + 0.04 \neq 0.75 + 0.1$.

Based on the above ideas, we illustrate the function of FillUpQValues in Algorithm 3. For each state in $s \in S$ and each action in $a \in A$, if its Q-value $Q(s, a)$ has not been visited before, then we compute the state set whose elements are θ close to $s+a$ (line 4). If such state set is not empty, we first approximate the Q-value of the state-action pari (s, a) with that of this state set-action pair (L, a) (line 6), and then add the value of $V_{Q(s,a)}$ by 1 (line 7).

Algorithm 3 Function: FillUpQValues

```
1: for each s ∈ S do
2:     for each a ∈ A do
3:         if V_{Q(s,a)} = 0 then
4:             L = {s^# | s^# <_θ s + a}
5:             if |L| > 0 then
6:                 Q(s, a) = Q(L, a)
7:                 V_{Q(s,a)}++.
8:             end if
9:         end if
10:    end for
11: end for
```

5. SIMULATIONS

To evaluate the performance of the Q-learning based adaptive strategy, we conducted two sets of experiments. In the first set of experiments, we approximate the optimal strategy in different types of markets with a set of fix strategies. A fix strategy is a strategy that keeps the bidder remaining at a same state during the process of the auction. In the second set of experiments, we compare the performances of the adaptive strategy (AS), the Q-learning based adaptive strategy(Q-AS), the random strategy (RS), and the optimal strategy (OPT). The Adaptive strategy is a strategy that we proposed in our previous work [29], which also achieve good results in different markets. The random strategy is a strategy that the bidder randomly transit among different states for different bidding records. The optimal strategy is the strategy that artificially generated according to the results of the first set of experiments. As the adaptive strategy also performs well in different markets, we compare the adaptive strategy and the Q-learning based adaptive strategy in detail. In addition, we also show the typical convergency process of the state of the bidder who uses the Q-learning based adaptive strategy in a single run in different markets.

5.1 Experiment Setup

In our experiments, each combinatorial auction is repeated for 500 rounds and an iteration of 500 rounds is referred to as a *run*. Motivated by other works [1][28], in each run, we have one test bidder using strategy X and others bidding their true valuations. Here, X can be the adaptive strategy, the random strategy, the Q-learning based adaptive strategy or the optimal strategy. The performances of different strategies are compared through accumulated utilities of the test bidder in 100 runs.

Settings of these experiments are as follows. A group of $n = 60$ users are competing for $m = 4$ types of resources provided by a resource provider. For each bidder, numbers of units that he can request for different resources are integers randomly drawn from uniform distributions of [0, 3], [0, 2], [0, 2] and [0, 1]. At the beginning of each run, each bidder initializes his valuations for resource bundles. His valuations for single unit of different resources are real numbers randomly drawn from uniform distributions of [3, 6], [4, 8], [4, 8] and [6, 10]. For a resource bundle T, which contains more than one type of resource, a synergy seed, $syn(T)$ is randomly drawn from a uniform distribution of [-0.2, 0.2], and his valuation for that bundle is the product of sum valuations of individual resources and $1 + syn(T)$. Positive synergy seed means there are complementarities among resources and negative synergy seed means there are substitutabilities among them.

In our settings, because the expected total demands of users are fixed, we modify the supplies of resources and use a capacity factor cf to denote different market types: if the ratio of total supplies and demands in the market is equal to $cf : 1$, we say that this is a $cf : 1$ market. For example, when the capacities of resources are 90, 60, 60 and 30, the total supplies and expected demands are equal. In this case, we say that the market is a $1 : 1$ market.

Parameters used in experiments are showed in Figure 1.

Parameter	Value	Description
τ	3	Maximum waiting round
ρ	5	Length of a bidding history
s_{ini}	0.05	Bidder's initial state
θ	0.1	Initial value for θ
ϵ	0.01	Threshold for θ to stop transition
λ	10	Length of a profit margin history
ϕ	7	Threshold to decrease θ
γ	1.4	Degree of decrease for θ

Figure 1: Parameters used in experiments

In addition, we use fixed learning rate of $\alpha = 0.2$ and discount rate of $\beta = 0.1$.

5.2 Experiment Results and Analysis

In the first set of experiments, our aim is to approximate the optimal strategy in different types of market with a set of fixed strategies. Here, we use 19 different fixed strategies in which the bidder's state is kept at s_1, s_2, ... s_{19}, where $s_l = l \times 0.05$ for $l = 1, 2, ...19$.

Figure 2 shows the results for the first set of experiments. Each red point represents the fixed strategy that performs best in that type of market, e.g. the red point $(0.7, 0.2)$ means that in the $0.7 : 1$ market, the fixed strategy that keeps the bidder at the state of 0.2 performs best among all fixed strategies. From this figure, we can see that the less competitive the market is for resource consumers, the higher value of the state that the bidder is kept at. For example, in the $1.2 : 1$ market, where bidders face few competitions from others, the strategy that the bidder's state is kept at 0.95 performs best; while in the $0.5 : 1$ market, where bidders face fierce competitions from others, the strategy that the bidder's state is kept at 0.15 performs best. This accords

with our intuition that in a market short of competition, it is better for a bidder to remain at the state with a high profit margin to gain a high utility, while in a market full of competition, it is better for him to remain at the state with a low profit margin to beat others.

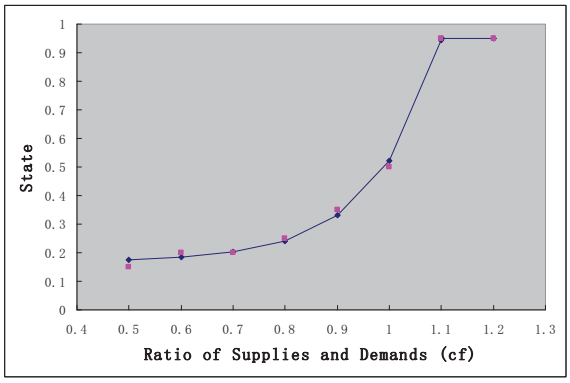

Figure 2: Estimation of The Optimal Strategy

Based on these red points, we estimate the optimal strategies in different markets with a regression method. We use a piecewise function $s_{opt}(cf)$ to fit the red points, which is given by:

$$s_{opt}(cf) = \begin{cases} a \times b^{cf} + c & cf < d \\ e & cf \geq d \end{cases} \quad (15)$$

The result of the regression is that $a = 0.0001382$, $b = 2561.574$, $c = 0.1683$, $e = 0.95$ and $d = 1.109873$, which is shown by the blue line in Figure 2. We can see that it fits the red points very well, and in the following, when talking about a market with type cf, we will use the function value as the optimal state.[1]

Figure 3 shows the results for the second set of experiments. Here, performances of three other different strategies are compared through the ratio of the accumulated utilities achieved by them and that achieved by the optimal strategy generated according to equation 15. Note that in equation 15, the maximum value for state that the bidder can remain at is 0.95, so to make the comparison fair, we also set up the same upper bound of 0.95 for these three strategies.[2]

From Figure 3, we can see that both the adaptive strategy and the Q-learning based adaptive strategy perform well, and outperform the random strategy much in different market environments. What is more, the Q-learning based adaptive strategy has an improvement on performances to the adaptive strategy from 2% to 5%. As described above, the optimal strategy artificially generated according to a set

[1]Here, an exponential function is used as the left part of the regression function, and actually, it does not matter too much if we use other functions. This is because in the second set of experiments, we never use equation 15 to estimate the optimal strategy in a market whose cf falls out of [0.5, 1.2], and the estimated optimal strategy will not vary much if other fit functions are used.

[2]Actually, setting this upper bound does not affect the performance of the adaptive strategy. This is because without this constraint, when the optimal profit margin is a value infinitely close to 1, the profit margin generated by the adaptive strategy is also very close to 1, and the bidder using the adaptive strategy does not losing utility at all.

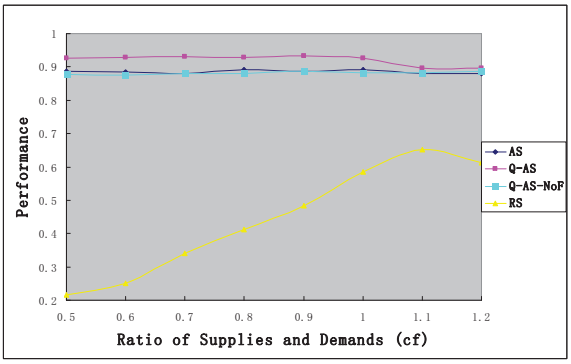

Figure 3: Utilities achieved by the test bidder using strategies of AS, Q-AS and RS

of fixed strategies, so the bidder using this strategy can be regarded as having prior knowledge about the market environment and is able to remain at the optimal state to obtain a high utility. On the contrary, the bidder using the random strategy can be regarded as not having any prior knowledge about the market and will transit randomly among different state for different bidding records. Therefore, it is very impressive that performances of the adaptive strategy and the Q-learning based adaptive strategy can be as high as about 90% of that of the optimal strategy in different market environments. As the bidder using the adaptive strategy or the Q-learning based adaptive strategy does not need to know the market type in advance, we can draw the conclusion that the bidder using either strategy performs well in different markets, even without any prior knowledge.

To explore the reason that the Q-learning based strategy outperforms the adaptive strategy, we conduct the following experiment. We test the performances of another strategy Q-AS-NoF, which is the same as the Q-learning based adaptive strategy except that we do not call the FillUpQValues function in the algorithm (line 31), and show its performance with the green line in Figure 3. Recall the algorithm of the adaptive strategy, we find that it shares the same principle with Q-AS-NoF but with a few differences, and it is also reasonable that it performs comparably with Q-AS-NoF. Summarizing the above results, we can come to a conclusion that the function FillUpQValues mainly contributes to the improvement of the Q-learning based adaptive strategy on performances to the adaptive strategy, by which the bidder will make correct choices on his action according to the historical information.

We also compare the performances of the adaptive strategy and the Q-learning based adaptive strategy in each run in eight types of market from 0.5:1 to 1.2:1, whose results are shown in Figure 4. For each type of market, the diagonal red line is a line that a point on which means the performances of the adaptive strategy and the Q-learning based adaptive strategy are the same, and the X-coordinate and Y-coordinate are the utilities achieved by the test bidder using the adaptive strategy and the Q-learning based adaptive strategy respectively. From Figure 4, we can see that for each type of market, the points falling into the left part of the line are more then those falling into the right part of the line, which means that in most cases, the performance of the Q-learning based adaptive strategy is bet-

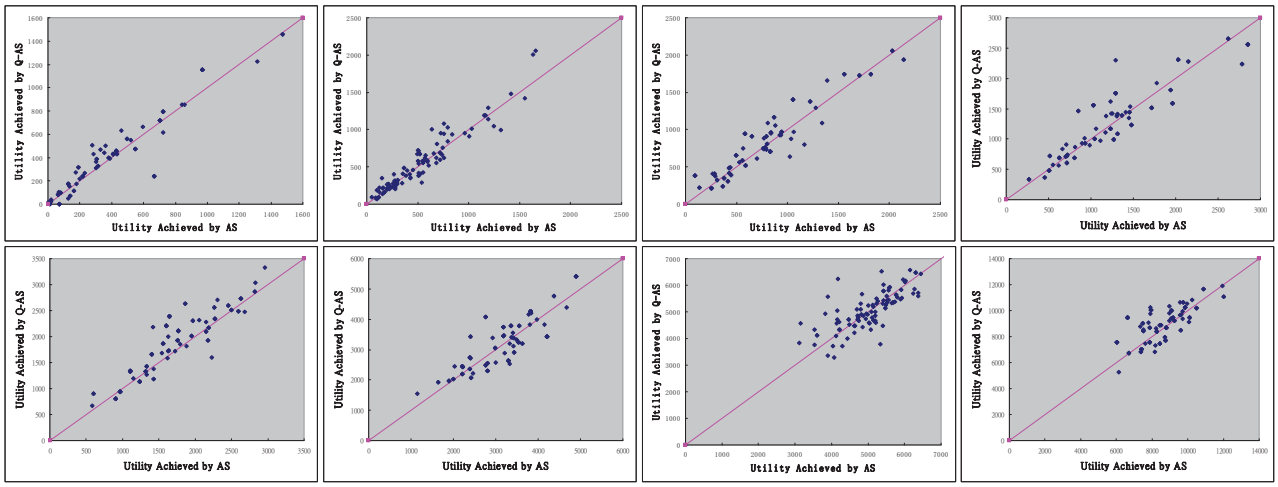

Figure 4: Comparison of Q-AS and AS in a single run in different markets

ter than that of the adaptive strategy. However, there are still cases in which the adaptive strategy outperforms the Q-learning based adaptive strategy, which needs further explorations in future works.

In addition, we also show the typical convergency process of the state of the bidder who uses the Q-learning based adaptive strategy in a single run in four types of market: 0.6:1, 0.8:1, 1.0:1 and 1.2:1. For each type of market, the horizontal line represents the optimal state that the bidder should remain at in that type of market.

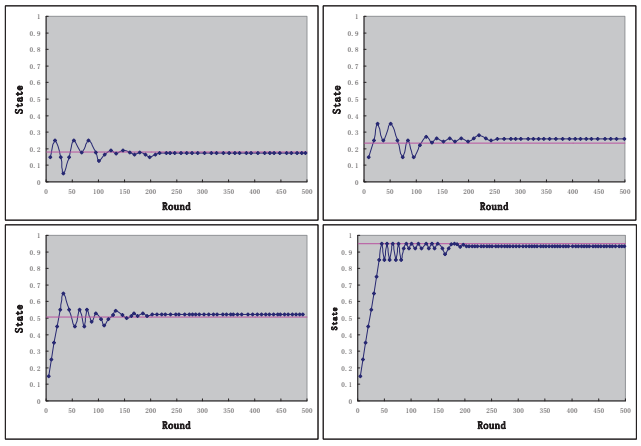

Figure 5: The typical convergency process of the bidder's state in a single run in different markets

From Figure 5, we can see that for each type of market, the transited state of the Q-learning based adaptive strategy has finally converged to the optimal state in that market type, which means the Q-learning based adaptive strategy is capable of learning and adapting in different markets. In addition, the convergence speed is fast: for each market type, the test bidder's state has converged at about the $100th$ round. The bidder keeps remaining at that state in subsequent rounds, which guarantees the Q-learning based adaptive strategy can generate a very good utility compared to the optimal strategy.

6. CONCLUSIONS

In this paper, we propose a Q-Learning based adaptive bidding strategy in combinatorial auctions. The bidder adopting this strategy can transit among different states according to bidding histories and finally converge to the optimal state. Experiment results show that 1) the Q-learning based adaptive strategy performs fairly well compared to the optimal strategy and the outperforms the adaptive strategy and the random strategy in different market environments. 2) the bidder using the Q-learning based adaptive strategy can generate a high utility when compared with that generated by the optimal strategy, even without any prior knowledge about the market. 3) the bidder using the Q-learning based adaptive strategy is capable of learning and adapting, and the convergency speed is fast.

Besides more extensive empirical evaluations, this work can be extended in the following several directions. First, although generally the Q-learning based adaptive strategy outperforms the adaptive strategy, there are still cases in which the adaptive strategy performs better. We are interested in exploring such situations in detail and trying to improve the Q-learning based adaptive strategy. Second, in this work, we use the fixed value for learning rate and discount rate. Next step, we want to make them also adaptive within our algorithm, such that the Q-learning based adaptive strategy can be used in dynamic markets, where resource capacities and the number of bidders vary over time. Another direction is that, as we have mentioned above, the Q-Learning based adaptive bidding strategy is not restrictive to combinatorial auctions. Actually, for any type of auction, maybe even more generally, any type of repeated game, this strategy can help the player to behave strategically and achieve good performance in the game if the reward function is appropriately defined. In the future, we are interested in applying this strategy into other type of auction and analyze its performance. Finally, in this paper, we use the single-agent model, which means that only one bidder uses the Q-Learning based adaptive strategy and the others fix their strategies. An interesting problem is what will happen when more bidders use this strategy simultaneously. Our guess is that when all the bidders adopt this strategy, a Nash Equilibrium will be achieved.

7. REFERENCES

[1] N. An, W. Elmaghraby, and P. Keskinocak. Bidding strategies and their impact on revenues in combinatorial auctions. *Journal of Revenue and Pricing Management*, 3(4):337–357, 2005.

[2] A. Archer, C. Papadimitriou, K. T. Tardos, and J. H. Foundation. An approximate truthful mechanism for combinatorial auctions with single parameter agents. In *In Proc. 14th Symp. on Discrete Alg. ACM/SIAM*, pages 205–214, 2003.

[3] L. Ausubel, P. Cramton, R. McAfee, and J. McMillan. Synergies in Wireless Telephony: Evidence from the Broadband PCS Auctions. *Journal of Economics & Management Strategy*, 6(3):497–527, 1997.

[4] L. M. Ausubel, P. Cramton, and P. Milgrom. The clock-proxy auction: A practical combinatorial auction design. Technical report, 2004.

[5] V. Avasarala, H. Polavarapu, and T. Mullen. An approximate algorithm for resource allocation using combinatorial auctions. In *IAT '06: Proceedings of the IEEE/WIC/ACM international conference on Intelligent Agent Technology*, pages 571–578, Washington, DC, USA, 2006. IEEE Computer Society.

[6] R. BELLMAN. *Dynamic Programming*. Princeton University Press, 1957.

[7] M. Berhault, H. Huang, P. Keskinocak, S. Koenig, W. Elmaghraby, P. Griffin, and A. Kleywegt. Robot exploration with combinatorial auctions. *Intelligent Robots and Systems, 2003. Proceedings. 2003 IEEE/RSJ International Conference on*, 2, 2003.

[8] R. Buyya, J. Giddy, and H. Stockinger. Economic models for resource management and scheduling. pages 1507–1542. Wiley Press, 2002.

[9] S. Clearwater. *Market-Based Control: A Paradigm for Distributed Resource Allocation*. World Scientific, 1996.

[10] P. Cramton, Y. Shoham, and R. Steinberg. *Combinatorial Auctions*. MIT Press, Cambridge, Massachusetts, 2006.

[11] S. de Vries and R. Vohra. Combinatorial Auctions: A Survey. *INFORMS Journal On Computing*, 15(3):284–309, 2003.

[12] I. Foster. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *Euro-Par 2001 Parallel Processing: 7th International Euro-Par Conference, Manchester, UK, August 28-31, 2001: Proceedings*.

[13] Y. Fujishima, K. Leyton-Brown, and Y. Shoham. Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 548–553, 1999.

[14] A. Holland and B. O'Sullivan. Truthful risk-managed combinatorial auctions. In *IJCAI*, pages 1315–1320, 2007.

[15] H. H. Hoos and C. Boutilier. Solving combinatorial auctions using stochastic local search. In *In Proceedings of the Seventeenth National Conference on Artificial Intelligence*, pages 22–29, 2000.

[16] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: a survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

[17] V. Krishna. *Auction Theory*. Academic Press, 2002.

[18] R. Lavi and N. Nisan. Towards a characterization of truthful combinatorial auctions. pages 574–583, 2003.

[19] D. Lehmann, L. I. O'callaghan, and Y. Shoham. Truth revelation in rapid, approximately efficient combinatorial auctions. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 96–102, 1999.

[20] R. Mailler, V. Lesser, and B. Horling. Cooperative negotiation for soft real-time distributed resource allocation. In *AAMAS '03: Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 576–583, New York, NY, USA, 2003. ACM.

[21] N. Nisan and A. Ronen. Computationally feasible vcg mechanisms. In *In ACM Conference on Electronic Commerce*, pages 242–252. ACM Press, 2000.

[22] D. Parkes and L. Ungar. Iterative combinatorial auctions: Theory and practice. *Proceedings of the National Conference on Artificial Intelligence*, pages 74–81, 2000.

[23] S. Rassenti, V. Smith, and R. Bulfin. A combinatorial auction mechanism for airport time slot allocation. *Bell Journal of Economics*, 13(2):402–417, 1982.

[24] B. Rastegari, A. Condon, and K. Leyton-Brown. Stepwise randomized combinatorial auctions achieve revenue monotonicity. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 738–747, 2009.

[25] M. Rothkopf, A. Pekec, and R. Harstad. Computationally Manageable Combinational Auctions. *Management Science*, 44(8):1131–1147, 1998.

[26] T. Sandholm. An implementation of the contract net protocol based on marginal cost calculations. *Proceedings of the National Conference on Artificial Intelligence*, pages 256–262, 1993.

[27] T. Sandholm, S. Suri, A. Gilpin, and D. Levine. CABOB: A Fast Optimal Algorithm for Winner Determination in Combinatorial Auctions. *Management Science*, 51(3):374–390, 2005.

[28] M. Schwind, T. Stockheim, and O. Gujo. Agents' bidding strategies in a combinatorial auction controlled grid environment. In *TADA/AMEC*, pages 149–163, 2006.

[29] X. Sui and H.-F. Leung. An adaptive bidding strategy in multi-round combinatorial auctions for resource allocation. *20th IEEE International Conference on Tools with Artificial Intelligence.*, 2:423–430, 2008.

[30] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[31] C. Watkins. Learning from delayed rewards. In *PhD Thesis University of Cambridge, England*, 1989.

[32] C. J. C. H. Watkins and P. Dayan. Technical note q-learning. *Machine Learning*, 8:279–292, 1992.

[33] M. Yokoo, T. Matsutani, and A. Iwasaki. False-name-proof combinatorial auction protocol: Groves mechanism with submodular approximation. In *AAMAS*, pages 1135–1142, 2006.

[34] E. Zurel and N. Nisan. An efficient approximate allocation algorithm for combinatorial auctions. In *Proceedings of the 3rd ACM conference on Electronic Commerce*, pages 125–136, 2001.