Research Note

Tractable combinatorial auctions and b-matching

Moshe Tennenholtz 1

Faculty of Industrial Engineering and Management, Technion, Israel Institute of Technology, Haifa 32000, Israel

Received 18 January 2001; received in revised form 31 May 2001

Abstract

Auctions are the most widely used strategic game-theoretic mechanisms in the Internet. Auctions have been mostly studied from a game-theoretic and economic perspective, although recent work in AI and OR has been concerned with computational aspects of auctions as well. When faced from a computational perspective, combinatorial auctions are perhaps the most challenging type of auctions. Combinatorial auctions are auctions where agents may submit bids for bundles of goods. Given that finding an optimal allocation of the goods in a combinatorial auction is in general intractable, researchers have been concerned with exposing tractable instances of combinatorial auctions. In this work we expose the use of b-matching techniques in the context of combinatorial auctions, and apply them in a non-trivial manner in order to introduce polynomial solutions for a variety of combinatorial auctions.

© 2002 Elsevier Science B.V. All rights reserved.

Keywords: Combinatorial auctions; b-matching; Electronic commerce

1. Introduction

The emergence of electronic commerce has led to increasing interest in the design of protocols for non-cooperative environments (see, e.g., [4,6,12,16]). The wide-spread of auctions in the Internet, and the fact that auctions are basic building blocks for a variety of economic protocols have attracted many researchers to tackle the challenge of efficient auction design (e.g., [7,8,10,14,17]). The design of auctions introduces deep problems and challenges both from the game-theoretic and from the computational perspectives. This paper mainly concentrates on computational aspects of auctions. More specifically, we
concentrate on addressing computational problems of combinatorial auctions, extending upon previous work on this basic topic [5,9,13,15].

In an auction, a seller sells several goods to several potential buyers. In typical single-object auctions, determining the auction’s winner and its payment is computationally trivial. The problem is still computationally tractable when agents’ valuations for the different objects are additive, i.e., determined in an additive manner by their valuations for the single goods. However, consider a situation where a VCR, a TV, and a Microwave are sold; an agent may be willing to pay $200 for the TV, $300 for the VCR, and $150 for the microwave, but might be willing to pay only $550 for getting all of them. In order to allocate the goods in a satisfactory manner, bids for bundles of goods should be allowed; given these bids, we need to find an optimal, revenue maximizing, allocation of the goods. This problem is referred to as the combinatorial auction problem, and it is in general intractable [13].

One can partition previous work on computational aspects of combinatorial auctions into two parts. One part deals with heuristics for the solution of combinatorial auctions (see, e.g., [5,15]), while the other part deals with the identification of tractable cases of the combinatorial auctions problem (see [9,13]). Our work fits into the latter category. Previous results in that category has introduced a general technique for tackling the complexity of combinatorial auctions. Namely, several authors [3,9] have dealt with the problem of winner determination in combinatorial auctions as an integer programming [IP] problem, and considered linear programming [LP] relaxations of that problem for isolating tractable cases of the general problem. This paper equips researchers who deal with the theory and practice of combinatorial auctions with an additional general technique for addressing the complexity of that problem. Namely, we expose and explore the use of b-matching techniques for the combinatorial auctions setup, and employ b-matching techniques in various ways in order to efficiently address several non-trivial instances of the combinatorial auctions problem. The use of b-matching techniques in the solution of other auctions is discussed in [11]. The b-matching techniques introduced there are however not applicable to combinatorial auctions as discussed in this paper; to the best of our knowledge this is the first paper to consider and apply b-matching techniques to the context of combinatorial auctions.

In Section 2 we present some preliminaries. In particular, we describe the basic b-matching problem, and the combinatorial auctions problem. In Sections 3–7 we expose the use of b-matching techniques for the solution of combinatorial auctions and present polynomial solutions for a variety of combinatorial auctions.

2. Preliminaries

Let $G = (V(G), E(G))$ be a graph, where $V(G)$ is a set of nodes, and $E(G)$ is a set of edges. Each edge $e \in E(G)$ is assigned a (real number) cost $w_e$. Let $b = ((l_1, b_1), (l_2, b_2), \ldots, (l_{|V(G)|}, b_{|V(G)|}))$, where the $b_i$’s are integers and $l_i$ equals $b_i$ or 0 ($1 \leq i \leq |V(G)|$). A $b$-matching is a set $M \subseteq E(G)$ such that, for each node $i \in V(G)$, the number of edges incident with $i$ is no more than $b_i$ and no less than $l_i$. The value of a b-matching is the sum of costs of its edges, i.e., $\sum [w_e \mid e \in M]$. The $b$-matching problem is to find a b-matching of maximum value.
An important result of the field of combinatorial optimization is that the b-matching problem is polynomial [1,2]. This result widely extends upon the more commonly known results about the computation of (standard) matchings, and will play a significant role in this paper.

In a combinatorial auctions setup a seller sells \( m \) goods to \( n \) potential buyers. A bid of agent \( i \) is a pair \((S, p)\), where \( S \) is a bundle of goods and \( p \) is a non-negative real number that denotes the price offer for \( S \). Let \( X = \{x_1, x_2, \ldots, x_t\} \), where \( x_i = (s_i, p_i) \), \(1 \leq i \leq t\), be a set of bids, and denote by \( S(x_i) \) and \( P(x_i) \) the bundle of goods and the price offer of bid \( x_i \), respectively. The combinatorial auction problem \( [\text{CAP}] \) is to find an \( X_o \subseteq X \), for which \( \sum_{x_i \in X_o} P(x_i) \) is maximal, under the constraint that \( S(x_i) \cap S(x_j) = \emptyset \) for every \( x_i, x_j \in X_o, i \neq j \). The CAP is NP-hard [13].

The literature distinguishes between two types of combinatorial auctions. In a sub-additive combinatorial auction an agent’s bid for every bundle \( S = S_1 \cup S_2, S_1 \cap S_2 = \emptyset \) of goods, is less than or equal to the sum of its bids for \( S_1 \) and \( S_2 \). In a super-additive combinatorial auction an agent’s bid for every bundle \( S = S_1 \cup S_2, S_1 \cap S_2 = \emptyset \) of goods, is greater than or equals to the sum of its bids for \( S_1 \) and \( S_2 \). Auctions for substitute goods are sub-additive, while auctions for complementary goods are super-additive.

### 3. Quantity restrictions in multi-object auctions

Consider an auction for the reservation of seats in a particular flight. Each potential buyer submits bids for each of the seats in the airplane, but restricts the total number of seats he may wish to obtain. This auction has the property that the payment of agent \( i \) for the set of seats allocated to it, subject to his quantity constraint, is the sum of his bids for the individual seats in this set. However, this auction is a sub-additive combinatorial auction; a buyer will pay 0 for every additional seat assigned to him beyond his limit on the number of required seats.

**Definition 1.** A quantity-constrained multi-object auction is a sub-additive combinatorial auction where bids are of the form \((a_1, p_1, a_2, p_2, \ldots, a_k, p_k, q)\) where \( p_i \) is a price offer for object \( a_i \), and \( q \) is the maximal number of objects that are to be assigned.

Notice that in the above definition we used the term *multi-object auction*. This is in order to emphasize that although the auction is combinatorial, it has some syntactic similarity with other types of multi-object auctions, such as constrained multi-object auctions [11], since the bids are not stated explicitly for bundles of goods.

**Theorem 1.** Quantity-constrained multi-object auctions are computationally tractable.

**Proof.** We reduce the input of a quantity-constrained multi-object auction to an input of a b-matching problem in a bipartite graph \( G = (V_1 \cup V_2, \ E = V_1 \times V_2) \) where \( V_1 \) is isomorphic to the set of bids and \( V_2 \) is isomorphic to the set of objects. An edge \( e_{ij} \) which connects a node associated with the \( i \)th bid to a node associated with the \( j \)th object will be assigned a cost that equals bid \( i \)’s offer for good \( j \) (the cost equals 0 if bid \( i \)
does not refer to object \( j \). The pair of \( b \) values of a node \( v_i \in V_1 \) associated with a bid \((a_1, p_1, a_2, p_2, \ldots, a_k, p_k, q)\) will be \((0, q)\). The \( b \) value of \( v_j \in V_2 \) will be \((0, 1)\).

The above reduction is polynomial and creates an input of a \( b \)-matching problem. There is a one to one and onto mapping between the \( b \)-matchings and the possible allocation of the goods based on the agents’ bids. A \( b \)-matching where \( v_i \in V_1 \) is connected to \( v_1^1, v_2^2, \ldots, v_t^2 \in V_2 \) (\( 1 \leq t \leq q \)) will be associated with the allocation of the objects associated with nodes \( v_1^1, v_2^2, \ldots, v_t^2 \in V_2 \) to the agent who submitted the bid \( b_i \). Similarly, given a feasible allocation of the goods, we can look at the edges associated with that allocation to define a corresponding \( b \)-matching. Moreover, the sum of costs associated with the given \( b \)-matching is by definition the sum of price offers for the objects as specified by the corresponding bids. Hence, the optimal weighted \( b \)-matching of the graph defines a solution to the quantity-constrained multi-object auction.

The importance of the above result is by showing a general connection between \( b \)-matchings and feasible allocations of the goods based on agents’ bids. In the sequel we will consider also cases where this mapping is more subtle. For ease of exposition, although some of the following results expand on the construction presented in the proof of Theorem 1, we will introduce each of these results separately, emphasizing the distinguished points of its proof.

4. Binary bids revisited

Theorem 1 shows that quantity constraints can be incorporated into simple multi-object auctions, while still getting tractable solutions. Previous work has tried to tackle the tractability of combinatorial auctions where bids are given for non-singleton bundles. It was shown that the case of bundles of size two is tractable, while the case of larger bundles is NP-hard. We now show that the case of bundles of size two and the case of quantity constraints can be tackled simultaneously in an efficient manner.

**Definition 2.** A quantity-constrained multi-object action with binary combinatorial bundles is a sub-additive combinatorial auction that allows two types of bids: (1) The bids allowed in a quantity-constrained multi-object auctions. (2) Bids of the form \((a, p, b, q, l)\) where \( p \) is the price offer for good \( a \), \( q \) is the price offer for good \( b \), and \( p + q - l \) is the combinatorial price offer for the pair \([a, b]\), where \( 0 < l < \min(p, q) \).

**Theorem 2.** Quantity-constrained multi-object auctions with binary combinatorial bundles are computationally tractable.

**Proof.** We construct a graph \( G \) as in Theorem 1, and for each bid of the form \( x = (a, p, b, q, l) \) we do the following:

1. We add three nodes \( v_x, v_{x1}, v_{x2} \).
2. We connect \( v_x \) to \( a, b, v_{x1} \) and \( v_{x2} \).
(3) Denote $w = l/2$. We assign edge weights as follows: $(v_x, a) \rightarrow p - w$, $(v_x, b) \rightarrow q - w$, $(v_x, v_{x1}) \rightarrow w$, $(v_x, v_{x2}) \rightarrow -w$.

(4) We require that $v_x$ will have the b-value $(2,2)$ (i.e., exactly 2), and that $v_{x1}$ and $v_{x2}$ will have the b-value $(0,1)$ (i.e., at most 1).

We now prove that the optimal b-matching in the graph $G$ corresponds to an optimal solution of the multi-object auction with binary combinatorial bundles.

Consider the node $v_x$ associated with the bid $x = (a, p, b, q, l)$. In a b-matching this node should be associated with two other nodes. If both $(v_x, v_{x1})$ and $(v_x, v_{x2})$ are in the b-matching then this situation will be associated with the situation where bid $x$ has not been selected, and both $a$ and $b$ have not been allocated based on that bid. If $(v_x, v_{x1})$ and $(v_x, a)$ are in the b-matching then this situation will be associated with the situation where only object $a$ has been assigned based on the bid $x$; similarly, for the case where $(v_x, v_{x1})$ and $(v_x, b)$ are in the b-matching. If both $(v_x, a)$ and $(v_x, b)$ are in the b-matching then this situation will be associated with the situation where the pair $(a, b)$ is allocated based on the bid $x$. Notice that an optimal b-matching cannot contain $v_{x2}$ but not $v_{x1}$, since $l > 0$. We need now to check that the value obtained by having $v_x$ connected both to $a$ and to $b$ in an optimal b-matching will be $p + q - l$, that the value obtained by having $v_x$ connected only to $a$ (respectively $b$) will be $p$ (respectively $q$), and that the value obtained by having $v_x$ disconnected from $a$ and $b$ in the b-matching (i.e., connected to both $v_{x1}$ and $v_{x2}$) will be 0.

We consider the possible allocations with regard to the bid $x = (a, p, b, q, l)$:

1. If both $a$ and $b$ are not allocated then the cost contributed by this bid in the corresponding b-matching is $w - w = 0$ as required.
2. If only $a$ is allocated then the cost contributed by this bid in the corresponding b-matching is $p - w + w = p$ as required.
3. If only $b$ is allocated then the cost contributed by this bid in the corresponding b-matching is $q - w + w = q$ as required.
4. If both $a$ and $b$ are allocated then the cost contributed by this bid in the corresponding b-matching is $q - w + p - w = p + q - l$ as required.

Notice that in order for the above to hold we must have that $p - w > w$ and $q - w > w$. since otherwise $(v_x, v_{x1})$ might be selected by the optimal b-matching instead of either $(v_x, a)$ or $(v_x, b)$ when the optimal assignment of the goods is to have both $a$ and $b$ based on bid $x$. This is guaranteed by the condition $l < \min(p, q)$. \(\square\)

4.1. Intuitive discussion of our proof technique

The reader may find the following discussion of our proof technique useful. One way to represent a simple bidding activity, is by using a bi-partite graph. One part of the nodes in this graph is associated with the bids’ names, and the other part of the nodes is associated with the objects’ names. If the bids were only for single objects, or if bids were fully additive (e.g., it cannot be the case that an agent’s bid for a pair of goods $(g_1, g_2)$ is different from the sum of its bids for $g_1$ and $g_2$) then we can associate edges in the graph with bids.
for the goods, where the weight of the edge connecting a bid $b$ to an object (good) $g$ is the value of the corresponding bid. Finding (standard) optimal weighted matching will be satisfactory in this case. However, in combinatorial auctions the bids are not fully additive. Our way to overcome this is by adding few additional nodes to be connected to the node denoting a particular bid (here a bid corresponds to a price offer for a bundle, e.g., a pair of goods, and price offers for all its subsets). The value of the edges are selected in a way that they will compensate for the deviation from the fully additive case. If the agent submits a bid for a pair of goods, and the singletons it consists of, then two additional nodes will be added, and the $b$-value of the original bid’s node will be taken to be 2. Notice now that when selecting two nodes which are adjacent to the original bid’s node, a (possibly empty) subset of the objects associated with the bid will be selected. The weights of the corresponding edges are calculated in a way that when two adjacent nodes to the original bid’s node are selected (as required by the $b$-matching), then the sum of the corresponding edge values (i.e., the values of the edges connecting the selected nodes to the original bid’s node) will be identical to the value of the bid on the corresponding objects (which may be a pair of goods, only a single good, or none of the goods).

As the reader may notice, the above idea is quite general and can be applied to relatively complicated non-additive cases. In the next sections we will show how it applies to other non-additive (combinatorial) auctions.

4.2. Remark: Multi-unit binary combinatorial auctions

An interesting generalization of combinatorial auctions with binary bids has to do with the case where there are several available units of each of the goods. In this case, when the agent makes a bid for the bundle $\{a, b\}$, it does not care which copies of the objects $a$ and $b$ it will obtain, as long as it will obtain a copy of each one of them. The related problem is termed the multi-unit CAP (muCAP). Using the idea of $b$-matching it is easy to show that multi-unit binary combinatorial auctions are tractable. All that is needed is to treat the goods as nodes in a graph, bids as corresponding edges and edge weights, and have the $b$-value of each node to be the number of units available from the corresponding good. The fact the corresponding muCAP is tractable easily follows.

5. Beyond binary bids

As we mentioned, combinatorial auctions where bids are only for single goods or for pairs of goods are tractable [13]. However, when bids are for bundles of size greater than two, the CAP is in general intractable. Notice that in Theorem 1 we presented general tractable non-binary combinatorial auctions where the bids for singletons can not be simply sum up in order to get the bid for a bundle of goods. However, this was a result of a constraint on the number of goods to be allocated; when this constraint is satisfied, the bid/payment for an allocated set of goods equals the sum of bids for the objects it consists of. In this section we wish to relax this property; namely, we wish to consider cases where the bid for an allocated set of goods is different from the sum of bids for the singletons it consists of. Our aim is to do so for auctions where bids are for bundles of size greater
than two. We now present two results that we believe to be of considerable importance in this regard. The first result shows that we can handle bundles of size larger than two in which the bid for an allocated set of goods is different from the sum of bids for the singletons it consists of. Then we show that a general form of combinatorial auctions where bids for triplets are permitted, i.e., combinatorial auctions with symmetric bids for triplets, is tractable. These results, and the exact assumption required for establishing them, are presented in the following subsections. The intuition of the proof techniques is similar to the one presented in the previous section, although the technical details in some cases are slightly more involved.

5.1. Almost additive auctions

Definition 3. An almost-additive multi-object auction is a combinatorial sub-additive auction where bids for non-singletons are of the form \((a_1, p_1, a_2, p_2, \ldots, a_k, p_k, q)\) where \(p_i\) is the price offer for object \(a_i\), the price offer for any \(A \subset \{a_1, \ldots, a_k\}\) equals \(\sum_{a_i \in A} p_i\), and the offer for \(\{a_1, \ldots, a_k\}\) is \(q\); in addition, \(w = \sum_{1 \leq i \leq k} p_i - q > 0\), and \(w < p_j\) (1 \(\leq j \leq k\)).

In an almost-additive multi-object auction a shopping list of items is gradually built until we reach a situation that the valuations become sub-additive; sub-additivity is a result of the requirement that \(w > 0\). The other condition on \(w\) implies that the bid on the whole bundle is not too low with respect to the sum of bids on the single goods it consists of. Notice that typically \(q\) will be greater than the sum of any proper subset of the \(p_i\)'s; our only requirement is that \(q\) will be lower than the sum of all the \(p_i\)'s; hence, bidding on and allocation of the whole \(\{a_1, a_2, \ldots, a_k\}\) bundle is a feasible and reasonable option.

Theorem 3. Almost-additive multi-object auctions are computationally tractable.

Proof. We start from the graph \(G\) that was built for the quantity-constrained multi-object auction, and add the following for each almost additive bid \((a_1, p_1, a_2, p_2, \ldots, a_k, p_k, q)\):

(1) Construct \(k + 1\) new nodes: \(v, v_1, v_2, \ldots, v_k\), and connect \(v\) to each of the \(v_j\)'s. In addition, we connect \(v\) to each object \(a_j\) (1 \(\leq j \leq k\)).
(2) Let \(w = (\sum_{1 \leq i \leq k} p_i) - q\), and let \(a = \frac{k - 1}{k}w\).
(3) The weights of the newly added edges will be as follows:
   (a) \((v, v_1) \rightarrow a\),
   (b) \((v, v_j) \rightarrow \frac{a}{k - 1}\) (1 \(\leq j \leq k\)),
   (c) \((v, a_j) \rightarrow p_j - \frac{a}{k - 1}\), for (1 \(\leq j \leq k\)).
(4) The b-value of \(v\) is taken to be exactly \(k\) (i.e., \((k, k)\)). The b-values of the \(v_j\)'s are taken as \((0, 1)\) (i.e., at most 1).

Consider now the optimal weighted b-matching in the corresponding graph. It follows that if none of the \(a_j\)'s are allocated then the sum of costs contributed to the corresponding matching is 0, as required.
If a strict non-empty subset $a_{i_1}, \ldots, a_{i_s}$ ($1 \leq s < k$) of the $a_j$’s is allocated then the sum of costs contributed to the corresponding matching is
\[
\left( \sum_{1 \leq i \leq s} p_i \right) - s \frac{a}{k-1} + a - \frac{a}{k-1} (k - s - 1) = \sum_{1 \leq i \leq s} p_i,
\]
as required.

If all the $a_j$ are allocated then the sum of costs contributed to the corresponding matching is
\[
\sum_{1 \leq j \leq k} p_k - k \frac{a}{k-1} = \sum_{1 \leq j \leq k} p_k - w = q,
\]
as required.

Notice that we need to have $a < p_j - \frac{a}{k-1}$ for every $j$ in order to obtain the desired reduction. If this is not satisfied then the $b$-matching algorithm may select $(v, v_1)$ instead of one of the $(v, a_j)$’s, and the value obtained will be different from the price offer for the bundle. Hence, we need to have $\frac{a}{k-1} a < p_j$, i.e., we need to have that $w < p_j$, which is exactly our requirement. As a result, by finding an optimal weighted $b$-matching in the corresponding graph, we get a solution to the corresponding CAP. \( \square \)

### 5.2. The case of triples

The case of combinatorial auctions with bids for triples of goods, rather than only for pairs of goods, is NP-hard. However, consider the following:

**Definition 4.** A combinatorial auction with sub-additive symmetric bids for triplets is a sub-additive combinatorial auction where bids are either for singletons, for pairs of goods (and the singletons they are built of), or for triplets of goods (and the corresponding subsets). Bids for pairs of goods are as in Definition 2, while bids for triplets have the form $(a_{i_1}, p_{i_1}, a_{i_2}, p_{i_2}, a_{i_3}, p_{i_3}, b_1, b_2)$: $p_i$ is the price offer for good $a_i$, the price offer for any pair of goods $\{a_i, a_j\}$ ($1 \leq i, j \leq 3; i \neq j$) is $p_i + p_j - b_1$, and the price offer for the whole triplet $\{a_1, a_2, a_3\}$ is $p_1 + p_2 + p_3 - b_2$.

Symmetric bids may be applicable to many domains. One motivation is the case where each agent has certain fixed cost associated with any purchase (e.g., paper work expenses, etc.), which is independent of the actual product purchased; this additional cost per product will decrease as a function of the number of products purchased (e.g., one does not need to duplicate the amount of paper work done when purchasing a pair of products rather than only one).

**Theorem 4.** Combinatorial auctions with sub-additive symmetric bids for triplets, where each bid for triplet $(a_{i_1}, p_{i_1}, a_{i_2}, p_{i_2}, a_{i_3}, p_{i_3}, b_1, b_2)$ has the property that $b_2 > 3b_1$, and $p_i > b_2 - b_1$ ($1 \leq i \leq 3$), are tractable.

The theorem makes use of two conditions that connect $b_1$, $b_2$, and the bids on singletons. These conditions measure the amount of sub-additivity relative to the purely additive case.
where a bid for a bundle is the sum of bids for the singletons it consists of. The first condition is that the decrease in valuation/bid for a bundle, relative to the sum of bids for the singletons it consists of, will be proportional to the bundle’s size; the second condition connects that decrease to the bids on the singletons, and requires that the above-mentioned decrease will be relatively low compared to the bids on the single goods. Both of these conditions seem quite plausible for many sub-additive auctions.

**Proof.** We will use the graph $G$ constructed in Theorem 2 for quantity-constrained multi-object auctions with binary combinatorial bundles, and for any bid on a triplet, $(a_1, p_1, a_2, p_2, a_3, p_3, b_1, b_2)$, we will add the following:

1. Construct 4 new nodes: $v_0, v_1, v_2, v_3$, and connect $v_0$ to $v_1, v_2$ and $v_3$.
2. Let $k = \frac{b_2 - 3b_1}{2b_2 - 3b_1}$, and let $a = \frac{b_1}{2b_2 - 3b_1}$.
3. Assign weights to the new edges as follows:
   - (a) $(v_0, a_j) \rightarrow p_j - a + ka$ for $1 \leq j \leq 3$.
   - (b) $(v_0, v_1) \rightarrow a$.
   - (c) $(v_0, v_2) \rightarrow -ka$.
   - (d) $(v_0, v_3) \rightarrow -(1 - k)a$.
4. Take the b-value of $v_0$ to be exactly 3, and of $v_1, v_2, v_3$ to be at most 1 (i.e., $(0, 1)$).

We now compute an optimal weighted b-matching on the generated graph, and claim it defines an optimal allocation of the goods. The proof makes use of the following observations:

1. If none of the $a_j$ in the triplet are allocated then the cost contributed to the corresponding matching is $a - ka - (1 - k)a = 0$, as required.
2. If only one item $a_i$ is allocated, then the cost contributed to the corresponding matching is $p_i - a + ka + a - ka = p_i$, as required.
3. If $a_i$ and $a_j$, $i \neq j$, are allocated, then the cost contributed to the corresponding matching is $p_i + p_j - a + 2ka = p_i + p_j - b_1$, as required.
4. If the whole triplet is allocated then the cost contributed to the corresponding matching is $p_1 + p_2 + p_3 - 3a + 3ka = p_1 + p_2 + p_3 - 3a(1 - k)$, which can be shown to be equal to $p_1 + p_2 + p_3 - b_2$, as required.

Notice that we must require that $p_j - a + ka > a$ for every $j$; otherwise, the optimal b-matching will select $(v_0, v_1)$ instead of one of the edges connecting $v_0$ to the nodes associated with the goods, when the optimal allocation selects the bid that corresponds to the whole triplet. Hence, we need to require that $p_j > 2a - ka = (2 - k)a$. But,

$$2 - k = 2 - \frac{b_2 - 3b_1}{2b_2 - 3b_1} = \frac{3b_2 - 3b_1}{2b_2 - 3b_1}.$$  

On the other hand,

$$1 - 2k = 1 - \frac{2b_2 - 6b_1}{2b_2 - 3b_1} = \frac{3b_1}{2b_2 - 3b_1}.$$
which implies that \( \frac{1}{1-k} = \frac{2b_2-3b_1}{3b_1} \). Therefore
\[
(2-k)a = \frac{3b_2 - 3b_1}{2b_2 - 3b_1} \cdot \frac{2b_2 - 3b_1}{3b_1} = b_2 - b_1.
\]
Since we have required \( p_j > b_2 - b_1 \) we get the desired result. \( \square \)

6. A tractable super-additive combinatorial auction

In the previous sections we have presented solutions for some non-trivial sub-additive combinatorial auctions. In this section we show an instance of super-additive combinatorial auctions that can be solved by similar techniques.

**Definition 5.** A combinatorial auction with super-additive symmetric bids for triplets is a super-additive combinatorial auction where each agent submits a bid for a triplet of goods and its corresponding subsets, and is guaranteed to obtain at least one good. The bids for triplets have the form \((a_1, p_1, a_2, p_2, a_3, p_3, b_1, b_2)\): \(p_i\) is the price offer for good \(a_i\), the price offer for any pair of goods \(\{a_i, a_j\} (1 \leq i, j \leq 3; i \neq j)\) is \(p_i + p_j + b_1\), and the price offer for the whole triplet \(\{a_1, a_2, a_3\}\) is \(p_1 + p_2 + p_3 + b_2\). We assume \(1.5b_1 \leq b_2 \leq 2b_1\).

Notice that the major additional restriction we have here is that the auction procedure must allocate at least a single good to each agent. This restriction is in the spirit of constrained multi-object auctions (see [11] for results and discussion of these basic auctions): many auctions’ settings are not concerned only with revenue maximization, but should obey additional constraints (this is for example the case in the famous FCC auctions), and constraints on the allocations are considered. One of the most simple such constraints is restricting the allocation of goods to an agent to include at most/at least a particular number of particular elements. For example, when allocating communication lines a user may be guaranteed to obtain one of the several communications lines that are needed for its operation. In such cases, a reserve price restricting the the value of bids may be useful (this modification will not change our result).

**Theorem 5.** Combinatorial auctions with super-additive symmetric bids for triplets are computationally tractable.

**Proof.** We build a bipartite graph \(G = (V_1 \cup V_2, E)\), where \(V_1\) is isomorphic to the set of agents and \(V_2\) is isomorphic to the set of goods. The set of edges \(E\) will be defined as follows. For each \(v_i \in V_i\), associated with the bid \((a_1, p_1, a_2, p_2, a_3, p_3, b_1, b_2)\) of agent \(i\), we add the following:

1. We add two additional nodes, \(v_{i1}\) and \(v_{i2}\).
2. Let \(a = \frac{3}{2}b_2 - b_1\), and let \(b = b_1 - \frac{1}{2}b_2\). Notice that both \(a\) and \(b\) are positive, since \(1.5b_1 \leq b_2 \leq 2b_1\). We build the following edges and corresponding weights:
   a. \((v_i, a_1) \rightarrow p_1 + a + b\).
   b. \((v_i, a_2) \rightarrow p_2 + a + b\).
(c) \((v_1, a_3) \rightarrow p_3 + a + b\).
(d) \((v_i, v_j) \rightarrow -a\).
(e) \((v_1, v_2) \rightarrow -b\).

(3) The b-value of \(v_i\) is exactly 3, the v-values of the \(a_j\)'s and the \(v_{0j}\)'s are at most 1 (i.e., \((0, 1))\).

We now compute an optimal weighted b-matching, which gives us the desired result. This stems from the following observations on the allocations determined by the b-matching with regard to the bid \((a_1, p_1, a_2, p_2, a_3, p_3, b_1, b_2)\) by an agent \(j\):

(1) If only item \(a_i\) is allocated then the corresponding weight in the b-matching will be \(p_i + a + b - a - b = p_i\), as required.
(2) If two items, \(a_i\) and \(a_j\), are allocated, then the corresponding weight in the b-matching will be \(p_i + p_j + 2a + 2b - a = p_i + p_j + a + 2b = p_i + p_j + b_1\), as required.
(3) If all three items are allocated then the corresponding weight in the b-matching will be \(p_1 + p_2 + p_3 + 3a + 3b = p_1 + p_2 + p_3 + b_2\), as required.

Notice that in order that the above will hold we need to require that \(-a \geq -b\), or \(a \leq b\), which is satisfied since \(b_1 \geq b_2/2\). \(\Box\)

7. Combinatorial network auctions

Auctions for linear goods are a useful case of tractable combinatorial auctions (see [9, 13]). In an auction for linear goods we have an ordered list of \(m\) goods, \(g_1, \ldots, g_m\), and bids should refer to bundles of the form \(g_i, g_{i+1}, g_{i+2}, \ldots, g_{j-1}, g_j\) where \(j \geq i\), i.e., there are no “holes” in the bundle. Auctions for linear goods can be used for time scheduling (e.g., for the allocation of time slots in a conference room), or for the allocation of one-dimensional space (e.g., for parts of a seashore), etc. In this section we widely extend the result on the tractability of auctions for linear goods, by considering combinatorial network auctions:

**Definition 6.** Let \(O = \{g_1, \ldots, g_m\}\) be a set of goods. A network of goods is a tree \(G(O) = (V(O), E(O))\), where the set of nodes, \(V(O)\), is isomorphic to the set of goods \(O\). A **combinatorial network auction** with respect to the set of goods \(O\) and the network \(G(O)\), is a combinatorial auction where bids can be submitted only for bundles associated with paths in \(G(O)\).

It is clear that combinatorial auctions for linear goods are simple instances of combinatorial network auctions, where the network is a simple path. Using, yet again, matching techniques we can now show:

**Theorem 6.** Combinatorial network auctions are computationally tractable.
Proof. Consider the graph $G(O)$. We construct the (weighted) graph $G_{\text{net}}$ which is built from $G(O)$ by adding the following nodes, edges, and edge weights:

1. Each edge of $G(O)$ will be assigned the weight 0.
2. For each bid $b$ that refers to a path from $v_1$ to $v_2$, we add a new node $v_b$, and two edges: one that connects $v_b$ to $v_1$ and has the weight 0, and one that connects $v_2$ to $v_b$ and has a weight that equals the price offer in $b$.
3. For each $v \in V(O)$ we add a simple loop that connects $v$ directly to itself, and has the weight 0. Similarly, for every node $v_b$, associated with bid $b$, we add an edge with weight 0 connecting it to itself.

The optimal allocation is given by computing an optimal weighted coverage of $G_{\text{net}}$ by circles. In other words, we look for a subset of the edges, such that in the induced sub-graph with the same nodes and the corresponding edges, both the in-degree and the out-degree of each node is exactly 1; we search for such a set of maximal weight.

There is an isomorphism between the coverings of $G_{\text{net}}$ by circles and the feasible selection of bids to determine the allocation of goods. If we consider a coverage of the graph by circles, and remove from them the ones added in step 3, we get a well defined set of bids that define a feasible allocation of the goods; similarly, if we consider a subset of the bids that define a feasible allocation then it defines a proper coverage of the graph by circles (after adding the additional self-loops for goods that have not been assigned and bids that have not been selected). Hence, we get that the CAP problem in this case is isomorphic to the problem of optimal coverage by circles.

The problem of optimal coverage by circles in a directed graph, with self-loops on any node is known to be solved by the computation of optimal (weighted) perfect matching in an undirected bi-partite graph.\footnote{Although this reduction does not appear as a theorem in classical combinatorial optimization texts, it is a well-known exercise. This is obtained by splitting any node $v_i$ into two node $v_i^f$ and $v_i^j$, and connecting $v_i^f$ to $v_i^j$ if and only if there is an edge leading from $v_i$ to $v_j$.}

Given that the problem of finding an optimal (weighted) perfect matching is a bi-partite graph is polynomial we get the desired result. \hfill \Box

The reader may wish to notice that the proof of the above result did not use the whole power of b-matching. Rather, we have reduced the problem to a standard matching technique. This is different from some of our previous results where the whole power of b-matching techniques has been applied.

8. Conclusion

The study of auctions has received much attention in the recent AI literature. Auctions are a basic building block of protocols for non-cooperative environments, and as a result are central to the theory of mechanism design in economics. Most of the related literature is concerned with game-theoretic aspects of auctions. However, when facing combinatorial
auctions, deep computational problems arise. Combinatorial auctions are essential when dealing with substitute or complementary goods. Several interesting heuristics for dealing with the combinatorial auctions problem have been presented, but the literature on analytic approaches to dealing with this problem is very limited. In particular, the only general combinatorial optimization technique suggested so far for dealing with this problem was the straightforward one: since the CAP is an integer programming problem, one can look at the LP relaxation of the problem, and study the cases where this relaxation yields integer solutions. In this paper we discuss another powerful combinatorial optimization technique for dealing with combinatorial auctions: b-matchings. The use of b-matching in this context is not immediate, and requires non-trivial reductions. In this paper we have introduced several such reductions, dealing with non-trivial instances of the combinatorial auctions problem. We believe that the b-matching techniques exposed in this paper, can be further used and applied to various types of combinatorial auctions, and can play an important role in addressing the complexity of these basic mechanisms.

References