Truthful Mechanism Design without Money: No Upward Bidding

Salman Fadaei and Martin Bichler
Department of Informatics, TU München, Munich, Germany
salman.fadaei@in.tum.de, bichler@in.tum.de

Abstract. Gibbard-Satterthwaite (G-S) theorem rules out the existence of any truthful, non-dictatorial and unanimous social choice function whose range comprises three or more alternatives. To circumvent the G-S impossibility, researchers have introduced restricted domains with additional assumptions to admit truthful mechanisms. We follow this line of research and look at a setting in which a set of items has to be assigned to a set of agents and transfer of money is not allowed. Agents have private cardinal valuations over packages of items. Agents are selfish and may report wrong valuations in order to maximize their utility which is to achieve a package of higher valuation. Our goal is to find a non-dictatorial and truthful mechanism optimizing social welfare. We eliminate the possibility of upward bidding. The assumption of no upward bidding may seem strong, nevertheless we argue that finding a truthful mechanism for this setting falls under the G-S impossibility. Consequently, we analyze the problem with additional assumptions such as small markets with identical items, single-parameter valuations and variations of the generalized assignment problem and show truthful and non-dictatorial mechanisms for these settings.

Keywords: Mechanism Design without Money, Generalized Assignment Problem, Truthfulness, Welfare Approximation

1 Introduction

Truthful mechanism design without money under general preferences is a classical topic in social choice theory. Truthfulness ensures that no agent can be better off by manipulating her true preferences. The Gibbard-Satterthwaite theorem proves that the class of truthful mechanisms is limited to dictatorships [1, 2]. In particular, it states that any truthful social choice function which selects an outcome among three or more alternatives has to be trivially aligned with the preference of a single agent. There have been a number of extensions analyzing more specific domains without money, typically resulting in impossibility results. See, for example [3–5].

Despite these impossibility results, there are some class of environments where truthful mechanisms exist. While some of these classes involve the transfer of money, others do not and our work belongs to the latter.
In search for reasonable mechanisms without money, one has to look at restricted domains of preferences in order to get around the impossibility result. For example, when agents valuations are restricted to single-peaked preferences over a one-dimensional public space, simply returning the median of the peaks determines a truthful social choice [6]. Single-peaked preferences have a single most-preferred point in an interval, and are decreasing as one moves away from that peak. The domain is simple yet useful to model political policies, economic decisions, and other useful applications.

Another area for truthfulness without money is matching markets. The stable matching problem is introduced to model the assignment of students to colleges. In the simplest version of two-sided matching, a set of men has a strict preference ordering over a set of women and vice versa. A matching is an assignment of men to women where each side is assigned to only one element of the other side. The deferred acceptance algorithm is well known to find a stable matching which is truthful for the proposing side and not necessarily truthful for the other side [7].

Procaccia and Tennenholtz introduced the technique of welfare approximation as a means to drive truthful approximation mechanisms without money [8]. This type of approximation is not meant to handle computational intractability but a method to achieve truthfulness. Follow up to this work there has been several works which employ this technique. Dughmi and Ghosh [9] and Chen et al. [10] for instance analyzed truthful mechanisms without money for a strategic variant of the generalized assignment problem and its specialized instances. In this domain, jobs are held by strategic agents and the set of compatible machines is the private knowledge of each agent however the values of job-machine assignments are assumed to be public or verifiable.

Budish proposes a multi-unit assignment mechanism to the course-allocation problem [12]. His mechanism adapts the idea of Competitive Equilibrium from Equal Incomes to the case of indivisible goods. The mechanism accommodates arbitrary ordinal preferences over schedules, and is approximately ex-post efficient. His mechanism is strategyproof in an economy with a large number of students.

Very recently, Nguyen et al., analyze a problem of allocating bundles of indivisible objects without money [13]. The agents have multi-unit demand preferences and the maximum size of a bundle assigned to each agent is known and fixed. The authors introduce mechanisms which provide strong efficiency and envy-freeness properties and are asymptotically strategy-proof. While the setting and the techniques therein are similar to ours, the main difference is that in that work the authors are not directly aiming for truthfulness rather they compute an envy-free solution.

In this paper, we study a setting where transfer of money is not allowed and agents have private cardinal valuations on packages of items. We assume agents may strategize by downward bidding and do not report values higher than their true value for any package, i.e., upward bidding is eliminated. We seek for welfare maximizing algorithms while providing agents with incentives to report their true valuations.
1.1 Our Results

We study truthful welfare maximizing mechanisms using welfare approximation technique. The need for approximating social welfare arises for two reasons. One is because the underlying optimization problem is computationally intractable. In the literature on quasi-linear mechanism design, there exists a long list of approximate mechanisms which stems from this ground [16]. In contrast, we approximate the welfare to obtain truthfulness. The approach is a follow-up to that of Procaccia and Tennenholtz [8]: we maximize welfare without considering incentives, and refer to this as optimal value. We will then say that a strategy-proof mechanism returns (at least) a ratio $\alpha$ of the optimal if it’s value is always greater than or equal to $\alpha$ times the optimal value.

First, we argue that the assumption of no upward bidding is not an escape route from the G-S impossibility result. Next, we show that there is a randomized mechanism with an approximation ratio of $O(\sqrt{m})$ for the problem.

We look at small markets with two bidders and two items only and find out there is a truthful deterministic mechanism with a golden approximation factor of $\frac{\sqrt{5}-1}{2}$. A randomized mechanism for the environment with two bidders and two units achieves a factor of 3/4. Unfortunately, the deterministic mechanism cannot be extended to larger markets, as we will show.

Finally, we analyze GAP-MS, a strategic variant of the generalized assignment problem in which the machines are held by strategic agents. We show that there exists a truthful 1/4 approximate algorithm if each job has the same size and value over machines. In addition, when each job has the same value density over the machines, there exists a truthful 1/4 approximate algorithm. Moreover, we show that there is a $\Omega(1/\ln(U/L))$ approximate mechanism for GAP-MS where $U$ and $L$ are the upper and lower bound for value densities of the jobs.

1.2 Paper structure

In Section 2 we introduce necessary notation and definitions used throughout the paper. In Section 3 we introduce a randomized approximation mechanism for general valuations that is universally truthful. Section 4 focuses on approximation mechanisms in markets with two units of a good and two bidders only. In Section 5, we analyze GAP-MS and some variants of it. Finally, we conclude with a summary and a discussion about future research questions.

2 Preliminaries

In a combinatorial market a set of $m$ items $J$ should be assigned to as set of $n$ agents $I$. Each agent $i$ has a private valuation $v_i(S)$ for every subset $S$ of items. Valuations are normalized and monotone. The welfare of an assignment $S_1, \ldots, S_n$ of items to the agents is $\sum_{i=1}^{n} v_i(S)$. Our objective is to assign disjoint packages $S_1, \ldots, S_n$ to agents in order to maximize social welfare.

Agents are selfish and every agent aims to maximize her utility by obtaining the package with the highest value. The utility of each agent $i$ for receiving
package $S$ is $v_i(S)$. An agent may misreport her valuation by bidding $v'_i$ rather than true valuation $v_i$. We eliminate the possibility of upward bidding. The assumption of no upward bidding translates to the following in our model. No agent $i$ reports $v'_i(S) > v_i(S)$ for some package $S$.

In a truthful mechanism, regardless of submitted reports by other agents, no individual agent $i$ will be better off by reporting $v'_i(S) < v_i(S)$ for any package $S$. We seek for truthful and non-dictatorial mechanisms to maximize social welfare.

3 General Setting

Here we argue that the assumption of no upward bidding is not an escape route from the G-S impossibility result.

Gibbard-Satterthwaite theorem states that any truthful, and unanimous social choice function whose range comprises three or more alternatives is dictatorial [1, 2]. It is well known that the main reason for this impossibility result is the unrestricted domain of preferences. If the preference ordering of the agents on the set of alternatives (of size at least three) can be any ordering (unrestricted domain), then every non-dictatorial social choice function will be manipulable [6, 17].

For example, agents can announce any of the following ordering of preferences for three outcomes $a$, $b$ and $c$:

- $\succ^1: a \succ b \succ c$
- $\succ^2: a \succ c \succ b$
- $\succ^3: b \succ a \succ c$
- $\succ^4: b \succ c \succ a$
- $\succ^5: c \succ a \succ b$
- $\succ^6: c \succ b \succ a$

When there is no prior restriction over preferences to be exploited by the mechanism designer in order to exclude some of the orderings above, then the G-S theorem holds.

Single-peaked preferences as a restricted domain allow non-dictatorial social functions because of the restriction over the preferences. With single-peaked preferences, most of the relevant information about a particular preference is described by its “peak” alternative. Therefore it is natural to ask each agent to simply announce her peak alternative, restricting the domain of preferences substantially. In the example above, if we assume that $b$ is the peak, the most preferred outcome for the agent, then only preferences $\succ^3$ and $\succ^4$ constitute the domain of admissible preferences.

The restriction of no upward bidding does not enable us to exclude any of the possible orderings over outcomes. Therefore the G-S theorem continue to hold. For example, consider a setting with two agents and the set of items $\{A, B, C, D\}$. Valuations are $v_1(AB) = 9$, $v_1(AC) = 7$, $v_1(AD) = 4$, $v_2(CD) = 3$, $v_2(BD) = 7$ and $v_2(BC) = 8$. Valuations of the agents for other packages not containing aforementioned packages are zero. The possible alternatives are
\[ a = ((1 : AB), (2 : CD)), b = ((1 : AC), (2 : BD)), \text{ and } c = ((1 : AD), (2 : BC)). \]

We see that all preferences \( \succ^1 \text{ to } \succ^6 \) are admissible from agents even under the assumption of no upward bidding. For example while the true preference ordering of agent 1 is \( \succ^1 \), the agent can announce preference \( \succ^3 \) by reporting \( v_1(AB) = 3, v_1(AC) = 2 \) and \( v_1(AD) = 4 \) without violating the no upward bidding assumption. A formal proof of this fact is deferred to an extended version of the paper.

Knowing this fact and since we wish to maximize social welfare a good choice is to assign the grand bundle to a single agent that has the highest value for the grand bundle.

### 3.1 Randomized Mechanism

The G-S theorem does not exclude the existence of randomized algorithms. Here we show that an algorithm by Dobzinski et al. [18] which is introduced for quasi-linear bidders can be adapted to use in our model. Before presenting the algorithm we need two observations.

Let us define a first-price auction with a reserve price as follows. Assign the grand bundle (the bundle of all items) to the agent whose valuation for the grand bundle is higher than the reserve price and is the highest among all agents; if there is no such an agent, no item is assigned. Obviously, in this type auction, downward bidding has no benefit to agents.

**Observation 1** *First price auction with a reserve price is truthful for our model.*

Next we define a fixed-price auction as follows. Set a fixed price for every item. Agents arrive in an arbitrarily order and take their highest-valued package whose price (sum of the price of the items contained in the package) is less than or equal to their value for the package. Since agents wish to obtain the highest-valued package there is no gain in downward bidding. Thus,

**Observation 2** *Fixed-price auction is truthful for our model.*

We now describe the adapted mechanism. The key modification to the algorithm proposed by Dobzinski et al. is to conduct a first-price auction with a reserve price rather than a second-price auction with a reserve price. A full version of the adapted algorithm is relegated to Appendix A.

First, agents are partitioned randomly in three sets according to predefined probabilities. Let us call the sets STAT, FIRST-PRICE and FIXED-PRICE. Agents in STAT do not get assigned any item and are used to gather statistics including a reserve price and a fixed price. After calculating a fixed and reserve price from agents in STAT a first-price auction with the calculated reserve price is conducted for agents in FIRST-PRICE only. Finally, if the grand bundle is not assigned in the first-price auction, a fixed price auction is conducted for the agents in FIXED-PRICE.

We show that the foregoing mechanism is *universally truthful*. A randomized mechanism is universally truthful, if agents maximize their utility by reporting
their true bids for any fixed outcome of the random choices made by the mechanism. We show that the mechanism is truthful in each of the three partitions of bidders STAT, FIRST-PRICE, and FIXED-PRICE over which the mechanism randomizes.

Agents in STAT never receive any item, and thus have no incentive to misreport their types. Agents in FIRST-PRICE and FIXED-PRICE are truthful according to Observation 1 and Observation 2, respectively.

Since the adaptation made in the algorithm makes no loss in the welfare guaranteed by the original algorithm, we can draw on the proof of the original algorithm and show the same approximation ratio.

**Theorem 3.** There exists a polynomial-time randomized mechanism for our model which is universally truthful and guarantees an approximation ratio of $O\left(\frac{\sqrt{m}}{\epsilon^3}\right)$ with probability at least $1 - \epsilon$, for $0 < \epsilon < 1$.

4 Markets with Two Agents

Here, we analyze the $2 \times 2$ markets with two identical items and two agents. For the simpler case of single-item markets, a greedy algorithm which assigns the item to the agents with the highest value is truthfull as well as value maximizing.

Below we describe the deterministic and truthful mechanism for the $2 \times 2$ market. Let $\Phi$ denote the golden ratio conjugate, $\Phi = \frac{\sqrt{5} - 1}{2}$.

<table>
<thead>
<tr>
<th><strong>Algorithm 1:</strong> Golden Ratio Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Case I: if $v_1(2) \geq v_2(2)$ then</td>
</tr>
<tr>
<td>if $v_2(1) &lt; \Phi \cdot v_1(2)$ assign both items to agent 1.</td>
</tr>
<tr>
<td>else assign one item to agent 2; assign the other item to agent 1 if $v_1(1) &gt; 0$.</td>
</tr>
<tr>
<td>end</td>
</tr>
<tr>
<td>2. Case II: if $v_2(2) &gt; v_1(2)$. symmetric to Case I.</td>
</tr>
</tbody>
</table>

The assumption that upward bidding is not allowed, is necessary for the the truthfulness of the golden ratio algorithm.

*Example 1.* Consider the true values in Table 1.

| Rows are representing agents’ valuations and columns are | thus each cell shows the value of the corresponding row agent for the number of units in the column. For example, in the Table 1 the value of agent 2 for one unit is $v_2(1) = 55$. |

<table>
<thead>
<tr>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rows</td>
</tr>
<tr>
<td>Agent 1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
</tbody>
</table>

With these bids, the algorithm assigns both units to agent 1. But without the upward bidding restriction, agent 2 could get one unit by pretending $v_2(2) = 101$, shown in Table 2.

**Theorem 4.** The golden ratio algorithm is an individually rational, truthful and $\Phi$-approximation mechanism.
Truthful Mechanism Design without Money: No Upward Bidding

Proof. We show the properties of the algorithm for bidder 1; the argument for bidder 2 is analogue.

For truthfulness, we assume bidder 1 gets \( j \in \{0,1\} \) units assigned and show that she cannot get \( j' > j \) units by manipulating her bids.

First, assume \( j = 1 \) and the bidder wants to get \( j' = 2 \) units.

In case I of Algorithm 1, getting 2 units requires \( v_2(1) < \Phi \cdot v_1(2) \). Yet, bidder 1 cannot change bid to make this happen as it depends on \( v_2(1) \) and upward bidding is eliminated by assumption. In case II, bidder 1 can never get 2 units.

Now, we assume that the algorithm assigns no unit to the bidder \( j = 0 \), yet the bidder wants to get \( j' \in \{1,2\} \) units. In case I, getting 0 unit implies \( v_2(1) > \Phi \cdot v_1(2) \) and \( v_1(1) = 0 \). The point that \( v_1(1) = 0 \) shows that the bidder is not truly interested in getting 1 unit. So, she may want to win 2 units. In case I, she cannot change the bid to get 2 units as it depends on \( v_2(1) \) and upward bidding is eliminated. In case II, the bidder can never get 2 units. Getting one unit in case II requires \( v_1(1) > \Phi \cdot v_2(2) \). The bidder cannot make this happen as it depends on \( v_2(2) \) and upward bidding is eliminated.

Regarding the approximation ratio of the algorithm, in case I, first assume \( v_1(2) \geq v_1(1) + v_2(1) \) (the optimal value will then be \( v_1(2) \)). Then, it is easy to see that the lower bound is respected in any assignment of the algorithm. Second, assume \( v_1(2) < v_1(1) + v_2(1) \). Bidder 1 may obtain \( j \in \{1,2\} \) units. It is easy to observe that when bidder 1 obtains 1 unit, bidder 2 gets 1 unit, as well. This outcome will be optimal. If bidder 1 gets both units, we must have \( v_2(1) < \Phi \cdot v_1(2) \). Thus the ratio of the algorithm will be \( \frac{v_1(2)}{v_1(1) + v_2(1)} \geq \frac{v_1(2)}{v_1(2) + \Phi \cdot v_1(2)} = \Phi \).

2 In the inequality we used the monotonicity assumption \( v_1(1) \leq v_1(2) \). The analysis for case II is analogue. This completes the proof. \( \square \)

As the next theorem shows, this ratio of value is tight and no deterministic, and truthful mechanism can achieve more even under our assumption that upward bidding is eliminated. Also it states that no optimal randomized mechanism exists.

Theorem 5. There is no deterministic and truthful mechanism for the 2 \( \times \) 2 market with an approximation ratio better than \( \Phi + \epsilon \), for any \( \epsilon > 0 \). Moreover, no truthful-in-expectation randomized mechanism gives better than a 0.917 approximation.

Proof. Consider bids shown in Table 3.

\[ \begin{array}{c|c|c|}
   & j = 1 & j = 2 \\
   \hline
   v_1(j) & 64 & 100 \\
   v_2(j) & 55 & 60 \\
   \end{array} \]

Table 1. True bids.

\[ \begin{array}{c|c|c|}
   & j = 1 & j = 2 \\
   \hline
   v_1(j) & 64 & 100 \\
   v_2(j) & 55 & 101 \\
   \end{array} \]

Table 2. Manipulated bids.

\[ \frac{1}{1 + \Phi} = \Phi. \]
Considering such bids, giving both items to agent 1 by an arbitrary mechanism will result in a ratio of $\Phi$, since $x/(\Phi \cdot x + \epsilon + x - \epsilon) = 1/(1 + \Phi) = \Phi$. Thus, we choose to give one item to each agent.

Now consider the new bids shown in Table 4. Agent 1 here shades her preference for 1 unit. Here, the best value is given by the assignment of two items to agent 1, but this cannot happen because then agent 1 would have an incentive to pretend having values as in Table 4 while her true value is as given in Table 3. Thus, the best possible ratio in this case will be $\Phi \cdot x + \epsilon \approx \Phi + \epsilon$.

For randomized mechanisms, we consider the bids given below.

Let us assume an arbitrary mechanism for the true bids in Table 5 assigns one unit to each agent with probability $q$ and two units to agent 1 with probability $1 - q$. The ratio of such a mechanism will be $r_1 = q(2\Phi x)/(1-q)x = q(1-q)/(2\Phi)$. The utility of agent 1 in this case will be $u_1 = q(\Phi x) + (1-q)x$. Similarly, we consider the bids in Table 6. Let us assume that the mechanism assigns one unit to agent 2 with probability $p$ and two units to agent 1 with probability $1 - p$. We obtain $r_2 = p(\Phi x)/(1-p)x = p \cdot \Phi + (1-p)x$ and $u_1' = (1-p)x$, as the approximation ratio and utility of agent 1, respectively.

Truthfulness demands that $u_1 \geq u_1'$. Therefore, $p \geq q(1 - \Phi)$. Now we try to satisfy this inequality and at the same time maximize the minimum of $r_1$ and $r_2$. It turns out that these objectives are achieved when $p = q(1 - \Phi)$ and $r_1 = r_2$. This leads to $r_1 = r_2 \approx 0.917$, the desired conclusion.

In what follows, we present a randomized truthful mechanism for $2 \times 2$ markets which yields a better approximation factor.
Algorithm 2: Randomized 2 × 2 Algorithm.

if $v_i(2) \geq v_j(2)$ then
Let $q = \frac{v_i(1)}{v_i(2)}$;
with probability $q$ do as follows: assign agent $j$ a single unit; assign
agent $i$ a single unit if $v_i(1) > 0$.
with the complementary probability of $1-q$ assign both units to
agent $i$.
end

Theorem 6. There is a randomized algorithm for 2×2 markets which is truthful-in-expectation and $3/4$-approximation.

Proof. First we analyze the approximation ratio. We assume w.l.o.g that $v_1(2) \geq v_2(2)$. Let us use a simpler notation by setting $\beta = v_1(1)$, $\theta = v_1(2)$, and $\alpha = v_2(1)$ as shown in Table 7.

<table>
<thead>
<tr>
<th>$i$</th>
<th>$\beta$</th>
<th>$\theta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>\beta</td>
<td>\theta</td>
</tr>
<tr>
<td>2</td>
<td>\alpha</td>
<td></td>
</tr>
</tbody>
</table>

Table 7. Notation for bids

In order to see the ratio of the mechanism we consider two cases: $\theta \geq \alpha + \beta$, and $\theta < \alpha + \beta$. First, $\theta \geq \alpha + \beta$. Here the ratio of the mechanism will be

$$\frac{\frac{\alpha}{\theta}(\alpha + \beta) + (1 - \frac{\alpha}{\theta})\theta}{\theta} \geq \frac{\alpha^2}{\theta} - \frac{\alpha}{\theta} + 1 \geq \frac{3}{4}.$$ 

The first inequality is because $\beta \geq 0$. Second, $\alpha + \beta > \theta$. Then the ratio of the mechanism will be

$$\frac{\frac{\alpha}{\theta}(\alpha + \beta) + (1 - \frac{\alpha}{\theta})\theta}{\alpha + \beta} \geq \frac{\alpha}{\theta} + \frac{\theta - \alpha}{\theta + \alpha} \geq 2(\sqrt{2} - 1).$$

The first inequality is because $\beta \leq \theta$, and the second inequality is because the minimum of the expression occurs when $\alpha = (\sqrt{2} - 1)\theta$. Thus, the minimum ratio will be $3/4$.

For truthfulness, first we define the stronger bidder as the one who has the highest bid for 2 units. Pretending to be the stronger bidder is ruled out by our restriction to upward bidding. Pretending to be the weaker bidder is never profitable to the stronger bidder. The weaker bidder never obtains 2 units. Moreover, the stronger bidder who has also the 1 unit in her demand, obtains it with probability one. This also shows that a stronger bidder has no incentive to change her bid for one unit.

The stronger bidder $i$ also cannot improve the probability of obtaining two units as it depends on $v_j(1)$ and upward bidding is eliminated.
The weaker bidder \( j \) obtains one unit with the probability that is depending on \( v_i(2) \) and public value \( v_j(1) \) and therefore non-manipulable by the bidder. This completes the proof. \( \square \)

A tight example for the randomized algorithm is when the non-zero bids are \( v_1(2) = x \) and \( v_2(1) = x/2 \).

The following theorem shows that the golden ratio mechanism cannot be extended to \( 2 \times m \) markets, for any \( m \geq 3 \).

**Theorem 7.** There is no deterministic and truthful mechanism for the \( 2 \times m \) market for any \( m \geq 3 \) with an approximation ratio better than \( \left( \frac{1}{2} + \epsilon \right) \) for any \( \epsilon > 0 \).

**Proof.** Suppose there is. Consider the bids given in Table 8.

<table>
<thead>
<tr>
<th>( j )</th>
<th>( v_1(j) )</th>
<th>( v_2(j) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( x )</td>
<td>( x + \epsilon )</td>
</tr>
<tr>
<td>2</td>
<td>( x + \epsilon )</td>
<td>( x + 2\epsilon )</td>
</tr>
<tr>
<td>3</td>
<td>( x + 3\epsilon )</td>
<td>( x + 3\epsilon )</td>
</tr>
</tbody>
</table>

**Table 8.** True bids.

<table>
<thead>
<tr>
<th>( j )</th>
<th>( v_1(j) )</th>
<th>( v_2(j) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( x )</td>
<td>( x + \epsilon )</td>
</tr>
<tr>
<td>2</td>
<td>( x + \epsilon )</td>
<td>( x + 2\epsilon )</td>
</tr>
<tr>
<td>3</td>
<td>( x + 2\epsilon )</td>
<td>( x + 3\epsilon )</td>
</tr>
</tbody>
</table>

**Table 9.** Manipulated bids.

An arbitrary mechanism must assign one unit to one bidder and two units to the other bidder, otherwise the mechanism returns a low ratio of \( \left( \frac{1}{2} + \epsilon \right) \) and the theorem holds. Assume w.l.o.g. one unit is assigned to bidder 2 and two units to bidder 1.

Now, consider the new bids in Table 9. If bidder 2 obtains two or three units with the new bids, then she would have an incentive to lie when her true bids were as given in Table 8. Therefore, the only solution to the new bids is the assignment of three units to bidder 1. However, this assignment is only \( \left( \frac{1}{2} + \epsilon \right) \) approximate, contradicting the assumption.

<table>
<thead>
<tr>
<th>( j )</th>
<th>( v_1(j) )</th>
<th>( v_2(j) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k )</td>
<td>( x )</td>
<td>( x + \epsilon )</td>
</tr>
<tr>
<td>( m - k )</td>
<td>( x + \epsilon )</td>
<td>( x + 2\epsilon )</td>
</tr>
<tr>
<td>( m )</td>
<td>( x + 3\epsilon )</td>
<td>( x + 3\epsilon )</td>
</tr>
</tbody>
</table>

**Table 10.** True bids.

<table>
<thead>
<tr>
<th>( j )</th>
<th>( v_1(j) )</th>
<th>( v_2(j) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k )</td>
<td>( x )</td>
<td>( x + \epsilon )</td>
</tr>
<tr>
<td>( m - k )</td>
<td>( x + \epsilon )</td>
<td>( x + 2\epsilon )</td>
</tr>
<tr>
<td>( m )</td>
<td>( x + 2\epsilon )</td>
<td>( x + 3\epsilon )</td>
</tr>
</tbody>
</table>

**Table 11.** Manipulated bids.

For \( m > 3 \), an analogue argument by considering the true and manipulated bids shown in Table 10 and 11 proves the claim. \( \square \)

5 Generalized Assignment Problem

Here, we consider a strategic variant of the *generalized assignment problem* where each machine is held by an agent, GAP-MS.
In GAP-MS, there are $m$ jobs $J$ and $n$ machines $I$. Each machine $i$ has a capacity $C_i$ and associates a value $v_{ij}$ and a size $w_{ij}$ to any job $j$. A feasible assignment may allocate a subset of jobs $S$ to machine $i$ such that $\sum_{j \in S} w_{ij} \leq C_i$. The value of a feasible set $S$ is additive in values of the jobs in the set, $v_i(S) = \sum_{j \in S} v_{ij}$.

In GAP-MS, we assume tuple $\{v_{ij}\}_{ij}, \{w_{ij}\}_{ij}, \{C_i\}_i$ is public. The private information that each agent/machine holds is the set of its compatible jobs. This can be best viewed by a bipartite graph $G$ where one side corresponds to jobs and the other side corresponds to machines. The edges of $G$, $E \subseteq I \times J$ represent the compatible job-machine pairs. The private type of a machine $i$ is therefore the set of edges in the graph incident on $i$, $E_i$. The LP relaxation of the problem, $\text{LP}[E]$, is shown below as program (1) – (5).

\[
\text{Maximize } \sum_{i=1}^{n} \sum_{j=1}^{m} v_{ij} x_{ij} \tag{1}
\]

subject to
\[
\sum_{i=1}^{n} x_{ij} \leq 1 \quad \forall j \in J \tag{2}
\]
\[
\sum_{j=1}^{m} w_{ij} x_{ij} \leq C_i \quad \forall i \in I \tag{3}
\]
\[
x_{ij} \geq 0 \quad \forall i, j \tag{4}
\]
\[
x_{ij} = 0 \quad \forall (i, j) \notin E. \tag{5}
\]

Each machine $i$ reports a subset of compatible jobs and would like to obtain a feasible subset of compatible jobs with maximum possible value. As machines are strategic, they may report compatible jobs untruthfully in order to improve their own value. Formally expressed, a machine $i$ may report $E'_i$ and not $E_i$. Our objective is to design a truthful mechanism which maximizes total value.

We remark that the solutions presented in Dughmi and Ghosh [9] and Chen et al. [10] where jobs are held by agents are not directly applicable to GAP-MS, yet in our solutions to GAP-MS we benefit from the techniques developed therein.

Our technique is as follows. We design a fractionally truthful approximation algorithm for the problem which returns a feasible solution to $\text{LP}[E]$. A fractionally truthful algorithm, allocates fractional assignments to machines and no machine can improve its fractional value by untruthful bidding. Next, we round the solution using a special rounding technique which makes sure that each machine obtains a fixed fraction of its fractional value in expectation. The rounding technique originally proposed by Carr and Vempala [19] later applied by Lavi and Swamy [20] to mechanism design can do so. The rounding method is termed randomized meta-rounding. Recently, more practical implementations of the randomized meta-rounding have been proposed to substitute usage of the Ellipsoid method. [21, 22].
To use the randomized meta-rounding, we have to scale down the fractional solution by factor 2, which is essentially the integrality gap of the LP[\(E\)], program (1) – (5). Using this technique the following theorem holds.

**Theorem 8.** Given a fractional truthful \(\alpha\) approximate solution to LP[\(E\)], there exists a truthful-in-expectation \(2\alpha\) approximate solution to GAP-MS.

The theorem above has also been previously observed in [9].

### 5.1 Multiple Knapsack Problem

We first consider a special case of GAP-MS in which neither the size nor the value of the job depends on the machine. Formally, for each job \(j\) we have \(v_{ij} = v_j\) and \(w_{ij} = w_j\) for all machines \(i\). First, we observe that simply returning the (fractional) optimal solution via solving the corresponding LP relaxation is not truthful. This is shown in Figure 5.1, where machine 1 hides its compatibility with item A and as a consequence it is better off (in expectation) when the mechanism simply maximizes the social welfare. Note that in (b) the tie can be broken randomly or deterministically for example alphabetically in favor of machine 1. In any case machine 1 is better off by manipulation.

![Fig. 5.1. Circles represent jobs, squares represent machines, and value/size of each job is on its left. Value maximizing assignments are in bold.](image)

The proposed algorithm therefore is as follows. We choose machine \(i\) in an arbitrary order and (fractionally) assign compatible jobs to it according to the decreasing order of value densities of jobs \(v_j/w_j\) until the capacity of the machine is exhausted or all compatible jobs are exhausted. Then we proceed to the next machine with remaining (fractional) jobs.
Algorithm 3: Algorithm for the Multiple Knapsack Problem.

1. Sort jobs according to the decreasing order of value densities \( v_j/w_j \), breaking ties arbitrarily.

2. foreach machine \( i \) chosen in an arbitrarily order do
   
   For each unassigned (fractional) job \( j \) where \((i, j) \in E\) in the order defined above, fractionally assign as much of the job to machine \( i \) until the job is exhausted or the machine is full.

end

return the resulting assignment \( x \).

It is well known that assigning jobs according to the decreasing order of value densities, when fractional assignments are allowed, produces the highest fractional value for the machine. Since machines wish to maximize their value and the algorithm is aligned with this goal, the algorithm is fractionally truthful.

Moreover, we can show that Algorithm 3 returns a 2 approximate fractional solution. We compare the outcome of the algorithm with the optimal solution to the LP formulation of the problem. The LP formulation (MKP-LP[E]) of the problems is shown below.

\[
\text{MKP-LP[E]:}
\begin{align*}
\text{Maximize} & \quad \sum_{i=1}^{n} \sum_{j=1}^{m} v_j x_{ij} \\
\text{subject to} & \quad \sum_{i=1}^{n} x_{ij} \leq 1, \quad \forall j \in J \\
& \quad \sum_{j=1}^{m} w_j x_{ij} \leq C_i, \quad \forall i \in I \\
& \quad x_{ij} \geq 0, \quad \forall i, j \\
& \quad x_{ij} = 0, \quad \forall (i, j) \notin E.
\end{align*}
\]

Lemma 1. Algorithm 3 returns a 2 approximate solution to MKP-LP[E].

Proof. We will construct a feasible dual solution with value at most twice the value obtained by the algorithm, then calling the weak duality theorem, the claim will follow.

Assume \( x \) is the outcome of the Algorithm 3. Using \( x \) we can construct a feasible solution to the dual of MKP-LP[E] (MKP-LPD[E] given below).
MKP-LPD[E]:

\[
\begin{align*}
\text{Minimize} & \quad \sum_{j=1}^m p_j + \sum_{i=1}^n u_i C_i \\
\text{subject to} & \quad p_j + u_i w_j \geq v_j, \quad \forall (i,j) \in E \\
& \quad u_i \geq 0, \quad \forall i \\
& \quad p_j \geq 0, \quad \forall j.
\end{align*}
\]

Initially, let \( p = 0 \) and \( u = 0 \). If job \( j \) gets exhausted set \( p_j = v_j \). Furthermore, for all full machines \( i \), set \( u_i = v_j / s_j \), \( j \) being the last job (fractionally) assigned to \( i \). We can observe that this satisfies the constraint corresponding to each edge \((i,j)\). In particular, if machine \( i \) is full, then for each \( j \) incident on \( i \), either \( j \) gets exhausted with this assignment or does not. If \( j \) is exhausted we have \( p_j = v_j \) and therefore the constraint holds. If \( j \) is not exhausted, we have \( v_j / w_j \leq u_i \) since jobs are assigned in decreasing order of value density and thus the constraint holds. If machine \( i \) is not full, every job \( j \) which is assigned to it is exhausted by this assignment. That is we have \( p_j = v_j \) and the constraint thus holds. In sum, we have constructed a feasible dual solution using \( x \).

Now, we bound the value of the dual solution with respect to the primal solution. First, we observe that \( \sum_{i,j} v_j x_{ij} \geq \sum_j p_j \sum_i x_{ij} \), since \( p_j = v_j \) if \( j \) is fully exhausted and \( p_j = 0 \), otherwise. Second, \( \sum_{i,j} v_j x_{ij} = \sum_i \sum_j \frac{v_j}{w_j} (w_j x_{ij}) \geq \sum_i u_i \sum_j (w_j x_{ij}) \), since if \( x_{ij} > 0 \) then \( v_j / w_j \geq u_i \). Therefore, we obtain

\[
2 \sum_{i,j} v_j x_{ij} \geq \sum_j p_j \sum_i x_{ij} + \sum_i u_i \sum_j (w_j x_{ij}) = \sum_j p_j + \sum_i u_i C_i.
\]

Notice, only for jobs \( j \) which get exhausted (\( \sum_i x_{ij} = 1 \)) we have \( p_j > 0 \) and only for full machines (\( \sum_j w_j x_{ij} = C_i \)) we have \( u_i > 0 \). The final term is the value of the dual, the desired conclusion.

A tight example for the algorithm is shown in Figure 5.2.

Finally, we call Theorem 8 and obtain the following.

**Theorem 9.** There exists a truthful-in-expectation 4 approximate mechanism for the multiple knapsack problem in our model.

### 5.2 Truthful Mechanism for GAP-MS

Now, we attempt to design a truthful algorithm for the GAP-MS, yet first we solve the problem with an additional assumption. We assume that the value density of each job is the same over all machines. More formally, there exists a value \( d_j \) for each job \( j \) such that for all machines \( i \), we have \( \frac{v_{ij}}{w_{ij}} = d_j \). This
assumption will be relaxed in Subsection 5.3. Hence, we first design a truthful 4 approximate mechanism for GAP-MS under this extra assumption.

The proposed algorithm can be viewed as a variant of the deferred acceptance algorithm designed for matching marketplaces. Each job $j$ has a preference list $\mathcal{L}_j$ according to decreasing order of $v_{ij}$ where $(i, j) \in E$, breaking ties arbitrarily. The preference list of a machine is defined according to the decreasing order of value densities.

**Algorithm 4:** Algorithm for GAP with Equal Density

| **Data:** Preference lists of the jobs, $\{\mathcal{L}_j\}_j$. |
| **Result:** A feasible solution $x$ to LP$[E]$. |

1. Sort jobs according to their decreasing order of value densities $d_j$, breaking ties arbitrarily.
2. **foreach** job $j$ chosen according to the order above **do**
   - Fractionally assign as much of the job to the machines chosen according to the order specified by $\mathcal{L}_j$, until the job is exhausted or all the machines in $\mathcal{L}_j$ are full.
**end**

**return** the resulting assignment $x$.

We can construct a feasible dual solution with value at most twice the value obtained by the algorithm, then by calling the weak duality theorem, the following lemma holds.

**Lemma 2.** Algorithm 4 returns a 2 approximate solution to LP$[E]$ when each job has the same value density over machines.

**Proof.** An argument similar to that of Lemma 1 in addition to some required modifications will show the claim.

Assume $x$ is the outcome of Algorithm 4. Using $x$ we can construct a feasible solution to the dual of LP$[E]$ (LPD$[E]$ given below) which is not greater than twice the value of $x$. Then we call the weak LP-duality theorem and conclude...
that $x$ is a 2 approximate solution to LP$[E]$.

$$\text{LPD}[E]: \begin{align*}
\text{Minimize} & \quad \sum_{j=1}^{m} p_j + \sum_{i=1}^{n} u_i C_i \\
\text{subject to} & \quad p_j + u_i w_{ij} \geq v_{ij}, \forall (i,j) \in E \\
& \quad u_i \geq 0, \quad \forall i \\
& \quad p_j \geq 0, \quad \forall j.
\end{align*}$$

Initially, let $p = 0$ and $u = 0$. If job $j$ gets exhausted when assigned to machine $i$, set $p_j = v_{ij}$. Furthermore, for all full machines $i$, set $u_i = d_j$, $j$ being the last job (fractionally) assigned to $i$. We can observe that this satisfies the constraint corresponding to each edge $(i,j)$. In particular, if machine $i$ is full, then for each $j$ incident on $i$, either $j$ gets exhausted with this assignment or does not. If $j$ is exhausted we have $p_j = v_{ij}$ and therefore the constraint holds. If $j$ is not exhausted, we have $v_{ij} / w_{ij} = d_j \leq u_i$ since jobs are assigned in decreasing order of value density and thus the constraint holds. If machine $i$ is not full, every job $j$ which is assigned to it is exhausted by this assignment. That is we have $p_j = v_{ij}$ and the constraint thus holds. For every job $j$ which is not assigned to the machine but $(i,j) \in E$, we have $p_j \geq v_{ij}$ since the job is exhausted due to assignment to a machine of higher value.

Therefore, we have constructed a feasible dual solution using $x$.

Now, we bound the value of the dual solution with respect to the primal solution. First, we observe that $\sum_{i,j} v_{ij} x_{ij} \geq \sum_{i} p_j \sum_{j} x_{ij}$, since $p_j$ lower bounds the value of any edge on which any part of job $j$ is assigned because the job goes to machines according to the order specified by $L_j$. Second, $\sum_{i,j} v_{ij} x_{ij} = \sum_{i} \sum_{j} \frac{v_{ij}}{w_{ij}} (w_{ij} x_{ij}) \geq \sum_{i} u_i \sum_{j} (w_{ij} x_{ij})$, since if $x_{ij} > 0$ then $\frac{v_{ij}}{w_{ij}} = d_j \geq u_i$. Therefore, we obtain

$$2 \sum_{i,j} v_{ij} x_{ij} \geq \sum_{j} p_j \sum_{i} x_{ij} + \sum_{i} u_i \sum_{j} (w_{ij} x_{ij}) = \sum_{j} p_j + \sum_{i} u_i C_i.$$

Notice, only for job $j$ which gets exhausted ($\sum_{i} x_{ij} = 1$) we have $p_j > 0$ and only for full machines ($\sum_{j} w_{ij} x_{ij} = C_i$) we have $u_i > 0$. The final term is the value of the dual, the desired conclusion.

Regarding the truthfulness of the algorithm, it would have been easier to prove the truthfulness if we - similar to the algorithm for the multiple knapsack problem - allowed the machines to propose to jobs, yet as shown in Fig. 5.3 this might result in an arbitrarily low social value. In this example when machine 1 starts choosing its desired jobs, it obtains all jobs and the resulting allocation will be of very low value.
Now we show that Algorithm 4 is \textit{fractionally truthful}. To this end, we look at Algorithm 4 as a variant of the deferred acceptance algorithm where jobs propose capacities to machines. Notice, this is in contrast to Algorithm 3 where machines propose to jobs. Each machine then may accept or reject the whole or part of the proposed capacity by a job depending on its current empty capacity. Once a (fractional) job and a machine are matched, the assignment will never be broken.

Let $C_{ij}$ denote the capacity proposed by job $j$ to machine $i$. We observe that the proposed capacities to a machine are all accepted by the machine except for the last one which is accepted/rejected partly when the machine gets full.

In order to show truthfulness, we first show that in an instance with 2 jobs and 2 machines ($2 \times 2$), truthfulness actually holds. This instance contains the core of truthfulness proof for the general case. Truthfulness for simpler cases is trivial. Then a straightforward generalization of the argument for $2 \times 2$ shows truthfulness for settings with $m$ jobs and 2 machines ($2 \times m$), for any $m > 2$. For the general case of $(n \times m)$ we give an inductive argument.

\textbf{Lemma 3.} Algorithm 4 is fractionally truthful for a setting with 2 jobs and 2 machines.

\textit{Proof.} Let 1, 2, $p$, and $q$ denote the machines and jobs, respectively. Let us assume $p$ precedes $q$ in proposing to the machines, i.e. $d_p \geq d_q$. Fix the order of proposing jobs as well as the bids by machine 2. We argue that machine 1 is never better off by manipulating $E_1$, the set of its compatible jobs.

Machine 1 wouldn’t wrongly put itself on the preference list of any job (e.g. by reporting $E'_1 \supset E_1$) since the algorithm might assign the job to the machine and this is just a waste of capacity for the machine as the job brings zero utility for it.\footnote{One may assume that this will cause a negative utility for the machine. See [10].} Next, we show that the machine wouldn’t report $E'_1 \subset E_1$ which completes the argument for truthfulness.

Assume $(1,q) \in E_1$. Then machine 1 may receive a proposal from $q$ but obviously the machine receives no proposal from the job if the machine reports...
(1, q) \notin E_1. Thus hiding compatibility in this case might only make a loss for the machine.

Now, we analyze the behavior of the algorithm for a similar change in bid for job p. We need to show that when (1, p) \in E_1 (case I) the obtained value is at least as good as when (1, p) \notin E_1 (case II) for the machine. Then we conclude that when truly (1, p) \in E_1, the machine has no incentive to report (1, p) \notin E_1.

In case I if no fraction of p is assigned to machine 1, then everything remains the same as case II. If only a fraction of p is assigned to the machine, then the machine has to be full thus obviously the utility of the machine is maximum and can’t be better off in case II. What remains is to show that the machine is not worse off when it accepts p fully in case I.

In case I if no fraction of p is assigned to machine 1, then everything remains the same as case II. If only a fraction of p is assigned to the machine, then the machine has to be full thus obviously the utility of the machine is maximum and can’t be better off in case II. What remains is to show that the machine is not worse off when it accepts p fully in case I.

![Diagram](image_url)

(a) case I. p is exhausted when it is assigned to 1. 1 may get a fraction or nothing from q.

(b) case II. 1 is not in the preference list of p. At least a fraction of q is assigned to 1. p is (fully) accepted by 2.

**Fig. 5.4.** Two cases where the machine is and is not on the preference list of the job. The amount of proposed capacities are shown on edges.

The situation is depicted in Figure 5.4. Considering the information provided in Fig. 5.4, we need to show that C'_{1q} \leq C_{1q} + C_{1p}. This will mean, in case II, the machine actually receives less capacity from jobs with less (or equal) value densities than in case I, which in turn means lower value for machine 1. Notice, to come up with this inequality we used the assumption that the order of proposing jobs is fixed in the two setup. To show the inequality we observe two facts about Algorithm 4.

**Observation 10** Assume currently a set of (fractional) jobs are assigned to a machine by the algorithm. If we increase the proposed capacity to the machine by C and let the previously proposing jobs propose the capacities as before to the machine after new capacity C, the machine then will reject a capacity of at most C from the next proposing jobs.

**Proof.** If the machine is currently full, it will reject exactly a capacity of C of the next jobs. If not (the machine still has an empty capacity of C_E), it accepts (a fraction of) the new capacity and rejects an amount equal to the max{0, C - C_E} of the next proposing capacities. Thus in both cases the rejected capacity will be lower bounded by C. \qed
**Observation 11** Let 1 and 2 be two subsequent machines in \( L_j \). If machine 1 rejects the proposed capacity \( C_{1j} \) by job \( j \) then this is an upper bound to \( C_{2j} \), the capacity that will be proposed by job \( j \) to 2, i.e. \( C_{1j} \geq C_{2j} \).

**Proof.** First, we must have \( \frac{w_{1j}}{v_{1j}} \geq \frac{w_{2j}}{v_{2j}} \) since \( \frac{v_{1j}}{w_{1j}} = \frac{v_{2j}}{w_{2j}} \) by the assumption of equal density over machines and \( v_{1j} \geq v_{2j} \) as 1 precedes 2 in \( L_j \). Rejecting \( C_{1j} \) means that this fraction of the job remains: \( C_{1j}/w_{1j} \). Then what will be proposed to 2 is \( C_{2j} = w_{2j} \cdot \left( \frac{C_{1j}}{w_{1j}} \right) \leq C_{1j} \).

Back to our goal, we notice that in case II there is an increase of amount \( C_{2p} \) in the proposed capacity to 2 compared to case I. The capacity rejected thus is upper bounded by \( C_{2p} \) according to Observation 10. Therefore, we have \( C_{2q} - C_{2p} \leq C_{2p} \). Moreover, according to Observation 11, the rejected capacity upper bounds the proposed capacity to the next machine. Hence we have, \( C_{1q} - C_{2q} \leq C_{2q} - C_{1p} \). Therefore, we obtain \( C_{1q} \leq C_{1q} + C_{2q} \leq C_{1q} + C_{1p} \). The last inequality holds again because of Observation 11. This completes the proof of Lemma 3. \( \square \)

Notice that in the proof of Lemma 3 we assumed a fix order of proposing jobs. We remark that this is in fact necessary for the proof. As otherwise, a machine might be better off under a different ordering of proposing jobs.

**Example 2.** In Figure 5.5 a change in the order of processing jobs in (b) will increase the utility of machine 2. In order to guarantee truthfulness we require a fixed order of jobs, therefore a consistent tie breaking rule in the algorithm is necessary.

![Fig. 5.5. Two cases where the machine is and is not on the preference list of the job.](image)

A simple generalization of the argument for \( 2 \times 2 \) shows truthfulness for the \( 2 \times m \) markets, \( m > 2 \). A useful observation here is that, we only need to show that machine 1 will always report \( E_1 \) rather than \( E_1 \setminus \{e_j\} \) for every \( e_j \in E_1 \). If we show this, we have in fact showed that reporting \( E_1 \) is better than reporting
This also shows that reporting \( E_1 \setminus \{ e_j \} \) is better than hiding one one edge from \( E_1 \setminus \{ e_j \} \), i.e. reporting \( E_1 \setminus \{ e_j, e_j' \} \) and so on.

For the general case we give an inductive argument. We assume that in a \((n-1) \times m\) setting machines are truthful and prove that in a \(n \times m\) setting this holds, as well.

**Lemma 4.** If Algorithm 4 is truthful for markets with \( m \) jobs and less than \( n \) machines, it will be truthful for \( n \times m \) markets, as well.

**Proof.** Consider machine \( i \) and fix the bids of other machines denoted by \(-i\). We assume \((i,p) \in E_i\) (case I) and show that the machine will never be better off by reporting \((i,p) \notin E_i\) (case II). We compare the utility of the machine in the two cases under a fixed order of proposing jobs. The two cases are depicted in Figure 5.6. Since the jobs before \( p \) are assigned similarly in both cases, we only consider the jobs which are processed after \( p \) denoted by \(-p\).

We show that \( C_{i,-p}' \leq C_{i,p} + C_{i,-p} \), where \( C_{i,-p} = \sum_{q \in -p} C_{i,q} \) and \( C_{i,-p}' = \sum_{q \in -p} C_{i,q}' \). This means that machine \( i \) in case II actually receives less capacity from jobs with less (or equal) value densities than case I which in turn implies lower value for the machine.

Consider case II. We look closer to the machine(s) to which job \( p \) will get assigned. We assume \( p \) is (fractionally) assigned to at least one machine otherwise we have \( C_{i,-p} = C_{i,-p}' \) and thus the claim holds. Let machine 1 be that machine.

Also, we assume that machine 1 (or more machines) gets full at some point otherwise this machine accepts the extra capacity \( (C_{1p}) \) without rejecting any job and therefore we have \( C_{i,-p} = C_{i,-p}' \) and thus the claim holds.

As a result, some of currently proposing jobs to machine 1 will stop proposing to it and go to the next machine in their preference list. Let us call these capacities as \( C_1 \). \( C_i \) is upper bounded by \( C_{1p} \) according to Observation 10 which in turn is upper bounded by \( C_{ip} \) based on Observation 11. This rejection of capacity is originally the consequence of strategic action of machine \( i \) who rejects
Taking this point of view, one can view this situation as machine $i$ rejecting capacity $C_i^*$ in a $(n - 1) \times m$ setting where machine 1 is eliminated as it is now full. According to our induction assumption, this strategy will not make machine $i$ better off in a $(n - 1) \times m$ setting.

If job $p$ actually goes to more than one machine, a similar argument for all rejected capacities by different machines and considering that we fall into a setting with fewer number of machines than $n$ (where we omit all full machines) holds. This completes the proof.

We can also give a simple inductive proof saying that no machine is willing to wrongly show compatibility with jobs, yet this is so intuitive that such a strategy will just waste the capacity of the machine, while other machines will continue accepting capacities as before and even with higher competition as some capacity is already wasted by the strategizing machine.

Taking into account, Lemma 3 and 4 we obtain the following.

**Lemma 5.** Algorithm 4 is fractionally truthful.

Finally, by calling Theorem 8, we obtain the following.

**Theorem 12.** There exists a truthful-in-expectation 4 approximate mechanism for GAP-MS when each job has the same value density over machines.

### 5.3 When Value Densities are not Equal

In the last subsection we presented a truthful-in-expectation 4-approximate mechanism for GAP-MS when each job has a unique value density over all machines. Now we explain how one can relax this assumption using a standard technique at the expense of a logarithmic loss in total value.

We assume there exist an upper bound and a lower bound on the value densities and these are publicly known. That is, there exist publicly known values $U$ and $L$ such that

$$L \leq \frac{v_{ij}}{w_{ij}} \leq U, \quad \forall i \in I, j \in J.$$

This assumption has been previously observed in the literature in the context of online algorithms [23].

Knowing this information we choose a density value $d$ uniformly at random from the set $D = \{U, \frac{U}{2}, \frac{U}{4}, \ldots, \frac{U}{2^{\Omega(\ln (U/L))}}\}$. Then we define a new valuation $\hat{v}$ as follows. For every edge $(i, j)$ in $E$ with $\frac{v_{ij}}{w_{ij}} < d$ we set $\hat{v}_{ij} = 0$, or equivalently the edge is discarded from the graph. For every $\frac{v_{ij}}{w_{ij}} \geq d$, define $\hat{v}_{ij}$ such that $\frac{\hat{v}_{ij}}{w_{ij}} = d$. Notice that always $\hat{v}_{ij} \leq v_{ij}$. Now we have an instance of GAP-MS with equal densities for which there exists a truthful-in-expectation 4-approximate mechanism according to Theorem 12. To ensure truthfulness, in the end when job $j$ is assigned to machine $i$ by the subroutine for equal value densities, we withdraw the job with probability $1 - \frac{v_{ij}}{\hat{v}_{ij}}$. In other words, we let the machine hold the job with probability $\frac{\hat{v}_{ij}}{v_{ij}}$. This way, we make sure that each job has
the same value density over all machines as it is required by the subroutine to

guarantee truthfulness.

Set $D$ contains $O(\ln (U/L))$ densities and each density has the probability of

$p = \frac{1}{O(\ln (U/L))}$ to be chosen. At least half of every valuation $v_{ij}$ with probability $p$

is counted in the expected total value, therefore, we obtain an $O(\ln (U/L))$

approximation factor. Thus, we obtain the following.

**Theorem 13.** There exists a truthful-in-expectation $O(\ln (U/L))$ approximate mechanism for GAP-MS.

We leave open the question of whether there exists a truthful mechanism

with a constant factor of approximation for GAP-MS.

**References**

A Randomized Mechanism for the General Setting

The Adaptation of the framework for randomized mechanisms by Dobzinski et al. is as follows.

**Algorithm 5:** Randomized approximation mechanism for the general setting

<table>
<thead>
<tr>
<th>Data:</th>
<th>$n$ bidders, each with valuation $v_i$, a rational number $0 &lt; \epsilon &lt; 1$.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Result:</td>
<td>An allocation of the items, which is an $O(\sqrt[3]{\epsilon m})$-approximation to the optimal value.</td>
</tr>
</tbody>
</table>

**Phase I: Partitioning the Bidders**

Assign each bidder to exactly one of the following three sets:
- FIRST-PRICE with probability $1 - \frac{\epsilon}{2}$,
- FIXED-PRICE with probability $\frac{\epsilon}{2}$, and
- STAT with probability $\frac{\epsilon}{2}$.

**Phase II: Gathering Statistics**

Calculate the value of the optimal fractional solution in the combinatorial auction with all $J$ items, but only with bidders in STAT. Denote this value by $OPT_{STAT}^\ast$.

**Phase III: A First-Price Auction**

Conduct a first-price auction with a reservation price of $r = \frac{OPT_{STAT}^\ast}{\sqrt[3]{m}}$, in which the bundle $J$ of all items is sold to the bidders in FIRST-PRICE. If there is a “winning bidder”, $i$, allocate all items to that bidder, and output this allocation. Otherwise, proceed to the next step.

**Phase IV: A Fixed-Price Auction**

Let $R = J$. Let $p = \frac{\epsilon OPT_{STAT}^\ast}{\sqrt{m}}$.

For each bidder $i \in$ FIXED-PRICE, in some arbitrary order:
- Let $S_i$ be the demand of bidder $i$ given the following prices:
  - $p$ for each item in $R$, and $\infty$ for each item in $J - R$.
- Allocate $S_i$ to bidder $i$.
- Let $R = R \setminus S_i$. 