Edge Manipulations for the Maximum Vertex-Weighted Bipartite b-matching

GENNARO AURICCHIO*, University of Bath, United Kingdom
JUN LIU, School of Mathematical Sciences, Beijing Normal University, China
QUN MA, Tianjin University, China
JIE ZHANG, University of Bath, United Kingdom

In this paper, we explore the Mechanism Design aspects of the Maximum Vertex-weighted b-Matching (MVbM) problem on bipartite graphs $(A \cup T, E)$. The set A comprises agents, while T represents tasks. The set E, which connects A and T, is the private information of either agents or tasks. In this framework, we investigate three mechanisms – $\mathbb{M}_{BFS}$, $\mathbb{M}_{DFS}$, and $\mathbb{M}_G$. We examine scenarios in which either agents or tasks are strategic and report their adjacent edges to one of the three mechanisms. In both cases, we assume that the strategic entities are bounded by their statements: they can hide edges, but they cannot report edges that do not exist. First, we consider the case in which agents can manipulate. In this framework, $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$ are optimal but not truthful. By characterizing the Nash Equilibria induced by $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$, we reveal that both mechanisms have a Price of Anarchy (PoA) and Price of Stability (PoS) of 2. These efficiency guarantees are tight; no deterministic mechanism can achieve a lower PoA or PoS. In contrast, the third mechanism, $\mathbb{M}_G$, is not optimal, but truthful and its approximation ratio is 2. We demonstrate that this ratio is optimal; no deterministic and truthful mechanism can outperform it. We then shift our focus to scenarios where tasks can exhibit strategic behaviour. In this case, $\mathbb{M}_{BFS}$, $\mathbb{M}_{DFS}$, and $\mathbb{M}_G$ all maintain truthfulness, making $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$ truthful and optimal mechanisms. In conclusion, we investigate the manipulability of $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$ through experiments on randomly generated graphs. We observe that (i) $\mathbb{M}_{BFS}$ is less prone to be manipulated by the first agent than $\mathbb{M}_{DFS}$, and (ii) $\mathbb{M}_{BFS}$ is more manipulable on instances in which the total capacity of the agents is equal to the number of tasks. [1]

CCS Concepts: • **Theory of computation → Algorithmic game theory**.

Additional Key Words and Phrases: Algorithmic Mechanism Design, Vertex Weighted Matching problems, Edge Manipulations

1 INTRODUCTION

Managers of large companies are periodically required to submit a list of the projects they carried out for external evaluations. Alongside the list of projects, a manager needs to identify a worker liable for the achievement of each project. Due to the workload cap, every worker can be found responsible for only a finite number of projects. Moreover, every project has a different prestige, which can be regarded as its overall value. The manager aims to maximize the total value of the reported projects by deploying its staff following the aforementioned constraints.

[1] A previous version of this paper has been published in the proceedings of the 26th European Conference on Artificial Intelligence ECAI 2023 [10]. This manuscript extends the previous work by addressing the case in which tasks exhibit strategic behavior and by supplementing our study with various experimental results.

Authors' addresses: Gennaro Auricchio, ga647@bath.ac.uk, webmaster@marysville-ohio.com, University of Bath, Bath, England, United Kingdom; Jun Liu, School of Mathematical Sciences, Beijing Normal University, Beijing, China, jliu@bnu.edu.cn; Qun Ma, Tianjin University, Tianjin, China; Jie Zhang, jz2558@bath.ac.uk, University of Bath, Bath, England, United Kingdom.

ACM 2157-6912/2024/11-ART
https://doi.org/10.1145/3702650

Meanwhile, every worker is interested in being associated with high-value projects rather than low-value ones. In this context, workers might benefit from hiding their connections to low-value projects. Indeed, by concealing certain connections, the worker might be able to compel the manager to match them to higher-valued projects in order to increase the overall quality of the report. A similar situation occurs in the universities of several European countries. Indeed, to assess the research impact of their higher education institutions, the country asks the university for a list of their best publications, along with the name of the main author.[2] On one hand, every university wants to find a matching between its lecturers and the publications that maximizes the total impact score. On the other hand, individual academics want to be designated as the main author of their best publications. In both scenarios, we need to allocate a set of resources that possess an objective and publicly known value, be the projects and their prestige or publications and their impact score, to a set of self-interested entities. Moreover, in both cases, agents are unable to falsely claim an association to a project or a paper to which they did not collaborate, as it would be easy to check. Thus the only way they have to manipulate the matching process is to hide their connection to some paper or work. A similar situation happens in Kidney Exchanges, where the patient cannot lie about their compatibility with a donor, but they can remove the connection with this donor by refusing to take part in the kidney exchange if they want to [45].

Aside from these two examples, there are many other real-life situations that can be rephrased as matching problems with self-interested agents. Matching problems were first introduced to minimize transportation costs [32, 36] and to optimally assign workers to job positions [21, 49]. Thereafter, bipartite matching found application in several applied problems, such as sponsored searches [4, 16], school admissions [1, 17], scheduling [30, 52], Ad Allocation [43], peer-reviewing [5, 11], and general resource allocation [26, 50]. Indeed, characterizing matching through the maximization or minimization of a functional allows us to define mathematical objects that arise in several applied contexts. For example, Maximum Cardinality Matchings have connections with the computation of perfect matchings [28], bottleneck matchings [22], and Lévy-Prokhorov distances [40]. Another example is the Maximum Edge-weighted Matching problem, which has been used in clustering problems [19], machine learning [14], and to compute Wasserstein-barycenters [9]. For a complete discussion of the matching problems and their applications, we refer to [41].

Game-theoretic aspects of matching problems

In many matching problems, the bipartite graph and the weights of its edges describe the preferences that a group of agents has over the members of a second, disjointed, group. This happens both in one-sided matching, *e.g.* house allocation [18, 33] or the resident/hospital problem [42]; and in two-sided matching, *e.g.* stable marriage [29], student admission [13], and market matching [37]. Starting from the basic models, several variants have been proposed, most of the time, these variants enforce some meaningful constraint, such as regional constraints [34], diversity constraints [23], or lower quotas [51]. We refer the reader to the survey by Aziz et al. for a complete introduction to the economic aspects of matching problems [12]. In all these frameworks, however, the preferences of each agent are reported by the agent themself. Therefore, studying the social aspects of implementing a mechanism to find an allocation is essential. To this extent, several works studied the manipulability of these class of problems [25, 27, 38], the fairness [24, 35, 39], and the envy-freeness [3]. For a comprehensive study of the game-theoretical aspects of the matching problems we refer the reader to the recent survey by Benedek et al. [15].

The problem that is closer to ours is the Maximum Cardinality Matching, which in Mechanism Design is also known as the Kidney Exchange Problem (KEP). In the KEP the goal is to match as many pairs of compatible individuals as possible using truthfulful routines. For this reason, the problem is formulated as a maximum cardinality matching on a graph in which nodes represent agents and edges represent the compatibility between

[2]This, for example, is a common practice in the United Kingdom, see https://www.ref.ac.uk for a reference.

agents [8]. Initially, kidney exchanges were organized using the "top trading cycles" concept [46], which was later expanded by Roth [44] and Abdulkadiroğlu and Sönmez [2] to create exchange cycles involving multiple patient-donor pairs. However, due to the complexity of these algorithms, subsequent research focused primarily on pairwise exchanges [45]. The authors showed that such mechanisms are truthful, marking the first paper to study the KDE from a mechanism design perspective. Starting from this paper, different alternative frameworks for this problem have been presented, leading to several truthful mechanisms that are able to handle various types of constraints [7, 31, 48].

In this paper, we deal with a different game-theoretical problem in which every strategic entity has the same preference order over the other side of the bipartite graph. Given that the preference orders over the other side of the bipartite graph are common and public information, we assume that the private information of the strategic entities consists of the set of edges connecting them to the other side of the graph, which is the main novelty of our framework. In this framework, we study both truthful and optimal mechanisms and provide positive and tight results on the efficiency guarantees. Notice that a preliminary version of this paper appeared in the proceedings of the 26th European Conference on Artificial Intelligence [10].

Our Contributions

Throughout the paper, we assume that the two sides of the bipartite graph consisting of a set of agents $A = \{a_1, \ldots, a_n\}$ and a set of tasks $T = \{t_1, \ldots, t_m\}$. We represent E as the set of edges connecting agents to tasks. Each task t_j is assigned a positive value, denoted as q_j, which is equal for all agents. Moreover, each task can be matched with at most one agent in set A. Each agent a_i is associated with a positive integer, denoted as b_i, which specifies the maximum number of tasks that agent a_i can be allocated.

We introduce and study two novel game-theoretical frameworks for this class of problems. In the first framework, the agents' set is composed of strategic entities (see Section 3), while in the second framework, the tasks are behaving strategically (see Section 4). Notice that, since the roles of agents and tasks in the MVbM problem are not interchangeable, the two frameworks are inherently different and yield different results. Our main focus, however, pertains to the study of the first framework, i.e. the case in which the agents are reporting their information, which is also more challenging. In both frameworks, we study three mechanisms induced by known algorithms. Two of them, namely $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$, are induced by the algorithm proposed by Spencer and Mayr [47]. The third one, namely $\mathbb{M}_G$, is induced by the algorithm proposed by Dobrian et al. [20], which is an approximation of the one proposed by Spencer and Mayr. Throughout the paper, we assume that the agents (or tasks) are bounded by their statements, hence they can hide edges or part of their capacity (value), but they cannot report non-existing edges or a capacity (value) higher than the real one.

Our framework shares similarities with the one used in the Kidney Exchange Problem (KEP), as both involve designing algorithms to match agents with compatible tasks or objects. However, our problem differs from KEP in two key ways: (i) In KEP, the matching is strictly pairwise, meaning agent i is matched with agent j only if j is also matched with i, and each agent can be matched to at most one other agent. In contrast, our framework allows each agent to be matched with multiple nodes, depending on their capacity—a scenario that is not applicable to kidney exchanges due to the nature of the problem. (ii) In KEP, patients have binary preferences—they are indifferent to which kidney they receive as long as it is compatible. However, in our problem, tasks have varying values to agents, meaning agents are not indifferent to the tasks they are assigned. Lastly, it is worth noting that the $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$ mechanisms can be viewed as specific priority mechanisms, which have been studied in the context of KEP [45]. Although these mechanisms are well-known, their effectiveness in the Maximum Vertex-weighted b-Matching (MVbM) problem has not been previously emphasized, especially in terms of achieving optimal outcomes with respect to the Price of Anarchy (PoA) and Price of Stability (PoS).

Agent Manipulation (Edge Manipulation/Edge and Capacity Manipulation)				
	Truthful	Group SP	Optimal	Efficiency
$\mathbb{M}_{BFS}$	No	No	Yes	PoA = PoS = 2
$\mathbb{M}_{DFS}$	No	No	Yes	PoA = PoS = 2
$\mathbb{M}_{G}$	Yes	Yes*	No	approx. ratio = 2
Task Manipulation (Edge Manipulation/Edge and Value Manipulation)				
	Truthful	Group SP	Optimal	Efficiency
$\mathbb{M}_{BFS}$	Yes	No	Yes	approx. ratio = 1
$\mathbb{M}_{DFS}$	Yes	No	Yes	approx. ratio = 1
$\mathbb{M}_{G}$	Yes	No	No	approx. ratio = 2

Table 1. The mechanisms we study and their properties in the two game-theoretical settings we consider. The "Yes*" indicates that the mechanism is group strategyproof under minor assumptions.

Agents' Manipulations – The properties of $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$. First, we study the case in which the agents are able to hide part of the edges connecting them to tasks and show that any mechanism that returns an MVbM cannot be truthful. In particular, we infer that both $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$ are not truthful. We then show that the first agent has a strategic advantage over the other agents and that the individual agent's ability to alter the outcome of the mechanisms depends on the order in which the mechanism processes the agents' set. Due to this correlation, we are able to describe the Nash Equilibria induced by the mechanisms and prove that the Price of Anarchy (PoA) and Price of Stability (PoS) of both $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$ are equal to 2. We then proceed to show that 2 is the best possible PoA and PoS achievable by deterministic mechanisms, thus $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$ are the best possible mechanisms in terms of PoA and PoS. Lastly, we extend our study to the case in which the agents report both the edges connecting them to the set of tasks and their capacity.

Agents' Manipulations – The properties of $\mathbb{M}_{G}$. Although $\mathbb{M}_{G}$ is not optimal, we show that $\mathbb{M}_{G}$ is truthful for agents' manipulations regardless of whether the agents can report only the edges or the edges and the capacity. Furthermore, it is group strategyproof whenever all the tasks have different values. We show that its approximation ratio is 2 and show that this bound is tight. Thus, there does not exist a deterministic truthful mechanism that has a lower approximation ratio. Moreover, we prove that the output produced by the approximation mechanism is actually the matching returned by $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$ one of their worst Nash Equilibrium.

Tasks' Manipulations. We then extend our study on the properties of the mechanisms when the tasks are allowed to act strategically. In this case, the set of incident edges and/or the values are the tasks' private information. In this simpler framework, we have that all of the three mechanisms we considered are truthful, albeit not group strategyproof.

We summarize the results we find of these mechanisms for both the Tasks and Agents Manipulation cases in Table 1.

Empirical Studies. In Section 5, we empirically compare the manipulability of $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$ over randomly generated instances. In particular, we observe that: (i) $\mathbb{M}_{BFS}$ is less prone to be manipulated by the first agent than $\mathbb{M}_{DFS}$. Moreover, the number of instances in which the first agent is able to manipulate $\mathbb{M}_{BFS}$ decreases as the number of agents increases, while the same does not hold for $\mathbb{M}_{DFS}$, and (ii) $\mathbb{M}_{BFS}$ is more susceptible to manipulations when the total capacity of the agents is equal to the number of tasks we can allocate.

2 PRELIMINARIES

In this section, we recall the basic notions and notations on the MVbM problem. Then, we describe the algorithm that defines the mechanisms we study.

The Maximum Vertex-weighted b-Matching Problem. Let $G = (A \cup T, E)$ be a bipartite graph. Throughout the paper we refer to $A = \{a_1, a_2, \ldots, a_n\}$ as the set of agents and refer to $T = \{t_1, t_2, \ldots, t_m\}$ as the set of tasks. In what follows, we assume that the agents' set A has an inner priority order. For the sake of simplicity, we assume that the agents' set is already ordered according to the priority order, that is a_i has an higher priority that a_j if $i < j$. The set E contains the edges of the bipartite graph. We say that an edge $e \in E$ belongs to agent a_i if $e = (a_i, t_j)$ for a $t_j \in T$. We denote with $M = |E|$ and $N = |A \cup T| = n + m$ the number of edges and the total number of vertices of the graph, respectively. Since the graphs are undirected, we denote an edge by (a_i, t_j) and (t_j, a_i) interchangeably. Moreover, for any given agent $a_i \in A$, we define the set $T_{E,i}$ as the set of tasks in T that are connected to agent a_i through the set of edges E. When it is clear from the context which set E we are referring to, we drop the subscript from $T_{E,i}$ and use T_i.

Let $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$ be the vector containing the capacities of the agents, where $b_i \in \mathbb{N}$ for every $i = 1, \ldots, n$. A subset $\mu \subset E$ is a b-matching if, for every vertex $a_i \in A$, the number of edges in μ linked to a_i is less than or equal to b_i and, for every vertex $t_j \in T$, the number of edges in μ linked to t_j is, at most, one. Given a b-matching μ, the vertex $a_i \in A$ is *saturated* with respect to μ if the number of edges in μ linked to a_i is exactly b_i. Otherwise, the vertex a_i is *unsaturated* with respect to μ. We denote with $\boldsymbol{q} = (q_1, q_2, \ldots, q_m)$ the vector containing the values of the tasks, in our framework, we have $q_j > 0$ for every j. The value of a matching μ is $w(\mu) \coloneqq \sum_{t_j \in T_\mu} q_j$, where $T_\mu \subset T$ is the set of tasks matched by μ. Given a bipartite graph and a vector $\boldsymbol{b}$, the MVbM problem consists in finding a b-matching μ that maximizes $w(\mu)$.

Finally, given two sets of edges μ_1 and μ_2, denote with $\mu_1 \oplus \mu_2 = (\mu_1 \backslash \mu_2) \cup (\mu_2 \backslash \mu_1)$ their *symmetric difference*. A path $P = \{(t_{j_1}, a_{i_1}), (a_{i_1}, t_{j_2}), (t_{j_2}, a_{i_2}), \ldots, (t_{j_L}, a_{i_L})\}$ in G is a sequence of edges that joins a sequence of vertices. We say that P has a length equal to R if it contains R edges. Given a path P and a b-matching μ, P is an *augmenting path* with respect to μ if every vertex a_{i_ℓ}, for $\ell = 1, \ldots, L-1$, is saturated with respect to μ, $a_{i_L} \in A$ is unsaturated with respect to μ, and the edges in the path alternatively do not belong to μ and belong to μ. That is, $(t_{j_\ell}, a_{i_\ell}) \notin \mu, \ell \in [L]$ and $(a_{i_\ell}, t_{j_{\ell+1}}) \in \mu, \ell \in [L-1]$, where $[L]$ is the set containing the first L natural numbers, i.e. $[L] \coloneqq \{1, 2, \ldots, L\}$.

The Algorithm Outline. In this paper, we consider two well-known algorithms from a mechanism design perspective. The first algorithm is presented in [47] and its routine consists in defining a sequence of matchings, namely $\{\mu_j\}_{j=0,1,\ldots,m}$, of increasing cardinality. Indeed, the first matching is set as $\mu_0 = \emptyset$ and, given μ_{j-1}, μ_j is defined as follows: if there exists P_j an augmenting path (with respect to μ_{j-1}) that starts from t_j, then $\mu_j = \mu_{j-1} \oplus P_j$, otherwise $\mu_j = \mu_{j-1}$. In Algorithm 1, we sketch the routine of the algorithm.

In [47] it has been shown that Algorithm 1 returns a solution to the MVbM problem in $O(NM)$ time, regardless of whether we find the augmenting path using the *Breadth-First Search* (BFS) or the *Depth-First Search* (DFS). Both the DFS and the BFS when they traverse the graph in search of an augmenting path implicitly assume that there is an ordering of the agents. We say that agent a_i has a higher *priority* than agent a_j if a_i is explored before a_j, so that agent a_1 has the highest priority. We say that Algorithm 1 is endowed with the BFS if we use the BFS to find an augmenting path. Similarly, we say that the algorithm is endowed with the DFS if we use the DFS to find an augmenting path.

The second algorithm is a greedy approximation version of Algorithm 1 that searches only among the augmenting paths whose length is 1. This greedy approximation mechanism has been previously introduced by Dobrian et al. [20] and subsequentally studied by Al-Herz and Pothen [6]. The authors showed that this greedy approximation algorithm finds a matching whose weight is, in the worst case, half the weight of the MVbM. Notice that for this greedy approximation algorithm, using the BFS or the DFS does not change the outcome thus we omit which traversing graph method is used.

Algorithm 1 Algorithm for MVbM [47]

Input: A bipartite graph $G = (A \cup T, E)$; agents' capacities $b_i, i = 1, \ldots, n$; task weights $q_j, j = 1, \ldots, m$; a traversing graph method that returns an augmenting path $\mathcal{R}$
Output: An MVbM μ.

$\mu_0 \leftarrow \emptyset$;
Relabel q_j $j = 1, \ldots, m$, in an non-increasing order;
for each $j \in T$ **do**
  if $\mathcal{R}$ finds an augmenting path P starting from j with respect to μ_{j-1} **then**
    $\mu_j = \mu_{j-1} \oplus P$;
  else
    $\mu_j = \mu_{j-1}$;
  end if
end for
return $\mu = \mu_m$;

3 THE MVBM PROBLEM WITH STRATEGIC AGENTS

In this section, we introduce and study a game-theoretical framework for the MVbM problem. We focus on the case in which the agents are strategic entities able to hide part of their connections to T and/or part of their own capacity.

3.1 The Game-Theoretical framework for agents' manipulation

Let us now assume that every agent in A reports its private information to a mechanism that, after gathering all the reports, returns a b-matching. In this subsection, we formally define the space of the agents' strategies and the mechanisms we are studying.

The Strategy Space of the Agents. Given a bipartite graph $G = (A \cup T, E)$, a capacity vector $\boldsymbol{b}$, a value vector $\boldsymbol{q}$, and a b-matching μ over G, we define the social welfare achieved by μ as the total value of the tasks assigned to the agents. Since the social welfare achieved by μ is equal to the total weight of the matching, we use $w(\mu)$ to denote it. We then define the utility of agent a_i as

$$w_i(\mu) := \sum_{t_j \in T_{\mu,i}} q_j,$$

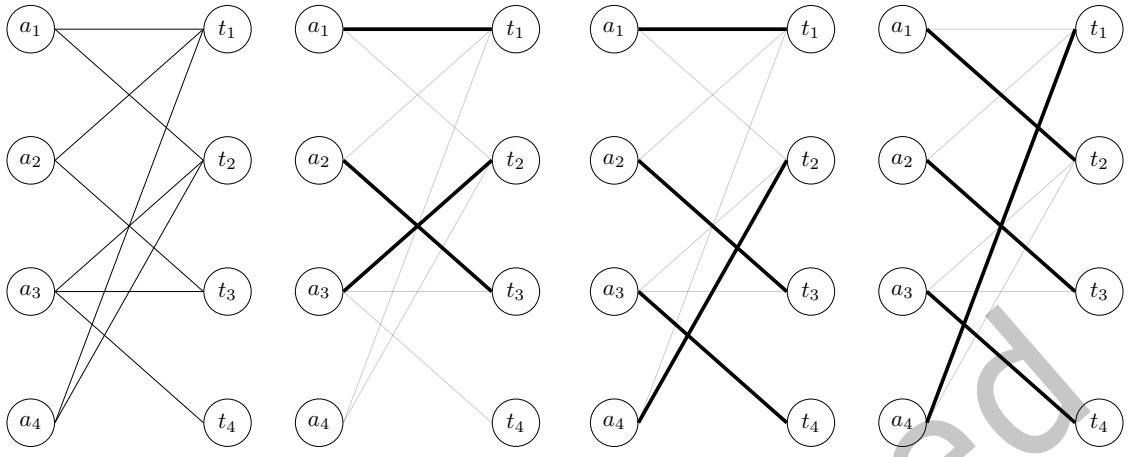
where $T_{\mu,i}$ is the set of tasks that μ matches with agent a_i. Notice that $w(\mu) = \sum_{i=1}^{n} w_i(\mu)$.

In this section, we consider two settings depending on what the private information of the agents is:

(i) The private information of each agent consists of the set of edges that connect the agent to the tasks' set. We call this setting Edge Manipulation Setting (EMS).
(ii) The private information of each agent consists of the set of edges and its own capacity. We call this setting Edge and Capacity Manipulation Setting (ECMS).

In both cases, we assume that every agent is bounded by its statement, thus every agent is able to report incomplete information, but they are not allowed to report false information.

In EMS, this means that an agent can hide some of the edges that connect it to the tasks, but it cannot report an edge that does not exist. The set of strategies of agent a_i, namely $\mathcal{S}_i$, is, therefore, the set of all the possible non-empty subsets of T_i, where T_i is the set containing all the existing edges that connect a_i to the tasks.



(a) The truthful input. (b) The output of $\mathbb{M}_G$. (c) The output of $\mathbb{M}_{BFS}$. (d) The output of $\mathbb{M}_{DFS}$.

Fig. 1. A truthful input and the different outputs provided by $\mathbb{M}_{BFS}$, $\mathbb{M}_{DFS}$, and $\mathbb{M}_G$. The thicker edges are the one composing the output of the considered mechanism.

In ECMS, this means that an agent can report only a capacity that is lower than its real one and that it cannot report an edge that does not exist. In this case, the set of strategies of agent a_i is then $\mathcal{S}_i \times [b_i]$, where b_i is the real capacity of the agent and $[b_i] := \{1, \dots, b_i\}$.

For the sake of simplicity, **from now on all the results and definitions we introduce are for the EMS**, unless we specify otherwise. We generalize our results to the ECMS in subsection 3.5.

The Mechanisms. A mechanism for the MVbM problem is a function $\mathbb{M}$ that takes as input the private information of the agents and returns a $\boldsymbol{b}$-matching. We denote with $\mathcal{I}_{\mathbb{M}}$ the set of possible inputs for $\mathbb{M}$ in the EMS. In our paper, we consider three mechanisms: (i) $\mathbb{M}_{BFS}$, which takes in input the edges of the agents and uses Algorithm 1 endowed with the BFS to select a matching[3]. (ii) $\mathbb{M}_{DFS}$, which takes in input the edges of the agents and uses Algorithm 1 endowed with the DFS to select a matching. (iii) $\mathbb{M}_G$, which takes in input the edges of the agents and uses a greedy version of Algorithm 1 to select a matching. Notice that $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$ are optimal since they are both induced by Algorithm 1.

Given a mechanism $\mathbb{M}$, every element $I \in \mathcal{I}_{\mathbb{M}}$ is composed by the reports of n self-interested agents, so that $\mathcal{I}_{\mathbb{M}} = \otimes_{i=1}^{n} \mathcal{I}_i$, where $\mathcal{I}_i$ is the set of feasible inputs for agent a_i. We say that a mechanism $\mathbb{M}$ is *strategyproof* (or, equivalently, *truthful*) for the EMS if no agent can get a higher payoff by hiding edges. More formally, if I_i is the true type of agent a_i, it holds true that

$$w_i(\mathbb{M}(I_i', I_{-i})) \leq w_i(\mathbb{M}(I_i, I_{-i}))$$

for every $I_i' \in \mathcal{S}_i$. Another important property for mechanisms is the group strategyproofness. A mechanism is *group strategyproof* for agent manipulations if no group of agents can collude to hide some of their edges in such a way that (i) the utility obtained by every agent of the group after hiding the edges is greater than or equal to the one they get by reporting truthfully and (ii) at least one agent gets a better payoff after the group hides the edges.

To evaluate the performances of the mechanisms, we use the Price of Anarchy (PoA), Price of Stability (PoS), and approximation ratio (ar), which we briefly recall in the following.

[3]Throughout the paper, we use $\mathbb{M}_{BFS}$ to denote both the mechanism that takes as input the edges and both the edges and the capacity of every agent. It will be clear from the context which is the input of the mechanism. The same goes for the other two mechanisms.

Price of Anarchy. The Price of Anarchy (PoA) of mechanism $\mathbb{M}$ is defined as the maximum ratio between the optimal social welfare and the welfare in the worst Nash Equilibrium, hence

$$PoA(\mathbb{M}) := \sup_{I \in \mathcal{I}} \frac{w(\mu(I))}{w(\mu_{wNE}(I))},$$

where $\mu_{wNE}(I)$ is the output of $\mathbb{M}$ when the agents act according to the worst Nash Equilibrium, i.e. the Nash Equilibrium that achieves the worst social welfare.

Price of Stability. The Price of Stability (PoS) of a mechanism $\mathbb{M}$ is defined as the maximum ratio between the optimal social welfare and the welfare in the best Nash Equilibrium, hence

$$PoS(\mathbb{M}) := \sup_{I \in \mathcal{I}} \frac{w(\mu(I))}{w(\mu_{bNE}(I))},$$

where $\mu_{bNE}(I)$ is the output of $\mathbb{M}$ when the agents act according to the best Nash Equilibrium, i.e. the Nash Equilibrium that achieves the maximum social welfare. Notice that, by definition, we have $PoS(\mathbb{M}) \leq PoA(\mathbb{M})$.

Approximation Ratio. The approximation ratio of a truthful mechanism $\mathbb{M}$ is defined as the maximum ratio between the optimal social welfare and the welfare returned by $\mathbb{M}$. Hence, we have

$$ar(\mathbb{M}) := \sup_{I \in \mathcal{I}} \frac{w(\mu(I))}{w(\mu_{\mathbb{M}}(I))},$$

where $\mu_{\mathbb{M}}(I)$ is the output of $\mathbb{M}$ when the input is I.

3.2 The Truthfulness of the Mechanisms

In this section, we study the truthfulness of the three mechanisms induced by Algorithm 1 and its approximation version. We show that, although $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$ are optimal, they are not truthful due to an impossibility result. Furthermore, we show that the manipulability of a mechanism is related to the length of the augmenting paths found during the routine of Algorithm 1 and use this characterization to prove that $\mathbb{M}_G$ is truthful.

Theorem 1. *There is no deterministic truthful mechanism that always returns an MVbM for agent manipulation.*

Proof. We show this using a counterexample. Consider two agents a_1 and a_2 and three tasks t_1, t_2, t_3. The edge set is $E = \{(a_1, t_1), (a_1, t_2), (a_2, t_1), (a_2, t_3)\}$. The values of the three tasks are $q_1 = 1, q_2 = 0.1$, and $q_3 = 0.1$, respectively, while the capacity of both agents is $b_1 = b_2 = 1$. It is easy to see that the optimal matching is not unique. In particular, both $\{(a_1, t_1), (a_2, t_3)\}$ and $\{(a_1, t_2), (a_2, t_1)\}$ are feasible solutions whose total weight is 1.1. Let us assume that the mechanism returns $\{(a_1, t_1), (a_2, t_3)\}$. In this case, if agent a_2 does not report edge (a_2, t_3), the maximum matching becomes $\{(a_1, t_2), (a_2, t_1)\}$. According to this, agent a_2's utility increases from 0.1 to 1. Similarly, if the mechanism returns $\{(a_1, t_2), (a_2, t_1)\}$, agent a_1 can manipulate the result by hiding the edge (a_1, t_2). Therefore, there is no deterministic truthful mechanism that always returns a maximum matching. □

From Theorem 1, we infer that both $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$ are not truthful and, thus, not group strategyproof. In the following, we characterize a sufficient condition under which agents' best strategy is to report truthfully.

Lemma 1. *Let us consider an instance in which Algorithm 1 (endowed with either the DFS or the BFS) completes its routine using only augmenting paths of length equal to* 1*. Then, an agent can increase its utility by hiding its edges if and only if in doing so it forces the algorithm to use an augmenting path with length greater than* 1 *during the cycle at line* 3*.*

Proof. Let us denote with μ_k the matching found at the k-th step. By hypothesis, Algorithm 1 terminates by using only augmenting paths of length equal to 1, thus the sequence of matching that it finds is monotone increasing. That is, $\mu_k \subset \mu_{k+1}$ for every k. Since the sequence $\{\mu_k\}_k$ is increasing, it means that the algorithm

is able to define μ_{k+1} by simply adding an edge connected to t^{k+1} to μ_k. Let us assume that an agent hides a set of edges and that the matching sequence found by the algorithm is still monotone increasing. Since the sequence of matching found in the manipulated instance is monotone increasing, it means that the final matching is obtained by adding an edge at every iteration. By the definition of $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$ and since the length of the augmenting path is 1, the edge added at step j, is taken from the ones that are connected to the task t_j. Therefore, the agent manipulating cannot benefit from the manipulation, since it is only reducing the set of tasks that it can receive. □

Since $\mathbb{M}_G$ uses only augmenting paths of length at most equal to 1, regardless of the input, Lemma 1 allows us to conclude that $\mathbb{M}_G$ is truthful. Moreover, due to the results proven by Dobrian et al. [20], we are able to characterize the approximation ration of $\mathbb{M}_G$.

Theorem 2. *The mechanism $\mathbb{M}_G$ is truthful with respect to agent manipulation. Moreover, its approximation ratio is 2.*

Proof. The first part of the statement follows directly from Lemma 1. The second part of the statement has been proven by Dobrian et al. [20]. Indeed, Dobrian et al. have shown that the weight of the matching returned by the approximation algorithm that uses augmenting paths of length equal to 1 is, in the worst case, half of the maximal weight. In their paper, Dobrian et al. study the case in which both sides of the bipartite graph have a non-negative value. However, if we set the value of each agent in A to be 0, we have that the weight of the matching studied in [20] is the same as the social welfare (from the agents' viewpoint) we defined in the present paper. Similarly, the maximum weight of the matching is the maximum social welfare achievable. We therefore conclude that the ratio between the social welfare returned by $\mathbb{M}_G$ and the maximum one is greater than 0.5. Since the approximation ratio is the ratio between the maximum social welfare and the one achieved by $\mathbb{M}_G$, we infer that it has to be smaller than 2. To prove that the approximation ratio of the mechanism is 2, consider the following instances. There are two agents, namely a_1 and a_2, and two tasks t_1 and t_2, whose values are $1+\epsilon$ and 1, respectively, where $\epsilon > 0$. Let $E = \{(a_1, t_1), (a_1, t_2), (a_2, t_1)\}$ be the truthful set of edges. It is easy to see that $\mathbb{M}_G$ returns the matching $\mu_{\mathbb{M}_G} = \{(a_1, t_1)\}$, while the optimal one is $\{(a_2, t_1), (a_1, t_2)\}$. Hence the ratio between the maximum social welfare and the social welfare achieved by $\mathbb{M}_G$ on this instance is $\frac{2+\epsilon}{1+\epsilon}$. By taking the limit for $\epsilon \to 0$, we conclude the proof. □

The approximation ratio achieved by $\mathbb{M}_G$ is actually the best possible ratio achievable, as the next result shows.

Theorem 3. *There does not exist a truthful and deterministic mechanism for the MVbM problem that achieves an approximation ratio better than 2.*

Proof. Toward a contradiction, assume that there exists a mechanism $\mathbb{M}$ whose approximation ratio is equal to $2-\delta$, where δ is a positive value. Let us now consider the following instance. We have two agents, namely a_1 and a_2, and two tasks, namely t_1 and t_2. The capacity of both agents is set to be equal to 1. The value q_1 of t_1 is $1+\epsilon$ and the value q_2 of t_2 is equal to 1. Finally, we assume that, according to their truthful inputs, both agents are connected to both tasks, so that $E = \{(a_1, t_1), (a_1, t_2), (a_2, t_1), (a_2, t_2)\}$. It is easy to see that the maximum value that a matching can achieve is $2+\epsilon$. If the mechanism $\mathbb{M}$ does not allocate both the tasks, we have that the value achieved by the mechanism is, at most $1+\epsilon$. Therefore, we have that the approximated ratio of $\mathbb{M}$ is at least equal to $\frac{2+\epsilon}{1+\epsilon} = 1 + \frac{1}{1+\epsilon}$. If we take $\epsilon < \frac{\delta}{1-\delta}$, we get that the approximated ratio of $\mathbb{M}$ should be greater than $2-\delta$, which is a contradiction. Hence $\mathbb{M}$ allocates both the tasks in the previously described instance. Without loss of generality, let us assume that $\mathbb{M}$ allocates t_1 to a_2 and t_2 to a_1. Let us now consider the instance in which a_1 is not connected with the task t_2, so that $E' = \{(a_1, t_1), (a_2, t_1), (a_2, t_2)\}$. The maximal value that a matching can achieve is still $2+\epsilon$. However, since the mechanism is truthful, the agent a_1 cannot receive any task. Indeed, the only task that $\mathbb{M}$ can assign to a_1 is t_1, however, if $\mathbb{M}$ assigns t_1 to a_1, it means that agent a_1 can manipulate $\mathbb{M}$ by reporting

the set of edges $\{(a_1, t_1)\}$ over $\{(a_1, t_1), (a_1, t_2)\}$ in the instance when the truthful input is E. Since the first agent cannot receive any tasks from $\mathbb{M}$, we have that the maximum matching value achieved by $\mathbb{M}$ when the input is E' is, at most, $1 + \epsilon$, so that the approximation ratio of $\mathbb{M}$ is, at least $\frac{2+\epsilon}{1+\epsilon}$. Again, by taking $\epsilon < \frac{\delta}{1-\delta}$, we conclude a contradiction. □

From Theorem 3, we then infer that $\mathbb{M}_G$ is the best possible deterministic and truthful mechanism for our game theoretical setting. To conclude, we show that, $\mathbb{M}_G$ is also group strategyproof if all the tasks have different values.

Theorem 4. *If all the tasks in T have different values, then $\mathbb{M}_G$ is group strategyproof.*

Proof. Toward a contradiction, let us assume that there exists a coalition of agents $C = \{a_{i_1}, \ldots, a_{i_\ell}\}$ that is able to collude. Since hiding edges that are not returned by $\mathbb{M}_G$ does not alter the outcome of the mechanism, we assume, without loss of generality, that at least agent a_{i_1}, hides one of the edges that are in the matching found by $\mathbb{M}_G$ when all the agents report truthfully. Let us denote with t_l the task connected to a_{i_1} through the hidden edge. After misreporting agent a_{i_1} cannot be allocated with t_l. Furthermore, due to the routine of $\mathbb{M}_G$, each task is allocated independently from the others, hence a_{i_1} cannot be allocated with a better task. Finally, since there are no tasks with the same value, a_{i_1}'s payoff is necessarily lowered by misreporting, even if in a coalition, which is a contradiction. We therefore conclude the proof. □

The condition of Theorem 4 are tight. Indeed, as we show in the next example, even if just two tasks have the same value, the mechanism is no longer group strategyproof.

Example 1. *Let us consider the following instance. The set of agents is composed of three elements, namely a_1, a_2, and a_3. The capacity of each agent is set to 1, so that $b_1 = b_2 = b_3 = 1$. The set of tasks is composed of two elements, namely t_1 and t_2. The value of both tasks is equal to 1. Finally, let $E = \{(a_1, t_1), (a_1, t_2), (a_2, t_2), (a_3, t_1)\}$ be the truthful input. Then, $\mathbb{M}_G(E) = \{(a_1, t_1), (a_2, t_2)\}$. However a_1 and a_3 can collude: if agent a_1 hides the edge (a_1, t_1), the $\mathbb{M}_G$ returns $\mu' = \{(a_1, t_2), (a_3, t_1)\}$.*

3.3 Strategy Characterization and Worst-case Equilibrium

In this subsection, we study the Nash Equilibria induced by the mechanisms $\mathbb{M}_{BFS}$ or $\mathbb{M}_{DFS}$. First, we show that an agent who is not matched to any task when reporting truthfully cannot improve its utility by manipulation.

Lemma 2. *Given a truthful input, if an agent receives a null utility from $\mathbb{M}_{BFS}$, then its utility cannot be improved by hiding edges. The same holds for the mechanism $\mathbb{M}_{DFS}$.*

Proof. For the sake of simplicity, we present the proof only for the mechanism $\mathbb{M}_{BFS}$. Let us denote with G the truthful graph and with G' the graph manipulated by agent a_k. Let μ and μ' be the matching returned by $\mathbb{M}_{BFS}$ when G and G' are given as input, respectively. By contradiction, suppose $\mu' \neq \mu$. In this case, there exists t_j such that $P_j \neq P'_j$ and $P_r = P'_r$ for any $r < j$, where P_l and P'_l are the augmenting paths returned by the BFS at the l-th step. Since $G' \subset G$ and $\mu_{l-1} = \mu'_{l-1}$, we must have that the last edge of P_l is in G and not in G'. However, since G' is only missing edges connected to a_k, we must have that P_l contains an edge that connects agent a_k to a task. In particular, at the j-th step, the augmenting path returned by the BFS contains an edge connected to a_k, which is not possible since a_k is left unmatched by μ. □

For this reason, starting now, we make the assumption that if an agent can manipulate either $\mathbb{M}_{BFS}$ or $\mathbb{M}_{DFS}$, then that agent is assigned at least one task when reporting truthfully.

We now demonstrate that the first agent processed by either $\mathbb{M}_{BFS}$ or $\mathbb{M}_{DFS}$, denoted as a_1, is always capable of obtaining its highest possible payoff by misreporting.

Theorem 5. *For both $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$, agent a_1's best strategy is to report only the top b_1-valued tasks to which it is connected.*

Proof. Let us denote with $t_{j_1}, t_{j_2}, \ldots, t_{j_{b_1}}$ the top b_1 tasks agent a_1 is connected to. For every t_{j_r}, at the j_r-th step of Algorithm 1, agent a_1 will not be saturated; therefore the path $P_{j_r} = \{(t_{j_r}, a_1)\}$ is augmenting with respect to matching μ_{j_r-1}. Moreover, since the BFS searches among the vertices in lexicographical order, the path P_{j_r} will always be the first one being explored and, since it is augmenting, it will be the one returned. To conclude, we notice that, after the j_r-th iteration, there are no augmenting paths that pass by a_1 as all the edges connected to a_1 are already in the matching, hence the set of tasks allocated to a_1 will not change in later iterations of the algorithm.

By a similar argument, we get to the same conclusion for $\mathbb{M}_{DFS}$. □

In the next example, we show that the advantage described in Theorem 5 is only due to the fact that agent a_1 is self-aware of its position in the processing process. Indeed, if the same strategy is played by an agent that has the same truthful input but a different processing priority, the outcome of the manipulation might be detrimental to the agent.

Example 2. *Let us consider the following instance. There are 3 agents, namely α, β, and γ whose capacities are $b_\alpha = 2$ and $b_\beta = b_\gamma = 1$. Let us consider a set of 4 tasks, namely t_j with $j \in [4]$ whose respective values are $q_j = 2^{-j}$. In the truthful input, agent α is connected to all 4 tasks, while agent β is connected only to task t_1 and agent γ is connected only to task t_2. Let us consider the mechanism $\mathbb{M}_{BFS}$: the maximum matching for the truthful input allocates t_1 to agent β, t_2 to agent γ, and the other two tasks to agent α. Let us now assume that the processing order of the agents is α, β, and γ. That is, $a_1 = \alpha$, $a_2 = \beta$, and $a_3 = \gamma$. Then, if agent α applies the strategy highlighted in Theorem 5 and reports only the first two edges, it improves its utility from $q_3 + q_4$ to $q_1 + q_2$. However, in a different order of agents in which agent α is the second, i.e., $a_2 = \alpha$, if it applies the same strategy, then agent α gets only one of the two tasks (depending on who is the agent going first) while if it goes as the third, it receives no tasks. We also note that, if agent α is the second, its best strategy is to report the edges connecting it to tasks t_1 and t_3 if agent γ goes first or tasks t_2 and t_3 if β goes first. Hence, the priority of agent α and the priority of the other two agents determines what the best strategy for agent α is. Similarly, the same conclusion can be drawn for the mechanism $\mathbb{M}_{DFS}$.*

If the agents' order is changed and the same agent does not have the highest priority, then its best strategy depends on its priority and the agents before it. Indeed, the agents' order determines the Nash Equilibrium of both $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$. When all the tasks have different values, we define the sets $T^{(i)}$ and $B^{(i)}$ for $i = 0, 1, \ldots, n$, as follows:

(i) $T^{(0)} = T$ where T is the set of all the tasks and $B^{(0)} = \emptyset$;
(ii) $T^{(i)} = T^{(i-1)} \setminus B_i^{(i-1)}$, where $B_i^{(i-1)}$ is the set containing the top b_i-valued tasks among the ones in $T^{(i-1)}$ that agent i is connected to. If there are less than b_i tasks among the ones that the agent can take, it takes all the tasks in $T^{(i-1)}$ that it is connected to.

If two or more tasks have equal value, we need to fix a tie breaking rule before defining the the sets $T^{(i)}$ and $B^{(i)}$. Notice that both $T^{(i)}$ and $B^{(i)}$ may depend on the tie breaking rule.

We then define the First-Come-First-Served policy (FCFS policy) of agent a_i as $FCFSP_i = \{(a_i, t_j)\}_{t_j \in B_i^{(i-1)}}$. Notice that $FCFSP_i \in \mathcal{S}_i$ for every $i \in [n]$, so that it is a feasible strategy for every agent.

Theorem 6. *Given an MVbM problem such that $q_i \neq q_j$ for every $i, \neq j \in [m]$, the FCFS policies constitute a Nash Equilibrium that achieves the lowest social welfare in both mechanisms $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$. Moreover, we have*

$$\mathbb{M}_{BFS}(\cup_{a_i \in A} FCFSP_i) = \mathbb{M}_{DFS}(\cup_{a_i \in A} FCFSP_i) = \cup_{a_i \in A} FCFSP_i.$$

Proof. We prove the first part of the theorem in two steps. First, we prove that the FCFS policies constitute a Nash Equilibrium. Second, we show that the Nash Equilibrium they form is the one with the lowest possible social welfare.

Let us then consider agent a_i and, toward a contradiction, let us assume that, when all the other agents are using their FCFS policies, reporting the set of edges S_{a_i} gives a_i a bigger payoff than what it would get from reporting $FCFSP_i$. Let us set $s_i = \min_{(a_i,t_j)\in FCFSP_i} q_j$. Let us assume that S_{a_i} contains an edge that connects a_i to a task that has a higher value than s_i, namely t_l. We now show that, since the other agents are applying their FCFS policies, the task t_l is allocated to an agent with higher priority unless the edge already belongs to $FCFSP_i$. Indeed, since $|FCFSP_j| \leq b_j$ for every $a_j \in A$, there cannot be augmenting paths that pass by any of the agents playing their FCFS policy. Indeed, since the union of the FCFS policies is a b-matching, either $(a_i, t_l) \in FCFSP_i$ or there exists another agent whose FCFS policy connects it to t_l. If t_l is connected to an agent a_k, $(a_k, t_l) \in FCFSP_k$, and agent a_k's priority is higher than agent a_i's priority, the final output of the mechanism assigns t_l to a_k. We can then assume that S_{a_i} does not contain edges that connect agent a_i to tasks with values higher than s_i and that do not belong to $FCFSP_i$. To conclude, we notice that if S_{a_i} contains an edge connecting agent a_i to a task that has a value lower than s_i, then the payoff of agent a_i can only be lowered. Indeed, if $|FCFSP_i| = b_i$, then a_i cannot improve its payoff by reporting edges that connect a_i to tasks that have a value lower than s_i. This follows from the fact that all the other players are using their FCFS policies and therefore agent a_i is allocated the set $B_i^{(i-1)} = \{t_j \in T \text{ s.t. } (a_i, t_j) \in FCFSP_i\}$ if it uses its FCFS policy. If $|FCFSP_i| < b_i$, by definition, it means that there are no tasks that agent a_i can be connected to and that have a value lower than s_i. Therefore S_{a_i} does not contain edges connecting a_i to a task with a value lower than s_i nor edges connecting it with tasks that have a value greater than s_i and that are not included in $FCFSP_i$. Since reporting a subset of $FCFSP_i$ would result in a lower payoff, we deduce that $S_{a_i} = FCFSP_i$, which is a contradiction since we assumed $FCFSP_i \neq S_{a_i}$.

We now prove that the Nash Equilibrium induced by the FCFS policies is one of the worst equilibria. Toward a contradiction, let us consider another set of strategies, namely $\{S_{a_i}\}_{a_i\in A}$, such that the social welfare achieved by this equilibrium is strictly lower than the one obtained if every agent uses its FCFS policy. By the definition of social welfare, there must exist at least one agent that, according to the equilibrium defined by $\{S_{a_i}\}_{a_i\in A}$, receives a payoff that is strictly lower than the one it would obtain by using its FCFS policy. Let us denote with a_k the first agent that, according to the priority order of the mechanism, receives a lower value. Agent a_k cannot be the first agent, as it otherwise could apply its FCFS policy and get a better payoff. Then, agent a_k is among the remaining agents and it is not getting any of the tasks that are given to the first agent according to its FCFS policy, as otherwise, agent a_1 could increase its payoff by manipulating and the set of strategies $\{S_i\}_{a_i\in A}$ would not be a Nash Equilibrium. From a similar argument, we infer that agent a_k cannot be the second agent, as otherwise, it could get a better payoff by using its FCFS policy. Moreover, the second agent is allocated the tasks that are granted to it from its FCFS policy. Both of which we have already proved cannot be. By applying the same argument to the other agents, we get a contradiction, since no agent can be agent a_k. We, therefore, conclude that the set of strategies given by the FCFS policies is one of the worst Nash Equilibrium.

The last part of the Theorem follows from the fact that $\cup_{a_i\in A} FCFSP_i$ is itself a b-matching, hence it is an $MVbM$ for the edge set $E = \cup_{a_i\in A} FCFSP_i$. □

Following the same argument presented in the proof of Theorem 6, we infer that any set of strategies $\{S_i\}_{a_i\in A}$ for which it holds $FCFSP_i \subset S_i$ and $\min_{t_j,\,(a_i,t_j)\in FCFSP_i} q_j = \min_{t_j,\,(a_i,t_j)\in S_i} q_j$ for every i, defines a Nash Equilibrium. Among this class of Nash Equilibria, $\{FCFSP_i\}_{a_i\in A}$ is the only one for which it holds

$$\mathbb{M}_{BFS}(\cup_{a_i\in A} FCFSP_i) = \mathbb{M}_{DFS}(\cup_{a_i\in A} FCFSP_i) = \cup_{a_i\in A} FCFSP_i.$$

Moreover, this equilibrium achieves the same social welfare of the matching returned by $\mathbb{M}_G$.

Theorem 7. *Given an MVbM problem, let μ_E be the matching returned by $\mathbb{M}_G$. Then, it holds that $\mu_E = \cup_{a_i \in A} FCFSP_i$. That is, the output of $\mathbb{M}_G$ on any given instance is equal to the union of the agents' FCFS policies. Hence, the social welfare achieved by $\mathbb{M}_G$ is equal to the social welfare achieved by $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$ in one of their worst Nash Equilibria.*

Proof. We prove this Theorem by induction. First, we prove that $\mathbb{M}_G$ allocates a_1 with the set of tasks $FCFSP_1$. Second, we show that if all the agents $a_1, a_2, \ldots, a_k$ receive $FCFSP_1, FCFSP_2, \ldots, FCFSP_k$ from $\mathbb{M}_G$, then also agent a_{k+1} receives $FCFSP_{k+1}$.

Let us consider a_1. Since $\mathbb{M}_G$ checks the agents following their orders, a_1 is always the first agent checked. Hence, a task t_j is not allocated to a_1 if and only if there are b_1 other tasks that have a higher value than t_j and that are all connected to a_1. This means that the set of tasks allocated to a_1 is $FCFSP_1$.

Let us now assume that agents $a_1, a_2, \ldots, a_k$ are given the tasks contained in $FCFSP_1$, $FCFSP_2, \ldots, FCFSP_k$ respectively, and let us consider a_{k+1}. We have that $\mathbb{M}_G$ allocates a task t_j to a_{k+1} if, at step j of Algorithm 1, all the agents with priorities higher than a_{k+1} that are connected to t_j are already saturated and a_{k+1} is not saturated. However, by assumption, agents with a higher priority than a_{k+1} are getting the tasks that they would get from their FCFS policies. We, therefore, conclude that the tasks allocated to agent a_{k+1} from $\mathbb{M}_G$ consist of a subset of $T^{(k)}$. From an argument similar to the one used for agent a_1, we infer that a_{k+1} receives the top b_{k+1} higher valued tasks among the ones in $T^{(k)}$, which coincides with the set $FCFSP_{k+1}$. We conclude the proof by induction. □

Finally, we show that Theorem 7 along with Theorem 2 allows us to compute the PoA of both $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$.

Theorem 8. *The PoA of $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$ is equal to* 2.

Proof. From Theorem 7, we know that $\mathbb{M}_G$ returns a matching whose social welfare is equal to the social welfare of one of the worst Nash Equilibrium of $\mathbb{M}_{BFS}$. Since $\mathbb{M}_{BFS}$ returns an MVbM, we have

$$PoA(\mathbb{M}_{BFS}) = \sup_{I \in \mathcal{I}} \frac{w(\mu(I))}{w(\mu_{wNE}(I))} = \sup_{I \in \mathcal{I}} \frac{w(\mathbb{M}_{BFS}(I))}{w(\mathbb{M}_G(I))}. \tag{1}$$

Since the matching found by $\mathbb{M}_{BFS}$ achieves the maximum social welfare, the last term in equation (1) is bounded from above by the $a.r.(\mathbb{M}_G)$, so that $PoA(\mathbb{M}_{BFS}) \leq ar(\mathbb{M}_G) = 2$. To conclude $PoA(\mathbb{M}_{BFS}) = 2$ we show a lower bound of 2. The set of agents is composed of two agents, namely a_1 and a_2, we assume the agents to be ordered according to the algorithm priority. The capacity of both agents is equal to 1. The set of tasks contains two tasks, namely t_1 and t_2, whose values are $1 + \epsilon$ and 1, respectively. Finally, let us assume that the truthful input is given by $E = \{(a_1, t_1), (a_1, t_2), (a_2, t_1)\}$. The social welfare is then equal to $2 + \epsilon$. However, in the worst Nash Equilibrium, the welfare is $1 + \epsilon$. By taking the limit for $\epsilon \to 0$, we conclude that the PoA is equal to 2. By a similar argument, we infer $PoA(\mathbb{M}_{DFS}) = 2$. □

The previous bound is tight, indeed, there does not exist a deterministic mechanism for the *MVbM* problem that has a *PoA* lower than 2.

Theorem 9. *For every deterministic mechanism $\mathbb{M}$, we have $PoA(\mathbb{M}) \geq 2$ with respect to agent manipulations.*

Proof. Toward a contradiction, let $\mathbb{M}$ be a deterministic mechanism whose *PoA* is lower than 2. Let us consider the following instance. We have two tasks, namely t_1 and t_2, whose values are $1 + \epsilon$ and 1, respectively. We then have two agents, namely a_1 and a_2 and both have a capacity equal to 1. Let us now consider the instance in which both the agents are only connected to task t_1, hence the truthful input is $E = \{(a_1, t_1), (a_2, t_1)\}$.

Since $PoA(\mathbb{M})$ is finite, we have that $\mathbb{M}$ allocates t_1 to one agent. Indeed, if no agent receives a task, no one can improve its own payoff by hiding its only edge [4]. Hence, the truthful instance is already a Nash Equilibrium.

[4]we recall that every agent is bounded by its statement and that every agent has to report at least one edge according to how we defined the strategy set

Furthermore, since the social welfare of this Nash Equilibrium is 0, this is also one of the worst Nash Equilibria, thus we find a contradiction since we assumed that $\mathbb{M}$ has a finite PoA.

Let us then assume that one agent gets t_1. Without loss of generality, let us assume that $\mathbb{M}$ allocates t_1 to a_1, the other case is completely symmetric with respect to the one we are about to present.

Let us now consider the instance whose truthful input is $E = \{(a_1, t_1), (a_1, t_2), (a_2, t_1)\}$. If $\mathbb{M}$ allocates t_1 to a_2, we have that a Nash Equilibrium is obtained when agent a_1 hides arc (a_1, t_2). Indeed, if agent a_1 hides (a_1, t_2), the input of the mechanism is $E = \{(a_1, t_1), (a_2, t_1)\}$ which gives the first task to a_1. Since a_2 is bounded by its statements and its only alternative is to report no edges, it has no better strategy to play. Similarly, since a_1 is getting its best possible payoff, it has no better strategy to play. We then conclude that when a_1 hides the edge (a_1, t_2), we have a Nash Equilibrium. Finally, we observe that, by taking ϵ small enough, we get a contradiction with the assumption $PoA(\mathbb{M}) < 2$, since the maximum social welfare is $2 + \epsilon$, while the social welfare returned by the mechanism in the worst Nash Equilibrium is at most $1 + \epsilon$. Notice that there might be another Nash Equilibrium in which the social welfare is even lower, however, it suffice to notice that the social welfare achieved in the worst Nash Equilibrium is lower than $1 + \epsilon$ to conclude the proof.

Similarly, if the mechanism does not allocate t_1 to a_2, the instance is already a Nash Equilibrium. Indeed, by the same argument used before, a_2 cannot improve its own payoff, since it is getting no tasks. If a_1 is allocated with the task, it cannot improve its payoff either, since it is getting the maximum payoff it can get. Finally, if a_1 is not getting t_1, it can hide (a_1, t_2) and return to the instance we considered before. Again, by taking ϵ small enough, we retrieve that the $PoA(\mathbb{M})$ cannot be less than 2. □

We close the section by studying the PoS of $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$. We recall that the PoA of every mechanism is greater than its PoS, thus we infer that both $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$ have a PoS at most equal to 2. Moreover, since the Nash Equilibrium in the example we used in the proof of Theorem 9 is unique, the best and worst Nash Equilibria achieve the same social welfare. In particular, this allows us to prove that the PoS of both $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$ is equal to 2. Furthermore, this value is tight for the class of deterministic mechanisms.

Theorem 10. *The PoS of $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$ is equal to 2. Moreover, there does not exist a deterministic mechanism whose PoS is lower than 2.*

Proof. Since the Price of Anarchy of a mechanism is always greater than the Price of Stability, we have that $PoS(\mathbb{M}_{BFS}) \leq 2$. We now show that $PoS(\mathbb{M}_{BFS}) = 2$ by showing an instance on which the PoS is equal to 2.

Let us consider the example built in the proof of Theorem 8. Since there is only one Nash Equilibrium, it is both the worst and best Nash Equilibrium. Therefore, following the argument used in the proof of Theorem 8, we conclude $PoS(\mathbb{M}_{BFS}) \geq 2$, hence $PoS(\mathbb{M}_{BFS}) = 2$.

From a similar argument, we infer $PoS(\mathbb{M}_{DFS}) = 2$.

Finally, to prove that the PoS of every deterministic mechanism is greater than 2, consider the instance build in the proof of Theorem 9. Since the Nash Equilibrium of this instance is unique, we conclude that $PoS(\mathbb{M}) \geq 2$ for every deterministic and optimal mechanism. □

In particular, $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$ are the best possible mechanisms in terms of PoA and PoS, while $\mathbb{M}_G$ is the best possible truthful mechanism in terms of approximation ratio.

3.4 Agents Manipulating their Edges and Capacity

In this section, we extend our study on the truthfulness of $\mathbb{M}_{BFS}$, $\mathbb{M}_{DFS}$, and $\mathbb{M}_G$ to the ECMS, hence the agents self-report their capacity along with their edges.[5] As for the EMS, we assume the agents to be bounded by their

[5]With a slighty abuse of notation, we use $\mathbb{M}_{BFS}$, $\mathbb{M}_{DFS}$, and $\mathbb{M}_G$ to denote the mechanisms obtained from Algorithm 1 and its greedy version when applied to the ECMS case, respectively.

statements, thus they can manipulate only by hiding edges or by reporting a lower capacity than their real one. In this setting, a mechanism $\mathbb{M}$ is truthful if, for every $i \in [n]$, it holds $w_i((I'_i, b'_i), J_{-i}) \leq w_i((I_i, b_i), J_{-i})$, for every $(I'_i, b'_i) \in \mathcal{S}_i \times [b_i]$, where J_{-i} are the reports of the other agents. Once we fix the set of strategies of each agent, we can define the PoA, PoS, and approximation ratio of a mechanism $\mathbb{M}$ as for the EMS by carefully changing the set of strategies to fit the ECMS case.

As we show, neither the truthfulness nor the efficiency guarantees of $\mathbb{M}_{BFS}$, $\mathbb{M}_{DFS}$, and $\mathbb{M}_G$ change from EMS to ECMS. Furthermore, all the bounds are still tight.

Theorem 11. *In the ECMS, $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$ are both not truthful. The PoA and the PoS of both $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$ are equal to 2. Moreover, these bounds are tight, hence there is no other deterministic mechanism whose PoA or PoS is lower.*

Proof. First, we observe that, since the strategy set of every agent in the ECMS is larger than what it is in the EMS, the mechanisms cannot be truthful in the ECMS. Indeed, if the mechanisms are not truthful for the EMS, they cannot be truthful for the ECMS. To prove that $PoA(\mathbb{M}_{BFS}) = PoA(\mathbb{M}_{DFS}) = 2$ it suffices to show that the strategies $\{(FCFS_i, b_i)\}_{i\in[n]}$ forms a Nash Equilibrium for both $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$. Without loss of generality, we focus on $\mathbb{M}_{BFS}$, since the same conclusion can be drawn for $\mathbb{M}_{DFS}$.

Toward a contradiction, let us assume that there exists an agent, namely a_k, whose best strategy is to play (S_k, c_k) when all the other agents play $(FCFS_i, b_i)$. Since $c_k \leq b_k$, agent a_k gets at most the same number of tasks that it would get in the truthful case. Furthermore, since the tasks that have a higher value than the one a_k would get by playing $(FCFS_k, b_k)$ cannot be de-allocated from the agents already getting them (since every other player is playing $(FCFS_i, b_i)$), agent a_k is getting fewer tasks that have a lower value than the ones it would get by playing $(FCFS_k, b_k)$, which is a contradiction.

We deduce that this Nash Equilibrium is one of the worst ones, by using the same argument used in the proof of Theorem 6. Indeed, there cannot exist a Nash Equilibrium in which one agent gets a payoff lower than the one it would get by playing $(FCFS_i, b_i)$. We therefore conclude that $\{(FCFS_i, b_i)\}_{i\in[n]}$ is one of the worst Nash Equilibria, thus the PoA of $\mathbb{M}_{BFS}$ (and $\mathbb{M}_{DFS}$) is equal to 2.

To conclude that $PoS(\mathbb{M}_{BFS}) = PoS(\mathbb{M}_{DFS}) = 2$, consider the instance described in the proof of Theorem 10. Indeed, since in this instance, there is only one possible Nash Equilibrium, it is both the worst and best equilibrium possible. Moreover, since the PoS of this instance is equal to 2 and $PoS(\mathbb{M}_{BFS}) \leq PoA(\mathbb{M}_{BFS}) = 2$, we conclude the proof. Similarly, we have $PoS(\mathbb{M}_{DFS}) = 2$.

To show that all the bounds are tight, it suffice to notice that, according to our framework, an agent whose capacity is equal to 1 cannot manipulate the mechanism by reporting a null capacity. Indeed, an agent that reports a null capacity would automatically be excluded from the allocation procedure. Since in the instances we used to prove Theorem 9 and 10 all the agents have a capacity equal to 1, we can use them to prove the tightness of the bounds in the ECMS. □

Similarly, $\mathbb{M}_G$ is still truthful in the ECMS, and its approximation ratio is unchanged.

Theorem 12. *In the ECMS, $\mathbb{M}_G$ is truthful, its approximation ratio is equal to 2. Moreover, there is no deterministic truthful mechanism with a better approximation ratio. Finally, if the tasks have different values, $\mathbb{M}_G$ is group strategyproof.*

Proof. To show the truthfulness, it suffices to prove that if an agent misreports only its capacity, it cannot get a higher payoff. Indeed, if for any given set of edges reported by an agent its best strategy is to report the maximum capacity, we conclude that its best strategy, in the ECMS, is to report all the edges and the maximum capacity. Since every agent is bounded by its statements, this implies that $\mathbb{M}_G$ is truthful.

Let us assume that the capacity of agent a_i is b_i and that it gets a higher payoff by reporting $b'_i < b_i$ as its capacity. Let us denote with μ_i the set of tasks allocated to a_i from $\mathbb{M}_G$ when it reports truthfully and let us

denote with μ'_i the set of tasks allocated to a_i when it reports b'_i as its capacity. Toward a contradiction, let $t_j \in \mu'_i$ such that $t_j \notin \mu_i$. Since t_j is allocated to agent a_i by $\mathbb{M}_G$, it means that agent a_i is unsaturated at the j-th iteration of Algorithm 1 and since $b'_i < b_i$, we conclude that agent a_i is unsaturated at the j-th step also when it reports b_i as its capacity. Therefore, if task t_j is allocated to agent a_i when it reports b'_i as its capacity, the same holds when a_i reports a capacity of b_i. Since every task has a positive value and $\mu'_i \subset \mu_i$, we get a contradiction, therefore the mechanism cannot be manipulated by only misreporting the capacity. Since the mechanism $\mathbb{M}_G$ cannot be manipulated by hiding edges or by reporting a lower capacity, we conclude that $\mathbb{M}_G$ has to be truthful when the agents report both quantities. Finally, since $\mathbb{M}_G$ is truthful in the ECMS and the agents are bounded by their statements, its approximation ratio is the same as the one of $\mathbb{M}_G$ in EMS (since there is no difference in having the capacities publicly known and reporting them), hence $ar(\mathbb{M}_G) = 2$.

Again, to prove that the approximation ratio guarantee is tight, it suffice to consider the instance used to prove Theorem 3.

The group strategyproofness follows by the same argument used in the proof of Theorem 4 and by observing that reporting a lower capacity cannot increase the payoff of the agents. □

4 THE TASKS MANIPULATION CASE

We now consider the case in which the tasks' side is behaving strategically. As for the agents' case, we consider two frameworks. First, we consider the case in which the tasks' private information consists of only its adjacent edges. In the second one, the tasks' private information consists of their adjacent edges and their values. As we will see, all the mechanisms considered so far are truthful (albeit not group strategyproof). In particular, when the tasks' side behaves strategically, the optimality of a mechanism does not exclude its truthfulness.

4.1 The Game-Theoretical framework for tasks' manipulation

First, we adapt the definitions introduced in subsection 3.1 to this the Task manipulation framework.

The Strategy Space of the Tasks. Given a bipartite graph $G = (A \cup T, E)$, a capacity vector $\boldsymbol{b}$, a value vector $\boldsymbol{q}$, and a $\boldsymbol{b}$-matching μ over G. We define the social welfare achieved by μ as the total number of tasks assigned to agents. Therefore, in this setting, the social welfare achieved by μ is equal to the total number of edges in the matching, we use $v(\mu)$ to denote it. We then define the utility of task t_j as $v_j(\mu) := 1$ if t_j is assigned to any agent and 0 otherwise, so that $v(\mu) = \sum_{j=1}^{n} v_j(\mu)$. We focus on two settings:

- The private information of each task consists of the set of edges that connect it to the agent. As for the agents' framework, we call this setting Edge Manipulation Setting (EMS).
- The private information of each task consists of the set of edges and its own value. We call this setting Edge and Value Manipulation Setting (EVMS).

As for the agents' case, we assume that every task is bounded by its statement, which means that their reports can be incomplete, but they are not allowed to report false information. For the EMS, this means that a task can hide some of the edges that connect it to the agents, but it cannot report an edge that does not exist. The set of strategies of task t_j, namely $\mathcal{R}_j$, is therefore the set of all the possible non-empty subsets of G_j, where $G_j := \{e \in E \text{ s.t. } e = (a_i, t_j) \text{ for some } a_i \in A\}$ is the set containing all the existing edges that connect t_j to agents. In EVMS, it means that a task can report only a value that is lower than its real one and that it cannot report an edge that does not exist. In this case, the set of strategies of task t_j is $\mathcal{R}_j \times (0, q_j]$, where q_j is the real value of t_j.

The Mechanisms. A mechanism for the MVbM problem is a function $\mathbb{M}$ that takes as input the private information of the tasks and returns a $\boldsymbol{b}$-matching. For the sake of simplicity, we restrict our description to the EMS case, since the definitions for the EVMS case are similar. With a slight abuse of notation, we denote with $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$, the mechanisms induced by Algorithm 1 endowed with the respective graph traversing method. Similarly, we denote with $\mathbb{M}_G$ the mechanism induced by the approximated version of Algorithm 1.

Given a mechanism $\mathbb{M}$, every element $I \in \mathcal{I}_{\mathbb{M}}$ is composed by the reports of n self-interested tasks, so that $\mathcal{I}_{\mathbb{M}} = \otimes_{j=1}^{m} \mathcal{I}_j$, where $\mathcal{I}_j$ is the set of feasible inputs for task t_j. We say that a mechanism $\mathbb{M}$ is truthful if no task can get a higher payoff by hiding some of the edges. More formally, if I_j is the true type of task t_j, the mechanism is truthful if it holds true that $v_j(\mathbb{M}(I'_j, I_{-j})) \leq v_j(\mathbb{M}(I_j, I_{-j}))$ for every $I'_j \in \mathcal{R}_j$. Notice that, in this framework, this is equivalent to saying that an unmatched task cannot be matched if it hides some edges. Finally, a mechanism is group strategyproof for task manipulations if no group of tasks can collude to hide some of their edges in such a way that (i) all the tasks of the group that were matched in the truthful instance are still matched after the manipulation, (ii) at least one task that was unmatched in the truthful input gets matched to an agent.

4.2 The Properties of the Mechanisms

We now study whether $\mathbb{M}_{BFS}$, $\mathbb{M}_{DFS}$, and $\mathbb{M}_G$ are group strategyproof when the tasks are behaving strategically and are bound by their statements. The following theorem implies that any mechanism that returns an MVbM is not group strategyproof for task manipulation.

Theorem 13. *There is no group strategyproof mechanism that always returns an MVbM for task manipulation.*

Proof. We prove this statement using a counterexample. Consider two agents a_1 and a_2, three tasks t_1, t_2, t_3, and let the edge set be $E = \{(a_1, t_1), (a_1, t_3), (a_2, t_1), (a_2, t_2)\}$. The values of the three tasks are $q_1 = 1, q_2 = 0.9$, and $q_3 = 0.1$, respectively. The capacity of both the agents is equal to 1, so that $b_1 = b_2 = 1$. It is easy to see that the only MVbM is $\{(a_1, t_1), (a_2, t_2)\}$ with a total weight of 1.9. However, tasks t_1 and t_3 can collude. In fact, if t_1 hides the edge (a_1, t_1), the optimal matching becomes $\{(a_1, t_3), (a_2, t_1)\}$ with the total weight 1.1. Therefore, a mechanism that always returns a maximum weight matching is not group strategyproof. □

From Theorem 13 we immediately infer that both $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$ are not group strategyproof. Through a counterexample, we are able to prove that also $\mathbb{M}_G$ is not group strategyproof.

Theorem 14. *The mechanism $\mathbb{M}_G$ is not group strategyproof for task manipulation.*

Proof. Let us consider the following instance. We have three tasks, namely t_1, t_2, and t_3, whose values are $q_1 = 1$, $q_2 = 0.5$, and $q_3 = 0.3$, respectively, and two agents, namely a_1 and a_2. Let $E = \{(a_1, t_1), (a_1, t_3), (a_2, t_1), (a_2, t_2)\}$ be the truthful input. It is easy to check that $\mathbb{M}_G$ returns the matching $\{(a_1, t_1), (a_2, t_2)\}$. However task t_1 and t_3 can collude; indeed, if t_1 hides the edge (a_1, t_1), $\mathbb{M}_G$ returns the matching $\{(a_2, t_1), (a_1, t_3)\}$. □

Albeit $\mathbb{M}_{BFS}$, $\mathbb{M}_{DFS}$, and $\mathbb{M}_G$ are not group strategyproof, they are truthful for task manipulation. In particular, this means that both $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$ are truthful and optimal with respect to tasks manipulation.

Theorem 15. *$\mathbb{M}_{BFS}$, $\mathbb{M}_{DFS}$, and $\mathbb{M}_G$ are truthful for task manipulation. Moreover, we have that $ar(\mathbb{M}_{BFS}) = ar(\mathbb{M}_{DFS}) = 1$ and $ar(\mathbb{M}_G) = 2$.*

Proof. To show that $\mathbb{M}_{BFS}$ is truthful, it suffices to prove that an unmatched task cannot become matched by hiding the edges that connect it with agents in $\mathbb{M}_{BFS}$. Suppose that task t_j is not matched by the matching μ returned by $\mathbb{M}_{BFS}$. It is well known that, if t_j is unmatched by μ, it means that it was not matched during the j-th loop of Algorithm 1. If t_j hides some of its edges, there is still no augmenting path starting from t_j since the generated tree is a sub-tree of the one generated during the truthful iteration. We then conclude that t_j cannot be matched by misreporting and, hence, $\mathbb{M}_{BFS}$ is truthful. By the same argument, we infer that $\mathbb{M}_{DFS}$ and $\mathbb{M}_G$ are both truthful.

Since both $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$ return an MVbM, we have $ar(\mathbb{M}_{BFS}) = ar(\mathbb{M}_{DFS}) = 1$. Lastly, we show that $ar(\mathbb{M}_G) = 2$. Let us consider an MVbM problem, namely $\mathcal{P} := (A \cup T, E, \mathbf{b}, \mathbf{q})$. Without loss of generality, we assume that the tasks in T are ordered in a non-increasing fashion with respect to their value, that is $q_i \leq q_j$ if $j \leq i$. Let us then consider the auxiliary MVbM problem $\mathcal{P}' := (A' \cup T', E', \mathbf{b}', \mathbf{q}')$, for which $A' = A$, $T' = T$,

$E' = E$, and $\mathbf{b}' = \mathbf{b}$, so that the only parameter of $\mathcal{P}'$ that may differ from $\mathcal{P}$ is the vector $\mathbf{q}'$. The vector $\mathbf{q}'$ is such that $q'_i = 1$ for every $i = 1, \ldots, m$. Since all the tasks have the same value in $\mathcal{P}'$, we can assume that the tasks are ordered as in $\mathcal{P}$ while it still holds that $q'_j \leq q'_i$ whenever $i \leq j$. Since the set of tasks is ordered in the same way for both problems, $\mathbb{M}_G$ returns the same matching, namely μ_g, in both cases. The weight of μ_g for the problem $\mathcal{P}'$ is equal to the tasks' social welfare of μ_g for problem $\mathcal{P}$. Similarly, the weight of an MVbM solution to problem $\mathcal{P}'$ is equal to the maximal tasks' social welfare for problem $\mathcal{P}$. Finally, we recall that the algorithm defining $\mathbb{M}_G$ returns a matching whose weight is at least half of the weight of an MVbM solution [20]. Since this argument holds for every MVbM problem, we infer that

$$ar(\mathbb{M}_G) = \max_{I \in \mathcal{I}} \frac{v(\mu(I))}{v(\mathbb{M}_G(I))} \leq 2. \tag{2}$$

To conclude the thesis, we show that the bound in (2) is tight. Let us consider the following instance. There are two agents a_1 and a_2, whose capacities are both equal to 1, and two tasks t_1 and t_2, whose values are 1 and 0.5, respectively. Let $E = \{(a_1, t_1), (a_1, t_2), (a_2, t_1)\}$ be the truthful set of edges. It is easy to see that $\mathbb{M}_G(E) = \{(a_1, t_1)\}$, while the MVbM is $\{(a_2, t_1), (a_1, t_2)\}$. Hence the ratio between the maximum tasks' social welfare and the tasks' social welfare achieved by $\mathbb{M}_G$ is 2, which concludes the proof. □

4.3 Tasks Manipulating their Value.

We now consider the case in which the tasks are allowed to self-report their value. In this case, a mechanism $\mathbb{M}$ is truthful if it satisfies the condition that $v_j(\mathbb{M}((I'_j, q'_j), J_{-j})) \leq v_j(\mathbb{M}((I_j, q_j), J_{-j}))$ for every $(I'_j, q'_j) \in \mathcal{R}_j \times (0, q_j]$, where J_{-i} is the matrix containing the reports of the other tasks. To demonstrate that the mechanisms are truthful in the EVMS, we initially establish that all of them maintain truthfulness when the set of edges is publicly known, and the only private information pertains to the value of each task.

Lemma 3. *If the tasks are bounded by their statements, then both $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$ are truthful if the only input that the tasks have to provide is their value.*

Proof. We show that if a task, namely t_j, is not matched by Algorithm 1 when it reports q_j, it cannot be matched by reporting a value q'_j, where $q'_j < q_j$. Let us denote with μ and μ' the matching obtained when t_j reports value q_j and q'_j, respectively. Toward a contradiction, assume that t_j is not matched by μ, but, according to μ' it is matched with an agent, namely a_i. Since the set of edges does not change, and both μ and μ' are feasible b-matchings for both the instances (the one in which task t_j reports q_j and the one in which it reports q'_j), we must have

$$w(\mu) \leq w(\mu') = q'_j + \sum_{(i,j) \in \mu \setminus \{(a,t_j)\}} q_j < q_j + \sum_{(i,j) \in \mu \setminus \{(a,t_j)\}} q_j.$$

This concludes the contradiction since μ is an MVbM. □

By using a different argument, we prove that the same result holds for $\mathbb{M}_G$.

Lemma 4. *If the tasks are bounded by their statements, then $\mathbb{M}_G$ is truthful if the only input that the tasks have to provide is their value.*

Proof. Toward a contradiction, let us assume that there exists a task t_j whose real value is q_j and that, according to $\mathbb{M}_G$, is not matched when it reports truthfully but it gets matched if it reports a value q'_j such that $q'_j < q_j$. Let us denote with j the priority position t_j when it reports the value q_j and with j' the priority position that t_j gets when it reports the value q'_j. By hypothesis, we have $j \leq j'$. According to mechanism $\mathbb{M}_G$, task t_j is matched only if at the j-th step of Algorithm 1 t_j is connected to an unsaturated agent. Since $j < j'$, if an agent

is unsaturated at j'-th step and is also unsaturated at the j-th step, we conclude that t_j cannot be matched when its priority position is j' if it was not matched with priority position j, which is a contradiction. □

Finally, we show that if we allow the tasks to report both their edge set and value, all the mechanisms $\mathbb{M}_{BFS}$, $\mathbb{M}_{DFS}$, and $\mathbb{M}_G$, are truthful as long as the tasks are bounded by their statements. Thus, also in this extended case, both $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$ are truthful and optimal.

Theorem 16. *If the tasks are allowed to report both their value and their edges, the mechanisms $\mathbb{M}_{BFS}$, $\mathbb{M}_{DFS}$, and $\mathbb{M}_G$ are truthful as long as the tasks are bounded by their statements. Moreover, we have $ar(\mathbb{M}_{BFS}) = ar(\mathbb{M}_{DFS}) = 1$ and $ar(\mathbb{M}_G) = 2$.*

Proof. Indeed, let us consider a task, namely t_j, whose truthful input is (q_j, T_j), where q_j is its value and T_j is the set of edges it is connected to. Toward a contradiction, let us assume that t_j is not matched if it reports truthfully, but gets matched if it manipulates and reports (q'_j, T'_j). Since the task is bounded by its statement, we have both $q'_j \leq q_j$ and $T'_j \subset T_j$. Let us now consider the matching returned by the mechanism when the task reports (q'_j, T_j) (which means, t_j has only manipulated its value and not T_j). According to Lemma 3, if t_j could not get matched by reporting its true value, then it could not be matched by reporting a lower one. We, therefore, conclude that t_j is not matched when it reports (q'_j, T_j). To conclude, we observe that if the task is not matched when it reports $(q'_j.T_j)$, by Theorem 15, it cannot be matched when it reports (q'_j, T'_j) with $T'_j \subset T_j$, which contradicts the initial assumption.

Since both $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$ are optimal, we get $ar(\mathbb{M}_{BFS}) = ar(\mathbb{M}_{DFS}) = 1$. By the same argument used in the proof of Theorem 2, we conclude $ar(\mathbb{M}_G) = 2$. Indeed, since the agents are bounded by their statements and the mechanisms is truthful, there is no difference between letting the values of the tasks be publicly known and letting the tasks self-report their value. □

5 EXPERIMENTAL EVALUATION

In this section, we conduct several numerical tests to evaluate the manipulability of both $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$. Since $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$ are truthful when tasks behave strategically, our analysis pertains to scenarios where agents are the strategic entities. Furthermore, we narrow our focus to the EMS, as we have proven that the ECMS yields no meaningful differences from the EMS. Our experiments aim to achieve two primary objectives: (i) First, we compare the susceptibility of $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$ to manipulation by the first agent. (ii) Second, we focus specifically on $\mathbb{M}_{BFS}$ and evaluate its vulnerability to manipulation by every agent.

5.1 Dataset Generation

We now detail the parameters that characterize how the instances are randomly generated and introduce the class of manipulations we consider.

The truthful input. Given the number of agents n and the number of tasks m, we generate a random truthful instance $G = ([n] \cup [m], E, \boldsymbol{b}, \boldsymbol{q})$ as follows:

- the capacity b_i of the agent a_i is an integer number sampled from a uniform probability distribution on the set of natural numbers between $\underline{b}$ and $\overline{b}$, i.e. over the set $\{\underline{b}, \underline{b}+1, \ldots, \overline{b}-1, \overline{b}\}$;
- in all the tests, the value q_j of task t_j is sampled from a $Y = \max\{Z, 0\}$, where $Z \sim \mathcal{N}(3, 0.77)$, so that the values of the tasks are all between 1 and 5 with probability 0.99;
- every possible edge $(a_i, t_j) \in [n] \times [m]$ has a probability p to be in E. We will use the term *connection probability* to denote p.

We consider as truthful every instance generated this way.

Agent manipulation. Given an agent a_i, let T_i be the set of edges connected to a_i so that the set of a_i's possible strategies is the set of subsets of T_i. As a consequence, every agent has $2^{|T_i|}$ possible ways to misreport,

which is unfeasible from a computational point of view. Although the Nash Equilibria of the mechanisms can be easily computed, the best strategy of an agent given the reports of all the other agents does not have an explicit form. For this reason, we focus only on the case in which the agent hides its connections to the tasks with the lowest value. We describe these strategies using the *cutoff threshold* T. Given an agent a_i and a cutoff threshold T, the T-level manipulation of agent a_i consists of hiding all its connections with tasks with a value lower than T.

5.2 Comparison between $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$.

The test we run is as follows: given the number of agents n, the number of tasks m, and the connection probability p, we apply both $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$ on 250 randomly generated instances. As a metric to compare the manipulability of the two mechanisms, we consider the Percentage Utility Gain of first agent, which we define as

$$p_1(G) := \frac{\max\{w_1(G') - w_1(G), 0\}}{w_1(G)}, \tag{3}$$

where G is the truthful instance and G' the instance after agent a_1 plays the strategy outlined in Theorem 5. The Percentage Utility Gain is therefore the ratio between the payoff increase that the first agent gets from $\mathbb{M}_{BFS}$ or $\mathbb{M}_{DFS}$ if it manipulates the mechanism and the payoff it would get by being truthful. By definition, when the ratio is equal to 0, the agent is not able to manipulate. This metric tells us whether the agent is able to manipulate and how big is the gain the first agent would get by manipulating.

In Table 2 we report the maximum and minimum Percentage Utility Gains of the first agent among the 250 iterations we run. We consider the case in which the number of tasks is $m = 30, 50, 70$, the number of agents is $n = 20, 40, 60, 80$, and $p = 0.4, 0.6, 0.8$. For the sake of simplicity, we set the capacity of every agent equal to 3, so that $\underline{b} = \overline{b} = 3$ We notice that, in the vast majority of cases, the maximum Percentage Utility Gain of the $\mathbb{M}_{BFS}$ is equal to 0, which means that the first agent was unable to manipulate $\mathbb{M}_{BFS}$ on all of the 250 randomly generated instances. On the contrary, the maximum Percentage Utility Gain of the $\mathbb{M}_{DFS}$ is always greater than zero. Similarly, we observe that the minimum Percentage Utility Gain of the first agent from $\mathbb{M}_{DFS}$ is equal to 0 only in two cases. This means that, for all but these two parameter choices, on all the 250 instances we randomly generated according to those parameters, $\mathbb{M}_{DFS}$ was always manipulable by the first agent. It is worth to observe that, from our finding the $\mathbb{M}_{BFS}$ becomes less manipulable by the first agent as the parameters n and p grow. We observe a distinct behaviour from $\mathbb{M}_{DFS}$, for which the first agent's ability to manipulate $\mathbb{M}_{DFS}$ seems to remain constant for every value of n, m, and p.

5.3 The Manipulability of $\mathbb{M}_{BFS}$.

We now move our focus on the $\mathbb{M}_{BFS}$. In particular, we want to assess the manipulability of the $\mathbb{M}_{BFS}$ when every agent tries to manipulate. For this reason, we consider two metrics: (i) the Percentage of Manipulative Agents (PMA), which measures the average number of agents that can manipulate the mechanism, and (ii) the Percentage of Manipulable Instances (PMI) of the mechanism, which measures the number of instances that are manipulable by at least one agent.

Percentage of Manipulative Agents. We now run a test to assess the percentage of agents that can benefit from manipulation, i.e. the number of agents that are able to manipulate over the total number of agents. The parameters are fixed as follows. For each triplet of $n = 10, 15, 20$, $m = 100, 125, 150, 175, 200$, and $p = 0.1, 0.2, 0.4$, we generate 250 instances and compute the number of manipulative agents for each of those. The capacity of every agent is randomly sampled in a discrete set, in our tests we consider the cases in which $\{1, 2, 3, 4, 5\}$, $\{3, 4, 5, 6, 7\}$, and $\{5, 6, 7, 8, 9, 10\}$. Finally, for every agent, we consider the T-level manipulations where $T = 1.5, 2, 2.5, 3$.

In Table 4, we report the average PMA for each triplet, i.e. the average percentage of agents that are able to get a higher payoff by applying one of the previously described T-level manipulations. The results show that when the number of tasks is higher than the total capacity of the agents, the number of manipulative agents

$m = 30$	$n = 20$				$n = 40$				$n = 60$				$n = 80$			
	$\mathbb{M}_{BFS}$		$\mathbb{M}_{DFS}$		$\mathbb{M}_{BFS}$		$\mathbb{M}_{DFS}$		$\mathbb{M}_{BFS}$		$\mathbb{M}_{DFS}$		$\mathbb{M}_{BFS}$		$\mathbb{M}_{DFS}$	
	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min
$p = 0.4$	0.12	0.0	0.28	0.0	0.0	0.0	0.30	0.04	0.0	0.0	0.31	0.0	0.0	0.0	0.32	0.01
$p = 0.6$	0.0	0.0	0.24	0.0	0.0	0.0	0.26	0.06	0.0	0.0	0.24	0.06	0.0	0.0	0.24	0.06
$p = 0.8$	0.0	0.0	0.19	0.06	0.0	0.0	0.21	0.06	0.0	0.0	0.16	0.06	0.0	0.0	0.18	0.06
$m = 50$	$n = 20$				$n = 40$				$n = 60$				$n = 80$			
	$\mathbb{M}_{BFS}$		$\mathbb{M}_{DFS}$		$\mathbb{M}_{BFS}$		$\mathbb{M}_{DFS}$		$\mathbb{M}_{BFS}$		$\mathbb{M}_{DFS}$		$\mathbb{M}_{BFS}$		$\mathbb{M}_{DFS}$	
	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min
$p = 0.4$	0.26	0.0	0.24	0.02	0.0	0.0	0.24	0.03	0.0	0.0	0.24	0.02	0.0	0.0	0.26	0.03
$p = 0.6$	0.25	0.0	0.16	0.02	0.0	0.0	0.21	0.02	0.0	0.0	0.16	0.02	0.0	0.0	0.21	0.02
$p = 0.8$	0.14	0.0	0.13	0.02	0.0	0.0	0.10	0.03	0.0	0.0	0.13	0.02	0.0	0.0	0.13	0.02
$m = 70$	$n = 20$				$n = 40$				$n = 60$				$n = 80$			
	$\mathbb{M}_{BFS}$		$\mathbb{M}_{DFS}$		$\mathbb{M}_{BFS}$		$\mathbb{M}_{DFS}$		$\mathbb{M}_{BFS}$		$\mathbb{M}_{DFS}$		$\mathbb{M}_{BFS}$		$\mathbb{M}_{DFS}$	
	max	min	max	min	max	min	max	min	max	min	max	min	max	min	max	min
$p = 0.4$	0.39	0.0	0.23	0.02	0.0	0.0	0.21	0.04	0.0	0.0	0.23	0.02	0.0	0.0	0.25	0.03
$p = 0.6$	0.41	0.0	0.18	0.02	0.0	0.0	0.17	0.04	0.0	0.0	0.18	0.02	0.0	0.0	0.18	0.04
$p = 0.8$	0.41	0.0	0.15	0.04	0.0	0.0	0.15	0.04	0.0	0.0	0.15	0.04	0.0	0.0	0.14	0.04

Table 2. Every column labeled with $\mathbb{M}_{BFS}$ ($\mathbb{M}_{DFS}$, respectively) reports the maximum and minimum difference between what the first agent gets from $\mathbb{M}_{BFS}$ ($\mathbb{M}_{DFS}$) when it reports truthfully and the payoff that it can achieve by applying its FCFS policy. The average is taken on 250 randomly generated instances.

and manipulable instances decreases. Similarly, when the number of tasks is lower than the total capacity of the agents, the number of manipulable instances starts to decrease as well. Going back to the worker-project example, these experimental results suggest that when the number of projects is enough to ensure every worker will be associated with its maximum number of tasks, hiding the connections with low-value tasks always leads to a higher utility. On the contrary, when there is a shortage of projects, hiding the connections with low-value tasks may lead to a loss since it is no longer certain that each worker will be associated with a number of projects equal to its capacity.

Percentage of Manipulable Instances. Lastly, we run a test to determine the average number of instances in which there exists at least an agent that can manipulate the matching process. The test specifics are as follows. For each triplet of $n = 10, 15, 20$, $m = 100, 125, 150, 175, 200$, and $p = 0.1, 0.2, 0.4$, we generate 250 instances and compute the number of manipulative agents for each of those. The capacity of every agent is randomly sampled in a discrete set, namely B, in our tests we consider $B = \{1, 2, 3, 4, 5\}$, $\{3, 4, 5, 6, 7\}$, and $\{5, 6, 7, 8, 9, 10\}$. Finally, to simulate the manipulations of the agents we let them remove the edges connecting them to the tasks with values lower than $T = 1.5, 2, 2.5$, and 3.

For each triplet of n, m, and p, we create 250 instances and compute the number of manipulative agents for each of those. In Table 3, we report the average number of instances in which at least an agent was able to manipulate the outcome of the matching in its favour. According to what we observed for the PMA, the average PMI increases as the total capacity approaches the total amount of tasks.

6 CONCLUSION AND FUTURE WORK

In this paper, we studied a game-theoretical framework for the MVbM problems in which one side of the bipartite graph is composed of agents and the other one of tasks that possess an objective value. We have considered

$b \in \{1,2,3,4,5\}$	$n = 10$			$n = 15$			$n = 20$		
	$p = 0.1$	$p = 0.2$	$p = 0.4$	$p = 0.1$	$p = 0.2$	$p = 0.4$	$p = 0.1$	$p = 0.2$	$p = 0.4$
$m = 100$	0.26	0.0	0.0	0.52	0.02	0.01	0.81	0.41	0.38
$m = 125$	0.05	0.0	0.0	0.13	0.0	0.0	0.36	0.03	0.01
$m = 150$	0.01	0.0	0.0	0.01	0.0	0.0	0.06	0.0	0.0
$m = 175$	0.0	0.0	0.0	0.0	0.0	0.0	0.01	0.0	0.0
$m = 200$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$b \in \{3,4,5,6,7\}$	$n = 10$			$n = 15$			$n = 20$		
	$p = 0.1$	$p = 0.2$	$p = 0.4$	$p = 0.1$	$p = 0.2$	$p = 0.4$	$p = 0.1$	$p = 0.2$	$p = 0.4$
$m = 100$	0.90	0.20	0.04	1.0	0.98	0.94	1.0	1.0	1.0
$m = 125$	0.54	0.01	0.0	0.98	0.49	0.35	1.0	0.99	0.98
$m = 150$	0.19	0.0	0.0	0.59	0.02	0.02	0.97	0.78	0.73
$m = 175$	0.04	0.0	0.0	0.18	0.0	0.0	0.66	0.18	0.09
$m = 200$	0.01	0.0	0.0	0.03	0.0	0.0	0.13	0.01	0.01
$b \in \{5,6,7,8,9,10\}$	$n = 10$			$n = 15$			$n = 20$		
	$p = 0.1$	$p = 0.2$	$p = 0.4$	$p = 0.1$	$p = 0.2$	$p = 0.4$	$p = 0.1$	$p = 0.2$	$p = 0.4$
$m = 100$	0.96	1.0	0.96	0.99	1.0	1.0	0.98	1.0	1.0
$m = 125$	0.99	0.74	0.33	1.0	1.0	1.0	1.0	1.0	1.0
$m = 150$	0.96	0.10	0.03	1.0	0.99	0.98	1.0	1.0	1.0
$m = 175$	0.70	0.01	0.0	1.0	0.76	0.56	1.0	1.0	1.0
$m = 200$	0.35	0.0	0.0	0.93	0.18	0.09	1.0	0.99	0.98

Table 3. Average percentages of manipulable instances for different values of n, m, and randomly generated capacities. The averages are taken over 250 instances randomly generated. The utility vector is sampled from a uniform distribution over $[1, 5]$, and the thresholds T are taken in the set $\{1.5, 2, 2.5, 3\}$.

both cases in which the agents and the tasks can behave strategically by either hiding some of the connections with the other side of the bipartite graph or hiding their connections and lowering their capacity/value. In this framework, we considered three mechanisms, $\mathbb{M}_{BFS}$, $\mathbb{M}_{DFS}$, and $\mathbb{M}_G$, and fully characterized their truthfulness, approximation ratio, PoA, and PoS. For the agents' manipulation case, we have shown that, albeit $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$ are not truthful, they are optimal from a PoA and PoS point of view for all the cases. In particular, $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$ are optimal from both a complexity and game theoretical viewpoint. On the contrary, $\mathbb{M}_G$ is truthful and has an approximation ratio equal to 2, which we proved to be the best possible approximation ratio achievable by truthful mechanisms in this framework. Remarkably, these results are true in both the EMS and the EMCS.

We then studied these mechanisms when we allow tasks to behave strategically. In this simpler framework, all the mechanisms are truthful, thus both $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$ are optimal and truthful. To conclude, we numerically compared the manipulability of $\mathbb{M}_{BFS}$ and $\mathbb{M}_{DFS}$ on randomly generated instances and studied more in detail $\mathbb{M}_{BFS}$.

A future development of our work would be to further study the manipulability of the mechanisms when we introduce a random shuffling on the set of agents. This study could lead to detect other sets of instances in which the algorithm behaves truthfully. Another interesting development is to consider multi-layered graphs to better represent different characteristics that the tasks may have. Finally, it would be interesting to study how limiting the amount of edges that every agent can report affects the performance and manipulability of the mechanism itself.

$b \in \{1,2,3,4,5\}$	$n=10$			$n=15$			$n=20$		
	$p=0.1$	$p=0.2$	$p=0.4$	$p=0.1$	$p=0.2$	$p=0.4$	$p=0.1$	$p=0.2$	$p=0.4$
$m=100$	0.04	0.0	0.0	0.09	0.0	0.0	0.22	0.11	0.08
$m=125$	0.01	0.0	0.0	0.01	0.0	0.0	0.05	0.01	0.01
$m=150$	0.0	0.0	0.0	0.0	0.0	0.0	0.01	0.0	0.0
$m=175$	0.0	0.0	0.0	0.0	0.0	0.0	0.01	0.0	0.0
$m=200$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$b \in \{3,4,5,6,7\}$	$n=10$			$n=15$			$n=20$		
	$p=0.1$	$p=0.2$	$p=0.4$	$p=0.1$	$p=0.2$	$p=0.4$	$p=0.1$	$p=0.2$	$p=0.4$
$m=100$	0.33	0.06	0.01	0.59	0.62	0.44	0.59	0.83	0.66
$m=125$	0.14	0.01	0.0	0.44	0.15	0.08	0.71	0.66	0.50
$m=150$	0.04	0.0	0.0	0.17	0.01	0.01	0.54	0.28	0.20
$m=175$	0.01	0.0	0.0	0.04	0.0	0.0	0.21	0.04	0.01
$m=200$	0.01	0.0	0.0	0.01	0.0	0.0	0.03	0.01	0.0
$b \in \{5,6,7,8,9,10\}$	$n=10$			$n=15$			$n=20$		
	$p=0.1$	$p=0.2$	$p=0.4$	$p=0.1$	$p=0.2$	$p=0.4$	$p=0.1$	$p=0.2$	$p=0.4$
$m=100$	0.37	0.82	0.58	0.29	0.81	0.76	0.20	0.60	0.59
$m=125$	0.57	0.39	0.33	0.57	0.91	0.72	0.45	0.79	0.68
$m=150$	0.49	0.04	0.03	0.79	0.75	0.49	0.72	0.92	0.72
$m=175$	0.27	0.01	0.0	0.73	0.35	0.17	0.89	0.85	0.62
$m=200$	0.08	0.0	0.0	0.49	0.05	0.02	0.85	0.68	0.45

Table 4. Average percentages of manipulative agents for different values of n, m, and randomly generated capacities. The averages are taken over 250 instances randomly generated. The utility vector is sampled from a uniform distribution over $[1,5]$, and the thresholds T are taken in the set $\{1.5, 2, 2.5, 3\}$.

ACKNOWLEDGMENTS

Gennaro Auricchio and Jie Zhang are partially supported by a Leverhulme Trust Research Project Grant (2021 – 2024). Jie Zhang is also supported by the EPSRC grant (EP/W014912/1). Jun Liu is supported by Beijing Natural Science Foundation (No. 1232011).

REFERENCES

[1] Atila Abdulkadiroğlu. 2005. College admissions with affirmative action. *International Journal of Game Theory* 33, 4 (2005), 535–549.
[2] Atila Abdulkadiroğlu and Tayfun Sönmez. 1999. House allocation with existing tenants. *Journal of Economic Theory* 88, 2 (1999), 233–260.
[3] Atila Abdulkadiroğlu and Tayfun Sönmez. 2003. School choice: A mechanism design approach. *American economic review* 93, 3 (2003), 729–747.
[4] Gagan Aggarwal, S. Muthukrishnan, Dávid Pál, and Martin Pál. 2009. General auction mechanism for search advertising. In *WWW*. 241–250.
[5] Meltem Aksoy, Seda Yanik, and Mehmet Fatih Amasyali. 2023. Reviewer Assignment Problem: A Systematic Review of the Literature. *Journal of Artificial Intelligence Research* 76 (2023), 761–827.
[6] Ahmed Al-Herz and Alex Pothen. 2022. A 2/3-approximation algorithm for vertex-weighted matching. *Discrete Applied Mathematics* 308 (2022), 46–67.
[7] Itai Ashlagi, Felix Fischer, Ian A Kash, and Ariel D Procaccia. 2015. Mix and match: A strategyproof mechanism for multi-hospital kidney exchange. *Games and Economic Behavior* 91 (2015), 284–296.
[8] Itai Ashlagi and Alvin E Roth. 2021. Kidney exchange: An operations perspective. *Management Science* 67, 9 (2021), 5455–5478.

[9] Gennaro Auricchio, Federico Bassetti, Stefano Gualandi, and Marco Veneroni. 2019. Computing Wasserstein Barycenters via Linear Programming. In *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*. Springer International Publishing, Cham, 355–363.
[10] Gennaro Auricchio and Jie Zhang. 2023. On the Manipulability of Maximum Vertex-Weighted Bipartite b-Matching Mechanisms. In *Proceedings of the Twenty-sixth European Conference on Artificial Intelligence (ECAI 2023)*. 125 – 132.
[11] Gennaro Auricchio, Ruixiao Zhang, Jie Zhang, and Xiaohao Cai. 2023. A Bilevel Formalism for the Peer-Reviewing Problem. In *ECAI 2023*. IOS Press, 133–140.
[12] Haris Aziz, Péter Biró, and Makoto Yokoo. 2022. Matching Market Design with Constraints. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 12308–12316.
[13] Michel Balinski and Tayfun Sönmez. 1999. A Tale of Two Mechanisms: Student Placement. *Journal of Economic theory* 84, 1 (1999), 73–94.
[14] M. Bayati, D. Shah, and M. Sharma. 2005. Maximum weight matching via max-product belief propagation. In *Proceedings. International Symposium on Information Theory, 2005. ISIT 2005.* 1763–1767.
[15] Márton Benedek, Péter Biró, Matthew Johnson, Daniël Paulusma, and Xin Ye. 2023. The complexity of matching games: a survey. *Journal of Artificial Intelligence Research* 77 (2023), 459–485.
[16] Benjamin E. Birnbaum and Claire Mathieu. 2008. On-line Bipartite Matching Made Simple. *SIGACT News* 39, 1 (2008), 80–87.
[17] Péter Biró, Tamás Fleiner, Robert W. Irving, and David F. Manlove. 2010. The College Admissions problem with lower and common quotas. *Theor. Comput. Sci.* 411, 34-36 (2010), 3136–3153.
[18] Yan Chen and Tayfun Sönmez. 2002. Improving Efficiency of On-Campus Housing: An Experimental Study. *American Economic Review* 92, 5 (2002), 1669–1686.
[19] Lizhong Ding, Zheyun Feng, and Yongsheng Bai. 2019. Clustering analysis of microRNA and mRNA expression data from TCGA using maximum edge-weighted matching algorithms. *BMC medical genomics* 12, 1 (2019), 1–27.
[20] Florin Dobrian, Mahantesh Halappanavar, Alex Pothen, and Ahmed Al-Herz. 2019. A 2/3-approximation algorithm for vertex weighted matching in bipartite graphs. *SIAM Journal on Scientific Computing* 41, 1 (2019), A566–A591.
[21] Thomas. E. Easterfield. 1946. A Combinatorial Algorithm. *Journal of The London Mathematical Society* 21 (1946), 219–226.
[22] Jack Edmonds and Delbert Ray Fulkerson. 1970. Bottleneck Extrema. *Journal of Combinatorial Theory* 8, 3 (1970), 299–306.
[23] Lars Ehlers, Isa E Hafalir, M Bumin Yenmez, and Muhammed A Yildirim. 2014. School choice with controlled choice constraints: Hard bounds versus soft bounds. *Journal of Economic theory* 153 (2014), 648–683.
[24] Alireza Farhadi, Mohammad Ghodsi, Mohammad Taghi Hajiaghayi, Sebastien Lahaie, David Pennock, Masoud Seddighin, Saeed Seddighin, and Hadi Yami. 2019. Fair allocation of indivisible goods to asymmetric agents. *Journal of Artificial Intelligence Research* 64 (2019), 1–20.
[25] Michal Feldman, Federico Fusco, Stefano Leonardi, Simon Mauras, and Rebecca Reiffenhäuser. 2024. Truthful matching with online items and offline agents. *Algorithmica* 1432-0541 (2024), 1–23.
[26] Daquan Feng, Lu Lu, Yi Yuan-Wu, Geoffrey Ye Li, Gang Feng, and Shaoqian Li. 2013. Device-to-Device Communications Underlaying Cellular Networks. *IEEE Transactions on communications* 61, 8 (2013), 3541–3551.
[27] Daniel Fragiadakis, Atsushi Iwasaki, Peter Troyan, Suguru Ueda, and Makoto Yokoo. 2016. Strategyproof Matching with Minimum Quotas. *ACM Transactions on Economics and Computation (TEAC)* 4, 1 (2016), 1–40.
[28] Komei Fukuda and Tomomi Matsui. 1994. Finding All the Perfect Matchings in Bipartite Graphs. *Applied Mathematics Letters* 7, 1 (1994), 15–18.
[29] David Gale and Lloyd S Shapley. 1962. College admissions and the stability of marriage. *The American Mathematical Monthly* 69, 1 (1962), 9–15.
[30] Anupam Gupta, Amit Kumar, and Cliff Stein. 2014. Maintaining Assignments Online: Matching, Scheduling, and Flows. In *SODA*. 468–479.
[31] Chen Hajaj, John Dickerson, Avinatan Hassidim, Tuomas Sandholm, and David Sarne. 2015. Strategy-proof and efficient kidney exchange using a credit mechanism. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 29.
[32] Frank L. Hitchcock. 1941. The Distribution of a Product from Several Sources to Numerous Localities. *Journal of Mathematics and Physics* 20, 1-4 (1941), 224–230.
[33] Aanund Hylland and Richard Zeckhauser. 1979. The Efficient Allocation of Individuals to Positions. *Journal of Political Economy* 87, 2 (1979), 293–314.
[34] Yuichiro Kamada and Fuhito Kojima. 2017. Recent Developments in Matching with Constraints. *American Economic Review* 107, 5 (2017), 200–204.
[35] Yuichiro Kamada and Fuhito Kojima. 2023. Fair matching under constraints: Theory and applications. *Review of Economic Studies* 2 (2023), 1162–1199.
[36] L. Kantorovitch. 1958. On the Translocation of Masses. *Management Science* 5, 1 (1958), 1–4.
[37] Fuhito Kojima. 2019. New Directions of Study in Matching with Constraints. In *The Future of Economic Design*. Springer, 479–482.

[38] Piotr Krysta, David Manlove, Baharak Rastegari, and Jinshan Zhang. 2014. Size versus Truthfulness in the House Allocation Problem. In *Proceedings of the fifteenth ACM conference on Economics and computation*. 453–470.
[39] Amanda R Kube, Sanmay Das, and Patrick J Fowler. 2023. Fair and efficient allocation of scarce resources based on predicted outcomes: implications for homeless service delivery. *Journal of Artificial Intelligence Research* 76 (2023), 1219–1245.
[40] Nathaniel Lahn, Sharath Raghvendra, and Jiacheng Ye. 2021. A Faster Maximum Cardinality Matching Algorithm with Applications in Machine Learning. *Advances in Neural Information Processing Systems* 34 (2021), 16885–16898.
[41] László Lovász and Michael D Plummer. 2009. *Matching theory*. Vol. 367. American Mathematical Society, Michigan, USA.
[42] David Manlove. 2013. *Algorithmics of matching under preferences*. Vol. 2. World Scientific, Singapore.
[43] Aranyak Mehta. 2013. Online Matching and Ad Allocation. *Found. Trends Theor. Comput. Sci.* 8, 4 (2013), 265–368.
[44] Alvin E Roth. 1982. Incentive compatibility in a market with indivisible goods. *Economics letters* 9, 2 (1982), 127–132.
[45] Alvin E Roth, Tayfun Sönmez, and M Utku Ünver. 2005. Pairwise kidney exchange. *Journal of Economic theory* 125, 2 (2005), 151–188.
[46] Lloyd Shapley and Herbert Scarf. 1974. On cores and indivisibility. *Journal of mathematical economics* 1, 1 (1974), 23–37.
[47] Thomas H. Spencer and Ernst W. Mayr. 1984. Node Weighted Matching. In *Automata, Languages and Programming*, Jan Paredaens (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 454–464.
[48] Xuanming Su and Stefanos A Zenios. 2006. Recipient choice can address the efficiency-equity trade-off in kidney transplantation: A mechanism design model. *Management science* 52, 11 (2006), 1647–1660.
[49] Robert L. Thorndike. 1950. The problem of classification of personnel. *Psychometrika* 15 (1950), 215–235.
[50] Li Wang, Huaqing Wu, Wei Wang, and Kwang-Cheng Chen. 2015. Socially Enabled Wireless Networks: Resource Allocation via Bipartite Graph Matching. *IEEE Communications Magazine* 53, 10 (2015), 128–135.
[51] Yu Yokoi. 2017. A Generalized Polymatroid Approach to Stable Matchings with Lower Quotas. *Mathematics of Operations Research* 42, 1 (2017), 238–255.
[52] Chuanli Zhao and Hengyong Tang. 2010. Single machine scheduling with general job-dependent aging effect and maintenance activities to minimize makespan. *Applied Mathematical Modelling* 34, 3 (2010), 837–841.

Received 25 March 2024; revised 14 September 2024; accepted 18 October 2024