To Trust or Not to Trust: Assignment Mechanisms with Predictions in the Private Graph Model

RICCARDO COLINI-BALDESCHI, Central Applied Science, Meta, United Kingdom
SOPHIE KLUMPER*, Centrum Wiskunde & Informatica (CWI), Netherlands
GUIDO SCHÄFER†, Centrum Wiskunde & Informatica (CWI), Netherlands
ARTEM TSIKIRIDIS, Centrum Wiskunde & Informatica (CWI), Netherlands

The realm of algorithms with predictions has led to the development of several new algorithms that leverage predictions to enhance their performance guarantees. The challenge is to devise algorithms that achieve optimal approximation guarantees as the prediction quality varies from perfect (consistency) to imperfect (robustness). This framework is particularly appealing in mechanism design contexts, where predictions might convey private information about the agents. In this paper, we design strategyproof mechanisms that leverage predictions to achieve improved approximation guarantees for several variants of the Generalized Assignment Problem (GAP) in the private graph model. In this model, first introduced by Dughmi & Ghosh (2010), the set of resources that an agent is compatible with is private information. For the Bipartite Matching Problem (BMP), we give a deterministic group-strategyproof (GSP) mechanism that is $(1 + 1/\gamma)$-consistent and $(1 + \gamma)$-robust, where $\gamma \geq 1$ is some confidence parameter. We also prove that this is best possible. Remarkably, our mechanism draws inspiration from the renowned Gale-Shapley algorithm, incorporating predictions as a crucial element. Additionally, we give a randomized mechanism that is universally GSP and improves on the guarantees in expectation. The other GAP variants that we consider all make use of a unified greedy mechanism that adds edges to the assignment according to a specific order. For a special case of Restricted Multiple Knapsack, this results in a deterministic strategyproof mechanism that is $(1 + 1/\gamma)$-consistent and $(2 + \gamma)$-robust. We then focus on two variants: Agent Size GAP (where each agent has one size) and Value Consensus GAP (where all agents have the same preference order over resources). For both variants, our universally GSP mechanisms randomize over the greedy mechanism, our mechanism for BMP and the predicted assignment, leading to $(1+3/\gamma)$-consistency and $(3+\gamma)$-robustness in expectation. All our mechanisms also provide more fine-grained approximation guarantees that interpolate between the consistency and robustness guarantees, depending on some natural error measure of the prediction.

CCS Concepts: • **Theory of computation** → **Algorithmic mechanism design**.

Additional Key Words and Phrases: Learning-augmented Mechanisms, Predictions, Matching, Generalized Assignment Problem, Private Graph

ACM Reference Format:
Riccardo Colini-Baldeschi, Sophie Klumper, Guido Schäfer, and Artem Tsikiridis. 2024. To Trust or Not to Trust: Assignment Mechanisms with Predictions in the Private Graph Model . In *The 25th Conference on*

*Also with Institute for Logic, Language and Computation (ILLC), University of Amsterdam.
†Also with Institute for Logic, Language and Computation (ILLC), University of Amsterdam.

Authors' Contact Information: Riccardo Colini-Baldeschi, rickuz@meta.com, Central Applied Science, Meta, London, United Kingdom; Sophie Klumper, s.klumper@cwi.nl, Centrum Wiskunde & Informatica (CWI), Amsterdam, Netherlands; Guido Schäfer, g.schaefer@cwi.nl, Centrum Wiskunde & Informatica (CWI), Amsterdam, Netherlands; Artem Tsikiridis, artem.tsikiridis@cwi.nl, Centrum Wiskunde & Informatica (CWI), Amsterdam, Netherlands.

EC '24, July 8–11, 2024, New Haven, CT, USA

ACM ISBN 979-8-4007-0704-9/24/07
https://doi.org/10.1145/3670865.3673578

Economics and Computation (EC '24), July 8–11, 2024, New Haven, CT, USA. ACM, New York, NY, USA, 21 pages. https://doi.org/10.1145/3670865.3673578

1 Introduction

Mechanism design is centered around the study of situations where multiple self-interested agents interact within a system. Each agent holds some private information about their preferences (also called *type*), based on which they make decisions. The primary goal is to create systems such that, despite the agents acting in their own self-interest, the outcome is socially desirable or optimal from the designer's perspective. One of the key challenges is to design mechanisms that incentivize the agents to reveal their preferences truthfully. A prominent notion in this context is *strategyproofness* which ensures that it is in the best interest of each agent to reveal their preferences truthfully, independently of the other agents. Unfortunately, strategyproofness often imposes strong impossibility results on achieving the socially desirable objective optimally or even approximately. As a consequence, the worst-case approximation guarantees derived in the literature can be rather disappointing from a practical perspective (see, e.g., Roughgarden [2019]).

Mechanism Design with Predictions. To overcome these limitations, a new line of research, called *mechanism design with predictions*, is exploring how to leverage learning-augmented inputs, such as information about the private types of the agents or the structure of the optimal solution, in the design of mechanisms. While this line of research first emerged in the area of online algorithms (see, e.g., Lykouris and Vassilvitskii [2021]), it is particularly appealing in the context of mechanism design because in economic environments this information can oftentimes be extracted from data through machine-learning techniques. In the context of mechanism design, Agrawal et al. [2023] and Xu and Lu [2022] are among the first works along this line.

In the mechanism design with predictions framework, the designer can exploit the predicted information to improve the worst-case efficiency of their mechanism — but the predictions might be inaccurate, or even entirely erroneous. As a result, the goal is to design mechanisms that guarantee attractive approximation guarantees if the prediction is perfect (referred to as *consistency*), while still maintaining a reasonable worst-case guarantee when the prediction is imperfect (referred to as *robustness*). Ideally, the mechanism provides a fine-grained approximation guarantee, depending on some measure of the prediction error, which smoothly interpolates between these two extreme cases (referred to as *approximation*).

In this paper, we study how to leverage learning-augmented predictions in the domain of mechanism design *without money*. How to design strategyproof mechanisms without leveraging monetary transfers is a more complicated problem (see, e.g., Schummer and Vohra [2007] and Procaccia and Tennenholtz [2013]). In the standard mechanism design with money literature, monetary transfers can be employed to effectively eliminate the incentives for agents to misreport their types. On the other hand, in some practical settings the designer might not be allowed to leverage monetary transfers for ethical and legal issues (see Roughgarden [2010]), or due to practical constraints (see Procaccia and Tennenholtz [2013]).

Generalized Assignment Problem with Predictions. We focus on the *generalized assignment problem (GAP)*, which encompasses several fundamental special cases that have been studied in the literature on mechanism design without money (such as matching, multiple knapsack, GAP variants, etc.). In this problem, we are given a set L of strategic agents (or jobs) that can be assigned to a set R of resources (or machines). Each agent $i \in L$ has a value v_{ij} and a size s_{ij} for being assigned to resource $j \in R$. Further, each resource $j \in R$ has a capacity C_j (in terms of total size) that must not be exceeded. The goal of the designer is to compute a feasible assignment of agents to resources such that the overall value is maximized. This problem models several important use cases that naturally arise in applications such as online advertising, crew planning, machine scheduling, etc.

GAP Variant	Restrictions
Unweighted Bipartite Matching (UBM)	$v_{ij} = 1, s_{ij} = 1, C_j = 1$
Bipartite Matching Problem (BMP)	$s_{ij} = 1, C_j = 1$
Value Consensus GAP (VCGAP)	$\exists \sigma : v_{i\sigma(1)} \geq \ldots, \geq v_{i\sigma(m)}$
Agent Value GAP (AVGAP)	$v_{ij} = v_i$
Resource Value GAP (RVGAP)	$v_{ij} = v_j$
Agent Size GAP (ASGAP)	$s_{ij} = s_i$
Resource Size GAP (RSGAP)	$s_{ij} = s_j$
Restricted Multiple Knapsack (RMK)	$v_{ij} = v_i, s_{ij} = s_i$
Equal RMK (ERMK)	$v_{ij} = s_{ij} = v_i$

Table 1. Overview of GAP variants.

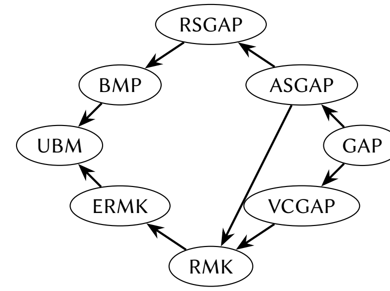

Fig. 1. Taxonomy of GAP variants.

Unfortunately, if the values of the agents are assumed to be private information, it is known that deterministic strategyproof mechanisms are unable to provide bounded approximation guarantees for GAP (see [Dughmi and Ghosh, 2010]).

Private Graph Model. In light of this, we turn towards a slightly more restrictive (but natural) model for GAP that was introduced by Dughmi and Ghosh [2010], called the *private graph model*. Here the agents' values are assumed to be public information, but whether or not the value v_{ij} can be generated by assigning agent i to resource j is private information. The latter can naturally be interpreted as *compatibility restrictions* that agents have with respect to the available resources. Note that this variation restricts the strategy space of the agents from having the ability to misreport their entire valuation vector to being able to misreport only their compatibility vector. Despite this restriction, GAP in the private graph model still has several natural applications (see also [Dughmi and Ghosh, 2010]).

We study GAP in the private graph model considering a learning-augmented setting.[1] Here, we assume that a prediction of the optimal assignment with respect to the private compatibilities is given. Note that this is weaker than assuming that the actual compatibilities are available as a prediction. To see this, note we can always compute an optimal assignment with respect to some given compatibility declarations (notwithstanding computational constraints). Depending on the underlying application, it seems reasonable to assume that such assignment are learnable (e.g., through deep reinforcement learning, graph convolution neural networks, etc.).

1.1 Our Contributions

We initiate the study of GAP in the private graph model with predictions. We assume that a (potentially erroneous) prediction of the optimal assignment for the true compatibility graph is given as part of the input. We derive both deterministic and randomized mechanisms that are (universally) group-strategyproof (GSP) for different variants of GAP; see Table 1 for an overview. Our mechanisms are parameterized by a *confidence parameter* $\gamma \geq 1$, which determines the trade-off between the respective consistency and robustness guarantees. Choosing a higher confidence value leads to a better consistency but a worse robustness guarantee, and vice versa. We also derive more fine-grained approximation guarantees that smoothly interpolate between consistency and robustness, depending on some error parameter $\hat{\eta}$ of the prediction. Here, $1 - \hat{\eta}$ measures the relative gap between the value of the predicted assignment and an optimal one; in particular, $\hat{\eta} = 0$

[1]For the more general model of GAP with private values, even having access to optimal assignment predictions does not help much: it is not hard to see that this leads to unbounded robustness if strategyproofness and bounded consistency is required; more details are given in [Colini-Baldeschi et al., 2024].

if the prediction is perfect, while $\hat{\eta} = 1$ if the prediction is arbitrarily bad. We summarize our main results below.

- We prove a lower bound on the best possible trade-off in terms of consistency and robustness guarantees that is achievable by any deterministic strategyproof mechanism for GAP (Section 3). More precisely, we show that no deterministic strategyproof mechanism can be $(1 + 1/\gamma)$-consistent and $(1 + \gamma - \epsilon)$-robust for any $\epsilon > 0$. In fact, our lower bound holds for the special case of the bipartite matching problem (BMP). We also extend our insights to derive a lower bound in terms of consistency and approximation guarantees.
- For BMP, we derive a deterministic GSP mechanism that is $(1 + 1/\gamma)$-consistent and $(1 + \gamma)$-robust (Section 4). In light of the lower bound above, our mechanism thus achieves the best possible consistency and robustness guarantees, albeit satisfying the stronger notion of GSP. Unlike the mechanism known in the literature for the problem without predictions, we crucially do not consider declarations in a fixed order. Instead, our mechanism draws inspiration from the well-known deferred acceptance algorithm algorithm by Gale and Shapley [1962]. Here, the agent proposal order is crucial for GSP and the resource preference order is crucial to improve upon the known guarantee for the problem without predictions. If an edge is in the predicted optimal matching, it potentially has a better ranking in the resource preference order. The result extends to many-to-one matchings and RSGAP.
- For GAP, we give a deterministic mechanism that greedily adds declared edges (while maintaining feasibility) to an initially empty assignment, according to some order of the declarations (Section 5). This order follows from a specific ranking function that is given as part of the input. We prove a sufficient condition, called *truth-inducing*, of the ranking function for GSP. For the special case of ERMK, we combine this greedy mechanism with a truth-inducing ranking function resulting in a deterministic GSP mechanism that is $(1 + 1/\gamma)$-consistent and $(2 + \gamma)$-robust. The same approach can be used to obtain a GSP 3-approximate mechanism for the setting without predictions, for which no polynomial time deterministic strategyproof mechanism was known prior to this work.
- For ASGAP and VCGAP, we derive randomized universally GSP mechanisms that are $(1 + 3/\gamma)$-consistent and $(3 + \gamma)$-robust (Section 6). To this aim, we randomize over three deterministic mechanisms, consisting of our mechanism for BMP, our greedy mechanism, and a third mechanism that simply follows the prediction. As the previously mentioned greedy mechanism is one of the three building blocks, it is crucial that for these variants there exist truth-inducing ranking functions. Notably, none of the three mechanisms achieves a bounded robustness guarantee by itself. Additionally, note that RMK and both AVGAP and RVGAP are special cases of VCGAP. For these special cases, no polynomial time deterministic strategyproof mechanism is known. Finally, for BMP and ERMK, we derive randomized universally GSP mechanisms that are $(1 + 1/\gamma)$-consistent and outperform the robustness guarantees of their respective deterministic counterparts in expectation. In particular, for BMP this provides a separation result showing that randomized mechanisms are more powerful than deterministic ones (at least in expectation).

1.2 Related Work

Algorithms with predictions represent one perspective within the "beyond worst-case" paradigm. The primary goal is to overcome existing worst-case lower bounds by augmenting each instance with a prediction, possibly a machine-learned one. Hence, this line of work is also sometimes referred to as *learning-augmented algorithms*. The conceptual framework that describes the trade-off between *α-consistency* and *β-robustness* was introduced by Lykouris and Vassilvitskii [2021] in the context of online algorithms. Since then, online algorithms have remained a major focus (see e.g., Azar et al. [2021, 2022], Banerjee et al. [2022], Purohit et al. [2018] for some reference works). Thematically relevant to us, are the works on online matching (e.g., Antoniadis et al. [2023a,b], Dinitz et al. [2022], Jin and Ma [2022], Lavastida et al. [2021a,b]) in non-strategic environments.[2] Other domains that have been studied under the lens of predictions include the reevaluation of runtime guarantees of algorithms (see e.g., Chen et al. [2022], Dinitz et al. [2021], Sakaue and Oki [2022] for bipartite matching algorithms), streaming algorithms, data structures, and more. We refer the reader to Mitzenmacher and Vassilvitskii [2020] for a survey of some of the earlier works. An overview of research articles that appeared on these topics is available at https://algorithms-with-predictions.github.io.

Recently, Xu and Lu [2022] and Agrawal et al. [2023] introduced predictions for settings involving strategic agents. In their work, Xu and Lu [2022] showcased four different mechanism design settings with predictions, both with and without monetary transfers. On the other hand, Agrawal et al. [2023] focused solely on strategic facility location. Most subsequent works with strategic considerations have also been in algorithmic mechanism design (see e.g., Balkanski et al. [2023a,b], Istrate and Bonchis [2022], Prasad et al. [2024]). However, other classic domains of economics and computation literature continue to be revisited in the presence of predictions; see, e.g., the works by Gkatzelis et al. [2022] on the price of anarchy, Berger et al. [2023] on voting, Lu et al. [2023] and Caragiannis and Kalantzis [2024] on auction revenue maximization.

We briefly elaborate on the relation between learning-augmented mechanism design and Bayesian mechanism design. As pointed out by Agrawal et al. [2023], the main difference is the absence of worst-case guarantees in the Bayesian setting. Indeed, in the standard Bayesian setting, it is implicitly assumed that one has perfect knowledge of the distribution when analyzing the expected performance of mechanisms. While this is a reasonable assumption in some settings, Bayesian mechanisms do not offer any guarantees if this assumption fails.

Mechanism design without money has a rich history spanning over fifty years, being deeply rooted in economics and social choice theory. As will be evident in Section 4, the seminal works of Gale and Shapley [1962], Roth [1982] and Hatfield and Milgrom [2005] on stable matching are particularly relevant to our study. However, our work aligns more closely with the agenda of *approximate* mechanism design without money set forth by Procaccia and Tennenholtz [2013] and, in particular, the subsequent work by Dughmi and Ghosh [2010]. In their work, Dughmi and Ghosh [2010] introduced the private graph model, that we use in our environment with predictions, and initiated the study of variants of GAP when the agents are strategic. (A variant of this model where the resources are strategic instead, was studied by Fadaei and Bichler [2017a]). Dughmi and Ghosh [2010] obtained a deterministic 2-approximate strategyproof mechanism for weighted bipartite matching and a matching lower bound. Furthermore, they developed randomized strategyproof-in-expectation[3] mechanisms for special cases of GAP; namely, a 2-approximation for RMK, a 4-approximation for ASGAP and a 4-approximation for a special case of VCGAP, termed *Agent Value GAP* (see Section 6.2 for some discussion). Finally, they proposed a randomized, strategyproof-in-expectation $O(\log n)$-approximate mechanism for the general case. Subsequently, Chen et al. [2014] improved upon these results by devising mechanisms which satisfy universal

[2]An exception to this is the work by Antoniadis et al. [2023b], who, even though their main focus is on designing algorithms for online bipartite matching, observe that their algorithm implies a strategyproof mechanism if monetary transfers are allowed.

[3]A randomized mechanism is strategyproof-in-expectation if the true declaration of an agent maximizes their expected utility. Note that this is a weaker notion than universal strategyproofness, which we obtain for the randomized mechanisms in this work.

strategyproofness, matching the guarantees of Dughmi and Ghosh [2010] for these special cases. Additionally, they showed an improved $O(1)$-approximation for GAP.

Beyond the private graph model, for the setting where values are private information but monetary transfers are allowed, Fadaei and Bichler [2017b] devised a $\frac{e}{e-1}$-approximate mechanism that is strategyproof-in-expectation. Finally, from an algorithmic perspective, the best known approximation ratio for GAP is $\frac{e}{e-1} - \epsilon$, for a fixed small $\epsilon > 0$ due to Feige and Vondrak [2006]. On the negative side, Chakrabarty and Goel [2010] have shown that GAP does not admit an approximation better that $11/10$, unless P=NP.

2 Preliminaries

2.1 Generalized Assignment Problem with Predictions

In the *generalized assignment problem (GAP)*, we are given a bipartite graph $G = (L \cup R, D)$ consisting of a set $L = [n]$ of $n \geq 1$ agents (or items, jobs) and a set $R = [m]$ of $m \geq 1$ resources (or knapsacks, machines, respectively).[4] Each agent $i \in L$ has a value $v_{ij} > 0$ and a size $s_{ij} > 0$ for being assigned to resource $j \in R$. Further, each resource $j \in R$ has a capacity $C_j > 0$ (in terms of total size) that must not be exceeded. We assume without loss of generality that $s_{ij} \leq C_j$ for every $i \in L$. Below, we use $\boldsymbol{v} = (v_{ij})_{i \in L, j \in R} \in \mathbb{R}^{n \times m}_{>0}$ to refer to the matrix of all agent-resource values, $\boldsymbol{s} = (s_{ij})_{i \in L, j \in R} \in \mathbb{R}^{n \times m}_{>0}$ to refer to the matrix of all agent-resource sizes and $\boldsymbol{C} = (C_j)_{j \in R} \in \mathbb{R}^{m}_{>0}$ to refer to the vector of all resource capacities.[5]

The bipartite graph $G = (L \cup R, D)$ encodes compatibilities between agents and resources; we also refer to it as the *compatibility graph*.[6] An agent $i \in L$ is said to be *compatible* with a resource $j \in R$ if $(i, j) \in D$; otherwise, i is *incompatible* with j. We use $D_i = \{(i, j) \in D\}$ to denote the set of all compatible edges of agent i. Similarly, we use $D_j = \{(i, j) \in D\}$ to refer the set of all compatible edges of resource j. For example, the compatibility $(i, j) \in D$ might indicate that agent i has access to resource j, or that item i can be assigned to knapsack j, or that job i can be executed on machine j. Generally, D can be any subset of $L \times R$. We use $G[D] = (L \cup R, D)$ to refer to the compatibility graph induced by the edge set $D \subseteq L \times R$ and $\mathcal{I}_{\text{GAP}} = (G[D], \boldsymbol{v}, \boldsymbol{s}, \boldsymbol{C})$ to refer to an instance of GAP.

An *assignment* $M \subseteq L \times R$ is a subset of edges such that each agent $i \in L$ is incident to at most one edge in M. Note that each agent is assigned to at most one resource, but several agents might be assigned to the same resource; we also say that M is a *many-to-one assignment.* If we additionally require that each resource $j \in R$ is incident to at most one edge in M, then M is said to be a *one-to-one assignment* (or, *matching*, simply). Note that every matching is also an assignment. Given an agent $i \in L$, we use $M(i) = \{j \in R \mid (i, j) \in M\}$ to refer to the resource assigned to i (if any); note that $M(i)$ is a singleton set. Also, we have $M(i) = \emptyset$ if i is unassigned. Similarly, for a resource $j \in R$, we define $M(j) = \{i \in L \mid (i, j) \in M\}$ as the set of agents assigned to j (if any); note that $M(j) = \emptyset$ if j is unassigned.

An assignment M is said to be *feasible* for a given compatibility graph $G[D]$ if (1) M is an assignment in $G[D]$, i.e., $M \subseteq D$, and (2) M satisfies all resource capacities constraints, i.e., for each resource $j \in R$, $\sum_{i \in M(j)} s_{ij} \leq C_j$. We define the *value* $v(M)$ of an assignment M as the sum of the values of all edges in M; more formally,

$$v(M) = \sum_{(i,j) \in M} v_{ij}. \tag{1}$$

[4]Throughout the paper, we use $[n] = \{1, \ldots, n\}$ to refer to the set of the first $n \geq 1$ natural numbers.

[5]We note that our assumption of all values, sizes and capacities being positive is without loss of generality in our model.

[6]We can assume without loss of generality that G does not contain any isolated nodes.

We overload this notation slightly and also write $v(M(i))$ and $v(M(j))$ to refer to the total value of all edges assigned to agent i or resource j, respectively. We use M_D^* to denote a feasible assignment of maximum value in the graph $G[D]$. We also say that M_D^* is an *optimal* assignment with respect to D.

As we consider GAP with assignment predictions, in addition to an instance $\mathcal{I}_{\text{GAP}} = (G[D], \boldsymbol{v}, \boldsymbol{s}, \boldsymbol{C})$, we are given a *predicted assignment* $\hat{M} \subseteq L \times R$ that respects the capacity constraints, i.e., for all $j \in R$ it holds that $\sum_{(i,j)\in\hat{M}} s_{ij} \leq C_j$. Generally, $\hat{M}$ can be any assignment in the complete graph $G[L \times R]$, and thus may contain edges which are not in D. It is important to realize that the predicted assignment $\hat{M}$ is considered to be part of the input. We use $\mathcal{I}_{\text{GAP}^+} = (G[D], \boldsymbol{v}, \boldsymbol{s}, \boldsymbol{C}, \hat{M})$ to refer to an instance of GAP augmented with an assignment prediction $\hat{M}$. We use a similar notation for the various special cases of GAP we study (see Table 1 for an overview of the special cases).

We say that $\hat{M}$ is a *perfect prediction* for $\mathcal{I}_{\text{GAP}^+}$ if it corresponds to an assignment of maximum value in the graph $G[D]$, i.e., $v(\hat{M} \cap D) = v(M_D^*)$. Namely, we define an error parameter that measures the quality of the predicted assignment $\hat{M}$ relative to an optimal assignment M_D^* of $G[D]$. We define the *prediction error* $\eta(\mathcal{I}_{\text{GAP}^+}) \in [0, 1]$ of an instance $\mathcal{I}_{\text{GAP}^+} = (G[D], \boldsymbol{v}, \boldsymbol{s}, \boldsymbol{C}, \hat{M})$ as

$$\eta(\mathcal{I}_{\text{GAP}^+}) = 1 - \frac{v(\hat{M} \cap D)}{v(M_D^*)}. \tag{2}$$

Note that with this definition an instance $\mathcal{I}_{\text{GAP}^+}$ with a perfect prediction has a prediction error of 0. As the value of $\hat{M} \cap D$ deteriorates from the value of an optimal assignment, the error measure approaches 1. If $\eta(\mathcal{I}_{\text{GAP}^+}) = 1$, we must have $v(\hat{M} \cap D) = 0$ which means that the prediction $\hat{M}$ does not contain any edge that is also in D (recall that the values are assumed to be positive).

Note that the definition of our error parameter in (2) is meaningful as it captures the relative gap between the *values* of the predicted assignment and the optimal one. Alternatively, one could compare structural properties of $\hat{M} \cap D$ and M_D^*. However, this seems less suitable in our context: For example, under an error notion that is not value-based, a predicted assignment may only miss one edge of an optimal assignment, i.e., $|M_D^* \setminus (\hat{M} \cap D)| = 1$, but still be of relatively low value if this missing edge is valuable. Further, a predicted assignment might contain none of the edges of the optimal assignment, i.e., $(\hat{M} \cap D) \cap M_D^* = \emptyset$, but still be very useful when its value is close to optimal; in fact, $(\hat{M} \cap D)$ might even be an optimal matching that is disjoint from M_D^* (because the optimal assignment might not be unique). Finally, note that accounting the value of edges in $\hat{M} \setminus D$ in a prediction error notion is not informative, as our goal is to compute a feasible assignment M (i.e., $M \subseteq D$). All these cases are captured by the definition of our prediction error as in (2).

Given a fixed error parameter $\hat{\eta} \in [0, 1]$, instances $\mathcal{I}_{\text{GAP}^+} = (G[D], \boldsymbol{v}, \boldsymbol{s}, \boldsymbol{C}, \hat{M})$ with $\eta(\mathcal{I}_{\text{GAP}^+}) \leq \hat{\eta}$ constitute the class of instances of prediction error at most $\hat{\eta}$.

Approximation Objectives. We introduce the following three approximation notions for GAP with predictions.

- **Consistency:** A mechanism $\mathcal{M}$ is α*-consistent* with $\alpha \geq 1$ if for every instance $\mathcal{I}_{\text{GAP}^+} = (G[D], \boldsymbol{v}, \boldsymbol{s}, \boldsymbol{C}, \hat{M})$ with a perfect prediction, the computed matching $M = \mathcal{M}(D)$ satisfies $\alpha \cdot v(M) \geq v(M_D^*)$.
- **Robustness:** A mechanism $\mathcal{M}$ is β*-robust* with $\beta \geq 1$ if for every instance $\mathcal{I}_{\text{GAP}^+} = (G[D], \boldsymbol{v}, \boldsymbol{s}, \boldsymbol{C}, \hat{M})$ with an arbitrary prediction $\hat{M}$, the computed matching $M = \mathcal{M}(D)$ satisfies $\beta \cdot v(M) \geq v(M_D^*)$.
- **Approximation:** A mechanism $\mathcal{M}$ is $g(\hat{\eta})$*-approximate* with $g(\hat{\eta}) \geq 1$ if for every instance $\mathcal{I}_{\text{GAP}^+} = (G[D], \boldsymbol{v}, \boldsymbol{s}, \boldsymbol{C}, \hat{M})$ of prediction error at most $\hat{\eta} \in [0, 1]$, the computed matching $M = \mathcal{M}(D)$ satisfies $g(\hat{\eta}) \cdot v(M) \geq v(M_D^*)$.

2.2 Private Graph Model and Assignment Mechanisms

We study GAP with predictions in a strategic environment. More specifically, we are interested in the setting where the agents are strategic and might misreport their actual compatibilities. To this aim, we use the *private graph model* introduced by Dughmi and Ghosh [2010] to capture such manipulations in a meaningful way.

Here, each agent $i \in L$ has a private compatibility set $E_i \subseteq \{(i, j) \in L \times R\}$ specifying the set of edges that are truly compatible for i. Crucially, the compatibility set E_i is *private* information, i.e., E_i is only known to agent i. In addition, each agent $i \in L$ declares a public compatibility set $D_i \subseteq \{(i, j) \mid j \in R\}$. The interpretation is that i claims to be compatible with resource $j \in R$ if and only if $(i, j) \in D_i$; but these declarations might not be truthful, i.e., $D_i \neq E_i$. We define $D = \cup_{i \in L} D_i \subseteq L \times R$ to refer to the union of all compatibility sets declared by the agents. We use $G[D] = (L \cup R, D)$ to refer to the compatibility graph induced by the declared edges in D. Similarly, we use $G[E]$ to refer to the compatibility graph induced by the true compatibility sets of the agents, i.e., $E = \cup_{i \in L} E_i$. We refer to $G[E]$ as the *private graph model* (or, *private graph* simply).

Subsequently, we use $(\mathcal{I}_{\text{GAP}^+}, G[E])$ to refer to an instance $\mathcal{I}_{\text{GAP}^+}$ of GAP with predictions in the private graph model $G[E]$. We note that all input data of $\mathcal{I}_{\text{GAP}^+} = (G[D], \boldsymbol{v}, \boldsymbol{s}, \boldsymbol{C}, \hat{M})$ is public information accessible by the mechanism, while the private graph $G[E]$ is private information. For the sake of conciseness, we often omit input parameters which remain fixed; in fact, most of the time is will be sufficient to refer explicitly to the compatibility declarations D only.

Given an instance $(\mathcal{I}_{\text{GAP}}, G[E])$ with compatibility declarations D, a deterministic mechanism $\mathcal{M}$ computes an assignment $M = \mathcal{M}(D)$ that is feasible for D. The *utility* u_i of agent $i \in L$ is defined as

$$u_i(D) = \begin{cases} v_{ij} & \text{if } (i, j) \in M \cap E_i, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Note that the utility of agent i is v_{ij} if i is assigned to resource j in M and i is truly compatible with resource j. In particular, the utility of i is 0 if i is unassigned in M, or if i is matched to an incompatible resource. We assume that each agent wants to maximize their utility. To this aim, an agent i might misreport their true compatibilities by declaring a compatibility set $D_i \neq E_i$. Note that we are considering a multi-parameter mechanism design problem here.

Note that if $M = \mathcal{M}(D)$ is the assignment computed by $\mathcal{M}$ for truthfully declared compatibilities, i.e., $D = E$, then its value $v(M)$ (as defined in (1)) is equal to the sum of the utilities of the agents.

Incentive Compatibility Objectives. The following incentive compatibility notions will be relevant in this paper.

- **Strategyproofness:** A mechanism $\mathcal{M}$ is *strategyproof* if for every instance $\mathcal{I}_{\text{GAP}^+} = (G[D], \boldsymbol{v}, \boldsymbol{s}, \boldsymbol{C}, \hat{M})$ and private graph $G[E]$, it holds that for each agent $i \in L$

$$\forall D'_i : \qquad u_i(E_i, D_{-i}) \geq u_i(D'_i, D_{-i}).$$

- **Group-Strategyproofness:** A mechanism $\mathcal{M}$ is *group-strategyproof* if for every instance $\mathcal{I}_{\text{GAP}^+} = (G[D], \boldsymbol{v}, \boldsymbol{s}, \boldsymbol{C}, \hat{M})$ and private graph $G[E]$, it holds that for every subset $S \subseteq L$

$$\forall D'_S : \exists i \in S \qquad u_i(E_S, D_{-S}) \geq u_i(D'_S, D_{-S}).$$

Randomized Mechanisms. A *randomized mechanism* is a probability distribution over a finite set of deterministic mechanisms. By extension, given a randomized mechanism $\mathcal{M}$ and an instance of GAP with predictions $\mathcal{I}_{\text{GAP}^+}$, $\mathcal{M}(\mathcal{I}_{\text{GAP}^+})$ is a probability distribution over a finite set of feasible assignments for $\mathcal{I}_{\text{GAP}^+}$.

All randomized mechanisms we suggest in this work are universally strategyproof. A randomized mechanism $\mathcal{M}$ is *universally strategyproof* if it is a probability distribution over a finite set

MECHANISM 1: TRUST($\mathcal{I}_{\text{GAP}^+}$)

Input: An instance $\mathcal{I}_{\text{GAP}^+} = (G[D], \boldsymbol{v}, \boldsymbol{s}, \boldsymbol{C}, \hat{M})$.
Output: A feasible assignment for $\mathcal{I}_{\text{GAP}^+}$.
return $\hat{M} \cap D$

of deterministic strategyproof mechanisms. The notion of *universally group-strategyproofness* is defined analogously. The three approximation objectives defined in Section 2.1 extend naturally to randomized mechanisms (simply by replacing the value of an assignment with the expected value in the respective definition).

Due to space limitations, several proofs and additional material are deferred to the full version of the paper, see Colini-Baldeschi et al. [2024].

3 Impossibility Results and the Baseline Mechanism

We prove a lower bound on the best possible trade-off in terms of consistency and robustness guarantees achievable by any deterministic strategyproof mechanism. We also derive a lower bound in terms of the error parameter η. Finally, we introduce a trivial mechanism, called TRUST, that serves as a baseline mechanism in subsequent sections.

Impossibility Results. We prove our lower bound for the bipartite matching problem (BMP$^+$) and the Value Consensus GAP (VCGAP$^+$) with predictions in the private graph model. Clearly, this lower bound extends to all variants of GAP$^+$ that contain BMP$^+$ or VCGAP$^+$ as a special case.

THEOREM 3.1. *Let $\gamma \geq 1$ be fixed arbitrarily. Then no deterministic strategyproof mechanism for BMP$^+$ or VCGAP$^+$ can achieve $(1 + 1/\gamma)$-consistency and $(1 + \gamma - \epsilon)$-robustness for any $\epsilon > 0$.*

Note that the lower bound holds independently of any computational assumptions. For the setting without predictions, Dughmi and Ghosh [2010] proved a lower bound of 2 for BMP.

Theorem 3.1 proves a lower bound in terms of consistency versus robustness. The next theorem establishes a lower bound in terms of the error parameter η as defined in (2) for any deterministic strategyproof mechanism that is $(1 + 1/\gamma)$-consistent.

THEOREM 3.2. *Let $\gamma \geq 1$ be fixed arbitrarily. Then no deterministic strategyproof mechanism for BMP$^+$ or VCGAP$^+$ can be $(1 + 1/\gamma)$-consistent and $(\frac{1}{1-\eta+\epsilon})$-approximate with $\epsilon > 0$ for any $\eta < \gamma/1+\gamma$.*

Baseline Mechanism. We conclude this section by introducing a naïve mechanism for GAP$^+$ in the private graph model, which simply adheres to the prediction: Given an instance $\mathcal{I}_{\text{GAP}^+} = (G[D], \boldsymbol{v}, \boldsymbol{s}, \boldsymbol{C}, \hat{M})$, the mechanism returns the assignment $\hat{M} \cap D$. We call this mechanism TRUST (see Mechanism 1). It is trivial to see that TRUST is 1-consistent. It is also not hard to prove that TRUST is group-strategyproof and achieves an optimal approximation guarantee matching the lower bound in Theorem 3.2 (as $\gamma \to \infty$).

THEOREM 3.3. *Fix some error parameter $\hat{\eta} \in [0, 1)$. Consider the class of instances of GAP$^+$ in the private graph model with prediction error at most $\hat{\eta}$. Then, TRUST is group-strategyproof and achieves an optimal approximation guarantee of $1/(1 - \hat{\eta})$.*

We conclude from Theorem 3.3 that TRUST realizes our strongest notion of incentive compatibility (i.e., group-strategyproofness) and even achieves the best possible consistency and approximation guarantees. But the point is, that it completely fails to achieve any bounded robustness guarantee.[7] In a nutshell, this demonstrates that the actual challenge in deriving strategyproof mechanisms

[7]To see this, just consider an instance $\mathcal{I}_{\text{GAP}^+}$ with $\hat{M} \cap D = \emptyset$.

MECHANISM 2: Boost($\mathcal{I}_{\text{BMP}^+}, \gamma$)

Input: An instance $\mathcal{I}_{\text{BMP}^+} = (G[D], \boldsymbol{v}, \hat{M})$, confidence parameter $\gamma \geq 1$.
Output: A feasible matching M for $\mathcal{I}_{\text{BMP}^+}$.
Initialize $M = \emptyset, A = L, P_i = D_i$ for each $i \in L$.
while $A \neq \emptyset$ **do**
  Choose $i \in A$ and let $(i, j) = \arg\max\{v_{ij} \mid (i, j) \in P_i\}$. // determine next proposal (i, j)
  Agent i offers $\theta_{ij} = \theta_{ij}(\gamma, \hat{M})$ to resource j. // i makes (boosted) offer to j
  if $\theta_{ij} > \theta_{M(j)j}$ **then** // check if i's offer is highest for j
    if $M(j) \neq \emptyset$ **then** $M = M \setminus \{(M(j), j)\}$ // j rejects current mate $M(j)$ (if any)
    $M = M \cup \{(i, j)\}$ // i tentatively matched to j
    $A = A \cup M(j) \setminus \{i\}$ // update active agents
  $P_i = P_i \setminus \{(i, j)\}$ // update i's proposal set
  if $P_i = \emptyset$ **then** $A = A \setminus \{i\}$ // remove i from A if no more proposals
return M

for GAP$^+$ in the private graph model is to achieve the best possible trade-off in terms of consistency/approximation *and* robustness guarantees; without the latter, the whole problem becomes trivial (as Trust is the best possible mechanism). Despite this deficiency, and perhaps surprisingly, we will use this non-robust mechanism Trust as an important building block in our randomized mechanisms described in Section 6.

4 Our Mechanism for Bipartite Matching with Predictions

We introduce our mechanism, called Boost, for bipartite matching with predictions (BMP$^+$) in the private graph model. Our mechanism is inspired by the *deferred acceptance algorithm* by Gale and Shapley [1962]. Boost is parameterized by some $\gamma \geq 1$, which we term the *confidence parameter*. Put differently, Boost defines an (infinite) family of deterministic mechanisms, one for each choice of $\gamma \geq 1$. It is important to realize that all properties proved below, hold for an arbitrary choice of $\gamma \geq 1$. Boost also constitutes an important building block to derive our randomized mechanisms for special cases of GAP in Section 6.

4.1 Boost Mechanism

Boost receives as input an instance $\mathcal{I}_{\text{BMP}^+} = (G[D], \boldsymbol{v}, \hat{M})$ of BMP$^+$ and a confidence parameter $\gamma \geq 1$. Our mechanism maintains a tentative matching M and a subset of agents $A \subseteq L$ that are called *active*. An agent i is *active* if it is not tentatively matched to any resource and has some remaining proposal to make; otherwise, i is *inactive*. Initially, the matching is empty, i.e., $M = \emptyset$, and all agents are active, i.e., $A = L$. In each iteration, the mechanism chooses an active agent $i \in A$ who then makes an offer to an adjacent resource $j \in R$ by following a specific proposal order:

Agent Proposal Order: Each agent $i \in L$ maintains an order on their set of incident edges $D_i = \{(i, j) \in D\}$ by sorting them according to non-increasing values v_{ij}. We assume that ties are resolved according to a fixed tie-breaking rule τ_i.[8]

The key idea behind our mechanism is that the value v_{ij} that i proposes to j is *boosted* if the edge (i, j) is part of the predicted matching $\hat{M}$. We make this idea more concrete: given an agent i

[8]Note that the choice of the edge (i, j) of maximum value v_{ij} in Line 2 is uniquely determined by this order.

and a declared edge $(i, j) \in D_i$, we define the *offer* $\theta_{ij} = \theta_{ij}(\gamma, \hat{M})$[9] for resource j as

$$\theta_{ij}(\gamma, \hat{M}) = \begin{cases} v_{ij} & \text{if } (i, j) \notin \hat{M}, \\ \gamma v_{ij} & \text{if } (i, j) \in \hat{M}. \end{cases} \tag{4}$$

Based on this definition, when it is i's turn to propose to resource j, then the offer that j receives from i is the actual value v_{ij} if (i, j) is not a predicted edge, while it is the boosted value γv_{ij} if (i, j) is a predicted edge. Intuitively, our mechanisms increases the chance that an agent proposing through a predicted edge is accepted (see below) by amplifying the offered value by a factor $\gamma \geq 1$.

Suppose resource j receives offer θ_{ij} from agent i. Then j *accepts* i if θ_{ij} is the largest offer that j received so far; otherwise, j *rejects* i. We define $\theta_{\emptyset j} = 0$ to indicate that the highest offer that j received is zero if j is still unmatched, i.e., $M(j) = \emptyset$. To this aim, each resource j maintains a fixed preference order over their set of incident edges.

Resource Preference Order: Each resource j maintains an order on their set of incident edges $D_j = \{(i, j) \in D\}$ by sorting them according to non-increasing offer values θ_{ij}. We assume that ties are resolved according to a fixed tie-breaking rule τ_j.[10]

If i is accepted, then i becomes *tentatively matched* to j, i.e., (i, j) is added to M, and i becomes inactive. Also, if there is some agent k that was tentatively matched to j before, then k is rejected by j, i.e., (k, j) is removed from M, and k becomes active again. Whenever an agent gets rejected, it moves on to the next proposal (if any) according to their offer order; in particular, an agent proposes at most once to each adjacent resource.

The mechanism terminates when all agents are inactive, i.e., $A = \emptyset$. The current matching becomes definite and is output by the mechanism. Note that we do not specify how an agent i is chosen from the set of active agents A in Line 2. In fact, any choice will work here.[11] For example, a natural choice is to always choose an active agent $i \in A$ whose next offer θ_{ij} is largest.

Intuitively, the confidence parameter $\gamma \geq 1$ specifies to which extent Boost follows the prediction. On the one extreme, for $\gamma = 1$ our mechanism ignores the prediction, which is the best choice in terms of achieving optimal robustness (at the expense of achieving worst consistency). As γ increases, our mechanism follows the prediction more and more. On the other extreme, for $\gamma \to \infty$ our mechanism becomes Trust (as introduced in Section 3) and simply returns the predicted matching; naturally, this is the best choice in terms of achieving optimal consistency (at the expense of unbounded robustness).

A crucial difference between Boost and the existing mechanism for BMP by Dughmi and Ghosh [2010], is that Boost maintains a tentative matching until it terminates. The 2-approximate mechanism of Dughmi and Ghosh [2010] greedily and permanently matches declared edges according to non-increasing values, while maintaining feasibility. This mechanism coincides with Boost if $\gamma = 1$, i.e., Boost ignores the prediction, and the next proposal is determined by the highest valued remaining edge among the active agents (assuming that ties are broken in the same way).

The following is the main result of this section:

Theorem 4.1. *Fix some error parameter $\hat{\eta} \in [0, 1]$. Consider the class of instances of BMP$^+$ in the private graph model with prediction error at most $\hat{\eta}$. Then, for every confidence parameter $\gamma \geq 1$,*

[9]Note that $\theta_{ij}(\gamma, \hat{M})$ depends on the confidence parameter γ and the predicted matching $\hat{M}$; but we omit these arguments and simply write θ_{ij} if they are clear from the context.

[10]Note that the comparison in Line 2 is done with respect to this order.

[11]In contrast, for the more general settings considered below, the specific choice of $i \in A$ will be crucial.

Boost is group-strategyproof and has an approximation guarantee of

$$g(\hat{\eta}, \gamma) = \begin{cases} \frac{1+\gamma}{\gamma(1-\hat{\eta})} & \text{if } \hat{\eta} \leq 1 - 1/\gamma, \\ 1+\gamma & \text{otherwise.} \end{cases} \tag{5}$$

In particular, Boost is $(1 + 1/\gamma)$-consistent and $(1 + \gamma)$-robust, which is best possible.

Note that Boost is not only strategyproof, but satisfies the stronger incentive compatibility notion of group-strategyproofness. Also, in light of the lower bound given in Theorem 3.1, Boost achieves the best possible trade-off in terms of consistency and robustness guarantees. Note that the approximation guarantee retrieves the best possible $(1 + 1/\gamma)$-consistency guarantee for $\hat{\eta} = 0$ and $(1 + \gamma)$-robustness guarantee for $\hat{\eta} \geq 1 - 1/\gamma$. For the range $\hat{\eta} \in [0, 1 - 1/\gamma]$, as $\hat{\eta}$ increases the approximation interpolates between the consistency and the robustness guarantee (see Figure 2a). The upper bound for Boost as stated above is off by a factor of $1 + 1/\gamma$ from the lower bound proven in Theorem 3.2 (see Figure 2b and Figure 2c).

The following lemma will turn out to be useful for proving the robustness guarantee.

Lemma 4.2. *Let $\gamma \geq 1$. Let $\mathcal{I}_{BMP^+} = (G[D], \boldsymbol{v}, \hat{M})$ be an instance of BMP$^+$ and let M be the matching returned by Boost$(\mathcal{I}_{BMP^+}, \gamma)$. Then $2v(M) + (\gamma - 1)v(M \cap \hat{M}) \geq v(M^*_D)$.*

We can now prove that Boost is $(1 + \gamma)$-robust.

Proof of Theorem 4.1 (robustness). Let $\gamma \geq 1$. Let $\mathcal{I}_{\text{BMP}^+} = (G[D], \boldsymbol{v}, \hat{M})$ be an instance of BMP$^+$ and let M be the matching returned by Boost$(\mathcal{I}_{\text{BMP}^+}, \gamma)$. Further, let M^*_D be an optimal matching. By Lemma 4.2, we have

$$v(M^*_D) \leq 2v(M) + (\gamma - 1)v(M \cap \hat{M}) \leq 2v(M) + (\gamma - 1)v(M) \leq (1 + \gamma)v(M). \qquad \square$$

The next lemma shows that the matching computed by Boost is a $(1 + 1/\gamma)$-approximation of the predicted matching $\hat{M} \cap D$ in $G[D]$. It will turn out to be useful when proving the approximation guarantee of Boost.

Lemma 4.3. *Let $\gamma \geq 1$. Let $\mathcal{I}_{BMP^+} = (G[D], \boldsymbol{v}, \hat{M})$ be an instance of BMP$^+$ and let M be the matching returned by Boost$(\mathcal{I}_{BMP^+}, \gamma)$. Then $(1 + 1/\gamma)v(M) \geq v(\hat{M} \cap D)$.*

Proof. We prove that the value of each edge in $\hat{M} \cap D$ can be covered by the value of an edge in the matching M output by Boost $(\mathcal{I}_{\text{BMP}^+}, \gamma)$. More precisely, we define a mapping $g : \hat{M} \cap D \longrightarrow M$ together with some scalars $(\alpha_e)_{e \in \hat{M} \cap D}$ such that for each edge $e \in \hat{M} \cap D$ it holds that $\alpha_e \cdot v_e \leq v_{g(e)}$ with $\alpha_e \geq 1$. We also say that e is $(1/\alpha_e)$-*covered* by edge $g(e) \in M$.

Let $e = (i, j) \in \hat{M} \cap D$. If $e \in M$, we define $g(e) = e$ and $\alpha_e = 1$. Suppose $e = (i, j) \notin M$. We distinguish the following cases:

(1) $\exists k \in M(i)$ with $v_{ik} \geq v_{ij}$. We define $g(e) = (i, k)$ and $\alpha_e = 1$.
(2) $\exists k \in M(i)$ with $v_{ik} < v_{ij}$. Note that i first proposes to j and only later to k. In particular, j must have rejected the offer of i at some stage. Thus, there must be an agent ℓ with $(\ell, j) \in M$ whose offer is larger than the one of i. Recall that $(i, j) \in \hat{M} \cap D$ and thus i made a boosted offer γv_{ij} to j. On the other hand, $(\ell, j) \notin \hat{M} \cap D$ and thus ℓ offers $v_{\ell j}$ to j. We conclude that $v_{\ell j} \geq \gamma v_{ij}$. We define $g(e) = (\ell, j)$ and $\alpha_e = \gamma$.
(3) $\nexists k \in M(i)$. Note that i proposed to j at some stage but was rejected (immediately or subsequently) and remained unassigned after all. Similarly to the previous case, this implies that there exists some agent ℓ with $(\ell, j) \in M$ and $v_{\ell j} \geq \gamma v_{ij}$. We define $g(e) = (\ell, j)$ and $\alpha_e = \gamma$.

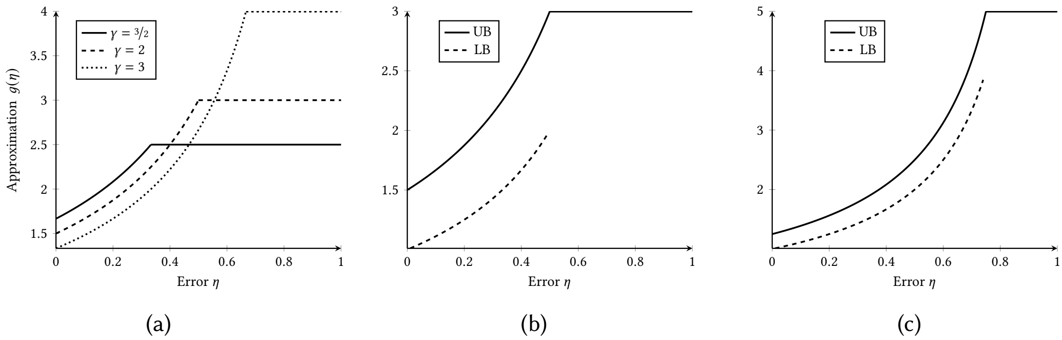

Fig. 2. Approximation guarantee $g(\eta)$ as a function of η. (a) For $\gamma \in \{3/2, 2, 3\}$. (b) Upper vs. lower bound for $\gamma = 2$. (c) Upper vs. lower bound for $\gamma = 4$.

Note that the mapping g defined above maps each edge $e \in \hat{M} \cap D$ either to itself, i.e., $g(e) = e \in M$, or to an edge $f = g(e) \in M$ that is adjacent to e. Also, because $\hat{M} \cap D$ is a matching, there are at most two edges in $\hat{M} \cap D$ which are adjacent to an edge f in M. Said differently, each edge $f \in M$ covers at most two edges in $\hat{M} \cap D$. Moreover, if edge $f = (\ell, j) = g(e) \in M$ $(1/\gamma)$-covers an edge $e = (i, j) \in \hat{M} \cap D$ (i.e., Cases (2) and (3) above), f and e must share a common resource j; in particular, the other edge in $\hat{M} \cap D$ that is mapped to f (if any) must be 1-covered by f. Using the above observations, we can now prove the claim:

$$v(\hat{M} \cap D) = \sum_{e \in \hat{M} \cap D} v_e \leq \sum_{e \in \hat{M} \cap D} v_{g(e)}/\alpha_e \leq \left(1 + \frac{1}{\gamma}\right) \sum_{f \in M} v_f = \left(1 + \frac{1}{\gamma}\right) v(M). \qquad \square$$

We can now complete the proof of Theorem 4.1.

Proof of Theorem 4.1 (Approximation). Let $\gamma \geq 1$ be fixed arbitrarily. Consider an instance $\mathcal{I}_{\text{BMP}^+} = (G[D], \boldsymbol{v}, \hat{M})$ of BMP^+ with prediction error $\eta(\mathcal{I}_{\text{BMP}^+}) \leq \hat{\eta}$. Let M be the matching returned by $\text{Boost}(\mathcal{I}_{\text{BMP}^+}, \gamma)$. Note that by Lemma 4.3 we have $(1 + 1/\gamma) v(M) \geq v(\hat{M} \cap D)$. By the definition of the error parameter in (2), we have $v(\hat{M} \cap D) = (1 - \eta(\mathcal{I}_{\text{GAP}^+})) v(M_D^*)$. Also, $\eta(\mathcal{I}_{\text{GAP}^+})) \leq \hat{\eta}$. Thus,

$$(1 + 1/\gamma) v(M) \geq v(\hat{M} \cap D) = (1 - \eta(\mathcal{I}_{\text{GAP}^+})) v(M_D^*) \geq (1 - \hat{\eta}) v(M_D^*),$$

and we conclude that Boost is $(1 + 1/\gamma)/(1 - \hat{\eta})$-approximate. Further, the robustness guarantee of $(1 + \gamma)$ holds independently of the prediction error $\hat{\eta}$. The claimed bound on the approximation guarantee $g(\hat{\eta}, \gamma)$ now follows by combining these two bounds. $\square$

4.2 Extensions of Boost

Boost is rather versatile in the sense that it can be adapted to handle more general settings while retaining its group-strategyproofness property. We summarize a few extensions below and defer further details to the full version of the paper: (E1) Boost can also be run with a prediction that is a many-to-one assignment and remain group-strategyproof. We exploit this in Section 6. In fact, the only change is that the offer function in (4) is defined with respect to a predicted many-to-one assignment $\hat{M}$. The proof of group-strategyproofness in Theorem 4.1 continues to hold without change. (E2) Boost can also handle many-to-one matchings. Also here, the offer function in (4) is defined with respect to a predicted many-to-one assignment $\hat{M}$. Further, each resource j now accepts the at most C_j highest offers among the set of proposing agents and rejects the remaining

MECHANISM 3: $\textsc{Greedy}(\mathcal{I}_{\mathrm{GAP}^+}, \boldsymbol{z})$

Input: An instance $\mathcal{I}_{\mathrm{GAP}^+} = (G[D], \boldsymbol{v}, \boldsymbol{s}, \boldsymbol{C}, \hat{M})$, a ranking function $\boldsymbol{z} : L \times R \mapsto \mathbb{R}^k$ for some $k \in \mathbb{N}$.
Output: A feasible assignment for $\mathcal{I}_{\mathrm{GAP}^+}$.

```
Set 𝓛 = sort(D, (z(e))_{e∈D}).                   // sort edges in D lexicographically by (z(e))_{e∈D}
Initialize M = ∅.
while 𝓛 ≠ ⟨⟩ do                                  // process edges in sorted order of 𝓛
    Let (i, j) be the first edge of 𝓛 and remove it.   // remove first edge (i, j) from 𝓛
    if ∑_{t∈M(j)∪{i}} s_{tj} ≤ C_j then           // i can be added to M(j) without exceeding capacity
        Set M(j) = M(j) ∪ {i}.                    // add i to M(j)
        Remove all edges of agent i from 𝓛.       // update 𝓛
return M
```

ones.[12] The resulting adaptation of Boost remains group-strategyproof. An easy way to see this by realizing that this adapted mechanisms mimics Boost on the instance obtained from the reduction described next. (E3) Boost can also be used to handle instances of RSGAP by a simple reduction to BMP. Recall that for an instance $\mathcal{I}_{\mathrm{RSGAP}} = (G[D], \boldsymbol{v}, \boldsymbol{s}, \boldsymbol{C})$ of RSGAP it holds that all agents have the same size s_j with respect to a resource j, i.e., $s_{ij} = s_j$ for all $i \in L$. It is not hard to see that we can reduce $\mathcal{I}_{\mathrm{RSGAP}}$ to an equivalent instance $\mathcal{I}_{\mathrm{BMP}}$ of BMP by introducing $\lfloor C_j/s_j \rfloor$ copies of j. Based on this, there is a natural correspondence between the compatibility declarations as well as the assignments in $\mathcal{I}_{\mathrm{RSGAP}}$ and $\mathcal{I}_{\mathrm{BMP}}$, respectively.

The latter observation leads to the following corollary.

Corollary 4.4. *Fix some error parameter $\hat{\eta} \in [0, 1]$. Consider the class of instances of RSGAP$^+$ in the private graph model with prediction error at most $\hat{\eta}$. Then, for any confidence parameter $\gamma \geq 1$, Boost is group-strategyproof and has an approximation guarantee of*

$$g(\hat{\eta}, \gamma) = \begin{cases} \frac{1+\gamma}{\gamma(1-\hat{\eta})} & \text{if } \hat{\eta} \leq 1 - 1/\gamma, \\ 1 + \gamma & \text{otherwise.} \end{cases}$$

Note that the above implies that Boost is group-strategyproof and 2-approximate for RSGAP without predictions (i.e., by choosing $\gamma = 1$). To the best of our knowledge, the current best mechanism for this problem is the randomized, universally strategyproof, 4-approximate mechanism by Chen et al. [2014].

5 Beyond Bipartite Matching Via Greedy Mechanisms

In this section, we first introduce a generic mechanism design template for GAP$^+$ that provides a unifying building block for several of our mechanisms. After that, we provide a first application of this template and derive a deterministic mechanism for a special case of RMK$^+$.

5.1 A Template Of Greedy Mechanisms

At high level, the greedy mechanism template behaves as follows: the mechanism first orders all declared edges according to some specific ranking (which is given as part of the input). According to this order, the mechanism then greedily adds as many edges as possible (while maintaining feasibility) to construct an assignment. We refer to this mechanism as Greedy; see Mechanism 3.

Greedy receives as input an instance $\mathcal{I}_{\mathrm{GAP}^+} = (G[D], \boldsymbol{v}, \boldsymbol{s}, \boldsymbol{C}, \hat{M})$ of GAP$^+$ and a *ranking function* $\boldsymbol{z} : L \times R \mapsto \mathbb{R}^k$ for some $k \in \mathbb{N}$. It then uses the sort operator (see Line 1) to sort the set of declared edges D in lexicographic decreasing order according to their values $(z_1, \ldots, z_k)$. As a result, the list

[12]Recall that in the many-to-one setting each agent has unit size and all resources have integer capacities.

$\mathcal{L} = \langle e_{\pi_1}, \ldots, e_{\pi_{|D|}} \rangle$ output by SORT satisfies $z(e_{\pi_i}) \geq^{\text{lex}} z(e_{\pi_j})$ for all $i < j$. GREEDY then processes the edges in this order by always removing the first element (i, j) from $\mathcal{L}$, and greedily assigns agent i to resource j whenever this maintains the feasibility of the constructed assignment M. If i can be assigned to j, the assignment M is updated accordingly and all edges of i are removed from the list $\mathcal{L}$. GREEDY terminates if there are no more edges in $\mathcal{L}$.

It is important to realize that GREEDY coupled with an arbitrary ranking function z may not result in a strategyproof mechanism for GAP$^+$ in general. However, we show that mechanisms derived through this template allow us to obtain meaningful results for several GAP$^+$ variants studied in this paper. Definition 5.1 captures a sufficient condition for group-strategyproofness for a ranking function z for GREEDY, as we show in Theorem 5.2.

Definition 5.1. Consider some class of instances of GAP$^+$ in the private graph model. We say that a ranking function z is *truth-inducing* if for every instance $\mathcal{I}_{\text{GAP}^+}$ of this class it holds that:

(1) The extended lexicographic order of $L \times R$ with respect to z is strict and total.
(2) For every agent $i \in L$, and every $e = (i, j)$, $e' = (i, j') \in L \times R$ with $z(e) \geq^{\text{lex}} z(e')$, it holds that $v_e \geq v_{e'}$.

THEOREM 5.2. *Consider some class of instances of GAP$^+$ in the private graph model and let z be a ranking function that is truth-inducing with respect to this class. Then GREEDY coupled with z is a group-strategyproof mechanism.*

Finally, we stress that, given an instance $\mathcal{I}_{\text{GAP}^+} = (G[D], \boldsymbol{v}, \boldsymbol{s}, \boldsymbol{C}, \hat{M})$, GREEDY is not necessarily dependent on the predicted assignment $\hat{M}$; it can handle a non-augmented instance of GAP as well. However, the flexibility is in place for the accompanying ranking function z to use $\hat{M}$ in a beneficial manner for the underlying optimization problem. In the following section we present an implementation of this concept.

5.2 Restricted Multiple Knapsack

We introduce the GAP variant called *Restricted Multiple Knapsack (RMK)*.[13] In this variant, given an instance $\mathcal{I}_{\text{GAP}} = (G[D], \boldsymbol{v}, \boldsymbol{s}, \boldsymbol{C})$, each agent $i \in L$ has a fixed value $v_i = v_{ij}$ and size $s_i = s_{ij}$ for all $j \in R$. For the private graph model, Dughmi and Ghosh [2010] showed a randomized, strategyproof-in-expectation, 2-approximate mechanism for this problem, while a 4-approximation under universal-strategyproofness is implied by Chen et al. [2014] (through a generalization). To the best of our knowledge, no deterministic strategyproof mechanism is known.

In this section, we focus on devising a deterministic, group-strategyproof mechanism for the special case where $v_i = s_i$ for all $i \in L$ and we refer to it as *Equal RMK (ERMK)* throughout the section.[14] We denote an instance of this special case as $\mathcal{I}_{\text{ERMK}} = (G[D], (v_i = s_i)_{i \in L}, \boldsymbol{C})$. Interestingly, ERMK is strongly NP-hard as shown by Dawande et al. [2000]. For this class of instances, we show that coupling our GREEDY mechanism with a carefully chosen ranking function z gives a deterministic group-strategyproof mechanism achieving constant approximation guarantees (for every fixed $\gamma \geq 1$). Our ranking function z combines our boosted offer notion $\theta_{ij}(\gamma, \hat{M})$ with a specific tie-breaking rule to favor edges in the predicted assignment $\hat{M}$. This allows us to derive improved approximation guarantees if the prediction error is small, while at the same time retaining bounded robustness if the prediction is erroneous.

Let $\mathcal{I}_{\text{ERMK}^+} = (G[D], (v_i = s_i)_{i \in L}, \boldsymbol{C}, \hat{M})$ be an instance of ERMK$^+$. Note that $\hat{M}$ is a many-to-one assignment here. Let $\gamma \geq 1$ be fixed arbitrarily. We define the ranking function $z : L \times R \mapsto \mathbb{R}^4$

[13] Also known as *Multiple Knapsacks with Assignment Restrictions* (see, e.g., Aerts et al. [2003], Dawande et al. [2000], Nutov et al. [2006]).

[14] We readdress the general problem in Section 6.2 through a more general variant, by devising a randomized mechanism.

as follows: Let the boosted offer $\theta_{ij}(\gamma, \hat{M})$ be defined as in (4). Also, let $\mathbb{1}_{(i,j)\in\hat{M}}$ be the indicator function which is 1 if and only if $(i, j) \in \hat{M}$. Then, for each $(i, j) \in L \times R$, we define

$$z((i,j)) := \Big(\theta_{ij}(\gamma, \hat{M}),\ \mathbb{1}_{(i,j)\in\hat{M}},\ -i,\ -j\Big). \tag{6}$$

The intuition behind our ranking function is to rank the edges in D by their γ-boosted value. Recall that for ERMK we have for each agent $i \in L$, $v_i = v_{ij}$ for all $j \in R$. In particular, for $\gamma > 1$, the first-order criterion $\theta_{ij}(\gamma, \hat{M})$ ensures that the predicted edge of agent i in D_i is ordered before the non-predicted ones. In fact, crucially, the second-order criterion ensures that this property also holds for $\gamma = 1$. Put differently, whenever $\theta_e(\gamma, \hat{M}) = \theta_{e'}(\gamma, \hat{M})$ we make sure that priority is given to edges in $\hat{M} \cap D$. Remarkably, the preference we give to the predictions in case of ties leads to improved approximation guarantees even for $\gamma = 1$, if the prediction error η is small, i.e., for $\eta < 1/3$. If any ties remain, they are broken in increasing index of first i and then j.

We use Greedy-by-Theta to refer to the mechanism that we derive from Greedy with the ranking function z as defined in (6).

Theorem 5.3. *Fix some error parameter $\hat{\eta} \in [0, 1]$. Consider the class of instances of ERMK$^+$ in the private graph model and prediction error at most $\hat{\eta}$. Then, for every confidence parameter $\gamma \geq 1$, Greedy-by-Theta is group-strategyproof and has an approximation guarantee of*

$$g(\hat{\eta}, \gamma) = \begin{cases} \frac{1+\gamma}{\gamma(1-\hat{\eta})} & \text{if } \hat{\eta} \leq 1 - \frac{\gamma+1}{\gamma(\gamma+2)}, \\ 2+\gamma & \text{otherwise.} \end{cases} \tag{7}$$

In particular, Greedy-by-Theta is $(1 + 1/\gamma)$-consistent and $(2 + \gamma)$-robust.

Note that for $\gamma = 1$ our result implies a 3-approximate, group-strategyproof mechanism for ERMK. To the best of our knowledge, no deterministic strategyproof mechanism was known prior to our work.

6 Randomized Mechanisms for GAP Variants With Predictions

In this section, we devise randomized mechanisms for variants of GAP with predictions. It is important to note that all randomized mechanisms in this section attain the stronger property of universal group-strategyproofness.

A common thread among all mechanisms in this section is that they may return the outcome of Trust with some probability. As discussed in Section 3, Trust alone lacks a robustness guarantee. However, we can obtain randomized schemes with improved (expected) robustness guarantees by mixing between Trust and other mechanisms. First, we present our methodology in Section 6.1 by applying it to BMP$^+$. Then, in Section 6.2, we derive our randomized mechanisms for more general variants of GAP.

6.1 Improved Robustness via Randomization: A Separation Result for Matching

We demonstrate our idea by applying it to BMP$^+$. Our mechanism Boost-or-Trust randomizes over two deterministic mechanisms, one with and one without a robustness guarantee. Boost-or-Trust can be summarized as follows: given an instance $\mathcal{I}_{\text{BMP}^+}$ and a confidence parameter $\gamma \geq 1$, with probability p the mechanism outputs $M_1 = \text{Boost}(\mathcal{I}_{\text{BMP}^+}, \delta(\gamma))$, with $\delta(\gamma) = \sqrt{2(\gamma + 1)} - 1$. With probability $1 - p$, Boost-or-Trust outputs the matching $M_2 = \text{Trust}(\mathcal{I}_{\text{BMP}^+}) = \hat{M} \cap D$. As we have discussed in Section 3, using Trust implies confidence in the quality of the prediction $\hat{M}$. In expectation, this trade-off gives Boost-or-Trust an edge regarding robustness compared to simply running Boost and, at the same time, allows it to retain the approximation of Boost.

MECHANISM 4: Boost-or-Trust($\mathcal{I}_{\mathrm{BMP}^+}, \gamma$)

Input: An instance $\mathcal{I}_{\mathrm{BMP}^+} = (G[D], \boldsymbol{v}, \hat{M})$, confidence parameter $\gamma \geq 1$.
Output: A probability distribution over matchings for $\mathcal{I}_{\mathrm{BMP}^+}$.
Let $\delta(\gamma) = \sqrt{2(\gamma+1)} - 1$.
Set $M_1 = \text{Boost}(\mathcal{I}_{\mathrm{BMP}^+}, \delta(\gamma))$. `// Note that` $\delta(\gamma) \geq 1$ `for all` $\gamma \geq 1$.
Set $M_2 = \text{Trust}(\mathcal{I}_{\mathrm{BMP}^+}, D)$.
Set $p = 2/(\delta(\gamma)+1)$. `// Note that` $p \in (0,1]$ `for all` $\gamma \geq 1$.
return M_1 with probability p and M_2 with probability $1-p$.

Theorem 6.1. *Fix some error parameter $\hat{\eta} \in [0,1]$. Consider the class of instances of BMP$^+$ in the private graph model with prediction error at most $\hat{\eta}$. Then, for every confidence parameter $\gamma \geq 1$, Boost-or-Trust is universally group-strategyproof and has an expected approximation guarantee of*

$$g(\bar{\eta}, \gamma) = \begin{cases} \frac{1+\gamma}{\gamma(1-\hat{\eta})} & \text{if } \hat{\eta} \leq 1 - \frac{\sqrt{2(\gamma+1)}}{2\gamma}, \\ \sqrt{2(\gamma+1)} & \text{otherwise.} \end{cases}$$

In particular, Boost-or-Trust is $(1 + {}^1\!/\gamma)$-consistent and $\sqrt{2(\gamma+1)}$-robust (both in expectation).

Remark 6.2. *By Corollary 4.4, we have concluded that Boost can be adapted to handle instances of RSGAP$^+$ with the same performance as the one-to-one case and remain group-strategyproof. Therefore, Theorem 6.1 applies for this case as well. Furthermore, the same technique can be used for ERMK$^+$ (Section 5.2) to yield a universally group-strategyproof randomized mechanism, see the full version of the paper for a detailed exposition.*

Recall that, by Theorem 3.1 and Theorem 4.1, we have concluded that Boost attains the optimal consistency-robustness trade-off among deterministic strategyproof mechanisms. However, as shown in Theorem 6.1, the (expected) consistency-robustness of Boost-or-Trust is strictly better than that of any deterministic mechanism. This implies a separation between the two classes of mechanisms for BMP$^+$ in our environment with predictions.

6.2 More General Variants of GAP$^+$

In this section we devise randomized universally group-strategyproof mechanisms for two variants of GAP$^+$, namely VCGAP$^+$ and ASGAP$^+$. We write $\mathcal{I}_{\mathrm{VCGAP}^+} = (G[D], \boldsymbol{v}, \boldsymbol{s}, \boldsymbol{C}, \hat{M})$ and $\mathcal{I}_{\mathrm{ASGAP}^+} = (G[D], \boldsymbol{v}, \boldsymbol{s}, \boldsymbol{C}, \hat{M})$ to denote an instance of VCGAP$^+$ and ASGAP$^+$ respectively. While Dughmi and Ghosh [2010] and Chen et al. [2014] studied multiple GAP variants for the private graph model, to the best of our knowledge, the VCGAP has not been considered in the literature and for the problems without predictions, i.e., VCGAP and ASGAP, no deterministic strategyproof $O(1)$-approximate mechanism for the private graph model is known.

Greedy Mechanisms for VCGAP and ASGAP. In Section 5.2 we demonstrated how Mechanism Greedy combined with an appropriate ranking function can serve as a group-strategyproof mechanism for the special case of ERMK$^+$. Here, we present two distinct instantiations of ranking functions, one for VCGAP and one for ASGAP, which can be coupled with Greedy to obtain a group-strategyproof mechanism for their respective classes of instances.

We present our ranking function for VCGAP first. We define the function $z_{\mathrm{VCGAP}} : L \times R \mapsto \mathbb{R}^3$ as follows: Let $\sigma = (\sigma(1), \ldots, \sigma(m))$ be the ordinal consensus permutation of resources in R for the instance (note that such a permutation is guaranteed to exist). Then, for every pair $(i,j) \in L \times R$, we define

$$z_{\mathrm{VCGAP}}((i,j)) := \Big(-\sigma(j), {}^{v_{ij}}\!/{}_{s_{ij}}, -i \Big). \tag{8}$$

This function instructs Greedy to rank edges in D so that the edges linked to the most "preferred" resources, according to the ordinal consensus σ, are considered first. Then, among the edges linked to each resource, the second-order criterion instructs Greedy to give precedence to the edge with the highest value per size ratio. If any ties remain, they are broken in increasing index of i.

For ASGAP$^+$, we define the function $z_{\text{ASGAP}} : L \times R \mapsto \mathbb{R}^3$ as follows: For every pair $(i, j) \in L \times R$, we define

$$z_{\text{ASGAP}}((i, j)) := \left(v_{ij}/s_{ij}, -i, -j\right). \tag{9}$$

This ranking function is particularly straightforward; it instructs Greedy to prioritize edges with the highest value per size ratio in the greedy ordering. Then, in case of ties, they are broken in increasing index of first i and then j.

Throughout this section, when invoking Greedy for VCGAP$^+$ and ASGAP$^+$ instances, we will refer to the pairing of Greedy with the corresponding ranking functions z_{VCGAP} and z_{ASGAP} respectively. In Lemma 6.3 we show that each of the two instantiations of Greedy described above is a group-strategyproof mechanism for its respective class of instances.

Lemma 6.3. *Mechanism Greedy coupled with ranking function z_{VCGAP} (resp. z_{ASGAP}) is a group-strategyproof mechanism for instances of VCGAP$^+$ (resp. ASGAP$^+$).*

Note that neither ranking function above depends on the predicted assignment $\hat{M}$ in any way. Furthermore, the greedy mechanisms we describe above do not guarantee worst-case approximation guarantees when run as stand-alone mechanisms. However, both ranking functions ensure that agents are processed by Greedy in an efficient way.

Observation 6.4. *For an instance $\mathcal{I}_{GAP^+} = (G[D], \boldsymbol{v}, \boldsymbol{s}, C, \hat{M})$ which is either VCGAP$^+$ or ASGAP$^+$, the ranking functions z as defined in* (8) *and* (9) *satisfy the following property: for every resource $j \in R$, and every $(i, j), (i', j) \in D_j$ with $z((i, j)) \geq^{lex} z((i', j))$, it holds that $\frac{v_{ij}}{s_{ij}} \geq \frac{v_{i'j}}{s_{i'j}}$.*

Randomized Mechanism for VCGAP$^+$ and ASGAP$^+$. We present our randomized universally group-strategyproof mechanism for VCGAP$^+$ and ASGAP$^+$. There are two main pillars in our approach. The first one is that we randomize over the respective Greedy mechanism presented above, which processes agents in order of efficiency (as argued in Observation 6.4), and a complementary mechanism, which processes agents in order of their values. While neither of these mechanisms achieves a bounded approximation guarantee by itself, their (probabilistic) combination does in expectation. In fact, this is the key idea that Chen et al. [2014] used to devise the current state-of-the-art universally strategyproof mechanism for ASGAP and special cases of VCGAP. The deferred-acceptance mechanism that they use can be cast into our Greedy template as it simply follows a specific ranking of the edges such that agents are never unmatched (see proof Theorem 1 of Chen et al. [2014]).

Inspired by this idea, we instead randomize over Greedy and our mechanism Boost for BMP$^+$ to leverage the predicted assignment. Additionally, we combine the above scheme with a third mechanism, namely Trust, to follow the prediction with some (small) probability. We refer to the resulting mechanism as Boost-or-Greedy-or-Trust; see Mechanism 5. Note that the assignments M_1 and M_2 computed by Boost and Greedy, respectively, are always returned with positive probability $p = 2/(3 + \gamma)$, while the predicted assignment output by Trust is returned with probability $1 - 2p$, which is positive only if $\gamma > 1$ (i.e., when there is some confidence in the prediction). A subtle point that needs some clarification here is that the predicted assignment $\hat{M}$ of the constructed instance $\mathcal{I}_{\text{BMP}^+}$ passed on to Boost is a many-to-one assignment. However, as argued in Section 4.2 Extension (E1), Boost can handle such predictions as well. In fact, M_1 being a one-to-one assignment output by Boost suffices to prove bounded approximation guarantees for Boost-or-Greedy-or-Trust.

MECHANISM 5: Boost-or-Greedy-or-Trust ($\mathcal{I}_{\text{GAP}^+}, \boldsymbol{z}, \gamma$)

Input: An instance $\mathcal{I}_{\text{GAP}^+} = (G[D], \boldsymbol{v}, \boldsymbol{s}, C, \hat{M})$, a ranking function $\boldsymbol{z} : L \times R \mapsto \mathbb{R}^k$ for some $k \in \mathbb{N}$ and a confidence parameter $\gamma \geq 1$.
Output: A probability distribution of feasible assignments for $\mathcal{I}_{\text{GAP}^+}$.
Construct an instance $\mathcal{I}_{\text{BMP}^+} = (G[D], \boldsymbol{v}, \hat{M})$.
Set $M_1 = \text{Boost}(\mathcal{I}_{\text{BMP}^+}, \gamma)$ and $M_2 = \text{Greedy}(\mathcal{I}_{\text{GAP}^+}, \boldsymbol{z})$.
Set $p = 2/(3 + \gamma)$.
return M_1 with probability p, M_2 with probability p and $\text{Trust}(\mathcal{I}_{\text{GAP}^+})$ with probability $1 - 2p$.

Theorem 6.5. *Fix some error parameter $\hat{\eta} \in [0, 1]$. Consider the class of instances of ASGAP$^+$ and VCGAP$^+$ in the private graph model with prediction error at most $\hat{\eta}$. Then, for every confidence parameter $\gamma \geq 1$, Boost-or-Greedy-or-Trust is universally group-strategyproof and has an expected approximation guarantee of*

$$g(\hat{\eta}, \gamma) = \begin{cases} \frac{3+\gamma}{\gamma(1-\hat{\eta})} & \text{if } \hat{\eta} \leq 1 - \frac{1}{\gamma}, \\ 3 + \gamma & \text{otherwise.} \end{cases} \tag{10}$$

In particular, Boost-or-Greedy-or-Trust is $(1+3/\gamma)$-consistent and $(3+\gamma)$-robust (both in expectation).

7 Conclusions

In this paper, we initiated the study of generalized assignment problems in the private graph model with assignment predictions. An important building block for the GAP$^+$ variants that we consider is the mechanism Boost. For BMP$^+$, Boost is an optimal group-strategyproof (GSP) mechanism in terms of consistency and robustness, with an interpolation of the approximation guarantee between the two. We provide a lower bound for this interpolation, which is not tight. We additionally prove a separation result between deterministic and randomized GSP mechanisms for the trade-off of consistency and robustness. Another important building block for VCGAP$^+$ and ASGAP$^+$ is the mechanism Greedy. Greedy is GSP when executed with a truth-inducing ranking function. However, as no polynomial time deterministic strategyproof mechanism is known for these variants, we turn to randomization over Boost, Greedy and Trust. This results in a universally GSP mechanism, with a non-optimal trade-off between consistency and robustness. Besides improving on this trade-off, through improving either the upper or lower bound (for VCGAP$^+$), it would be interesting to obtain deterministic mechanisms for these variants, with or without predictions.

We believe the problems and techniques considered in this paper are interesting and relevant to applications and other problems. A specific property of the problems that we consider, is that they are enriched with a prediction of the optimal assignment. Our mechanisms leverage this by increasing the preference for declared edges that are in the prediction, potentially giving these edges an advantage. A natural direction for further research would be to apply this technique on other mechanism design problem without money enriched with a structural prediction.

Acknowledgments

This work was supported by the Dutch Research Council (NWO) through its Open Technology Program, proj. no. 18938, and the Gravitation Project NETWORKS, grant no. 024.002.003. It has also been funded by the European Union under the EU Horizon 2020 Research and Innovation Program, Marie Skłodowska-Curie Grant Agreement, grant no. 101034253.

References

Joep Aerts, Jan Korst, Frits Spieksma, Wim Verhaegh, and Gerhard Woeginger. 2003. Random redundant storage in disk arrays: complexity of retrieval problems. *IEEE Trans. Comput.* 52, 9 (2003), 1210–1214.

Priyank Agrawal, Eric Balkanski, Vasilis Gkatzelis, Tingting Ou, and Xizhi Tan. 2023. Learning-Augmented Mechanism Design: Leveraging Predictions for Facility Location. *Mathematics of Operations Research* (2023).

Antonios Antoniadis, Christian Coester, Marek Eliáš, Adam Polak, and Bertrand Simon. 2023a. Online metric algorithms with untrusted predictions. *ACM Transactions on Algorithms* 19, 2 (2023), 1–34.

Antonios Antoniadis, Themis Gouleakis, Pieter Kleer, and Pavel Kolev. 2023b. Secretary and online matching problems with machine learned advice. *Discrete Optimization* 48 (2023), 100778.

Yossi Azar, Stefano Leonardi, and Noam Touitou. 2021. Flow time scheduling with uncertain processing time. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*. SIAM, 1070–1080.

Yossi Azar, Debmalya Panigrahi, and Noam Touitou. 2022. Online graph algorithms with predictions. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 35–66.

Eric Balkanski, Vasilis Gkatzelis, and Xizhi Tan. 2023a. Strategyproof Scheduling with Predictions. In *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*, Vol. 251. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 11.

Eric Balkanski, Vasilis Gkatzelis, Xizhi Tan, and Cherlin Zhu. 2023b. Online Mechanism Design with Predictions. *arXiv preprint arXiv:2310.02879* (2023).

Siddhartha Banerjee, Vasilis Gkatzelis, Artur Gorokh, and Billy Jin. 2022. Online nash social welfare maximization with predictions. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 1–19.

Ben Berger, Michal Feldman, Vasilis Gkatzelis, and Xizhi Tan. 2023. Optimal Metric Distortion with Predictions. *arXiv preprint arXiv:2307.07495* (2023).

Ioannis Caragiannis and Georgios Kalantzis. 2024. Randomized learning-augmented auctions with revenue guarantees. *arXiv preprint arXiv:2401.13384* (2024).

Deeparnab Chakrabarty and Gagan Goel. 2010. On the approximability of budgeted allocations and improved lower bounds for submodular welfare maximization and GAP. *SIAM J. Comput.* 39, 6 (2010), 2189–2211.

Justin Chen, Sandeep Silwal, Ali Vakilian, and Fred Zhang. 2022. Faster fundamental graph algorithms via learned predictions. In *International Conference on Machine Learning*. PMLR, 3583–3602.

Ning Chen, Nick Gravin, and Pinyan Lu. 2014. Truthful generalized assignments via stable matching. *Mathematics of Operations Research* 39, 3 (2014), 722–736.

Riccardo Colini-Baldeschi, Sophie Klumper, Guido Schäfer, and Artem Tsikiridis. 2024. To Trust or Not to Trust: Assignment Mechanisms with Predictions in the Private Graph Model. *arXiv preprint arXiv:2403.03725* (2024).

Milind Dawande, Jayant Kalagnanam, Pinar Keskinocak, F Sibel Salman, and R Ravi. 2000. Approximation algorithms for the multiple knapsack problem with assignment restrictions. *Journal of combinatorial optimization* 4 (2000), 171–186.

Michael Dinitz, Sungjin Im, Thomas Lavastida, Benjamin Moseley, and Sergei Vassilvitskii. 2021. Faster matchings via learned duals. *Advances in neural information processing systems* 34 (2021), 10393–10406.

Michael Dinitz, Sungjin Im, Thomas Lavastida, Benjamin Moseley, and Sergei Vassilvitskii. 2022. Algorithms with prediction portfolios. *Advances in neural information processing systems* 35 (2022), 20273–20286.

Shaddin Dughmi and Arpita Ghosh. 2010. Truthful assignment without money. In *Proceedings of the 11th ACM conference on Electronic commerce*. ACM, 325–334.

Salman Fadaei and Martin Bichler. 2017a. Generalized assignment problem: Truthful mechanism design without money. *Operations Research Letters* 45, 1 (2017), 72–76.

Salman Fadaei and Martin Bichler. 2017b. A truthful mechanism for the generalized assignment problem. *ACM Transactions on Economics and Computation (TEAC)* 5, 3 (2017), 1–18.

Uriel Feige and Jan Vondrak. 2006. Approximation algorithms for allocation problems: Improving the factor of 1-1/e. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*. IEEE, 667–676.

D. Gale and L. S. Shapley. 1962. College Admissions and the Stability of Marriage. *The American Mathematical Monthly* 69, 1 (1962), 9–15. https://doi.org/10.1080/00029890.1962.11989827

Vasilis Gkatzelis, Kostas Kollias, Alkmini Sgouritsa, and Xizhi Tan. 2022. Improved price of anarchy via predictions. In *Proceedings of the 23rd ACM Conference on Economics and Computation*. ACM, 529–557.

John William Hatfield and Paul R. Milgrom. 2005. Matching with Contracts. *American Economic Review* 95, 4 (2005), 913–935. https://doi.org/10.1257/0002828054825466

Gabriel Istrate and Cosmin Bonchis. 2022. Mechanism Design With Predictions for Obnoxious Facility Location. *arXiv preprint arXiv:2212.09521* (2022).

Billy Jin and Will Ma. 2022. Online bipartite matching with advice: Tight robustness-consistency tradeoffs for the two-stage model. *Advances in Neural Information Processing Systems* 35 (2022), 14555–14567.

Thomas Lavastida, Benjamin Moseley, R Ravi, and Chenyang Xu. 2021a. Learnable and Instance-Robust Predictions for Online Matching, Flows and Load Balancing. In *29th Annual European Symposium on Algorithms (ESA 2021)*, Vol. 204.

Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 59.

Thomas Lavastida, Benjamin Moseley, R Ravi, and Chenyang Xu. 2021b. Using predicted weights for ad delivery. In *SIAM Conference on Applied and Computational Discrete Algorithms (ACDA21)*. SIAM, 21–31.

Pinyan Lu, Zongqi Wan, and Jialin Zhang. 2023. Competitive Auctions with Imperfect Predictions. *arXiv preprint arXiv:2309.15414* (2023).

Thodoris Lykouris and Sergei Vassilvitskii. 2021. Competitive caching with machine learned advice. *Journal of the ACM (JACM)* 68, 4 (2021), 1–25.

Michael Mitzenmacher and Sergei Vassilvitskii. 2020. Algorithms with Predictions. *arXiv preprint arXiv:2006.09123* (2020).

Zeev Nutov, Israel Beniaminy, and Raphael Yuster. 2006. A (1−1/e)-approximation algorithm for the generalized assignment problem. *Operations Research Letters* 34, 3 (2006), 283–288.

Siddharth Prasad, Maria-Florina F Balcan, and Tuomas Sandholm. 2024. Bicriteria Multidimensional Mechanism Design with Side Information. *Advances in Neural Information Processing Systems* 36 (2024).

Ariel D Procaccia and Moshe Tennenholtz. 2013. Approximate mechanism design without money. *ACM Transactions on Economics and Computation (TEAC)* 1, 4 (2013), 1–26.

Manish Purohit, Zoya Svitkina, and Ravi Kumar. 2018. Improving online algorithms via ML predictions. *Advances in Neural Information Processing Systems* 31 (2018).

Alvin E. Roth. 1982. The Economics of Matching: Stability and Incentives. *Mathematics of Operations Research* 7, 4 (1982), 617–628.

Tim Roughgarden. 2010. Algorithmic game theory. *Commun. ACM* 53, 7 (2010), 78–86.

Tim Roughgarden. 2019. Beyond worst-case analysis. *Commun. ACM* 62, 3 (2019), 88–96.

Shinsaku Sakaue and Taihei Oki. 2022. Discrete-convex-analysis-based framework for warm-starting algorithms with predictions. *Advances in Neural Information Processing Systems* 35 (2022), 20988–21000.

James Schummer and Rakesh V Vohra. 2007. Mechanism design without money. *Algorithmic game theory* 10 (2007), 243–299.

Chenyang Xu and Pinyan Lu. 2022. Mechanism Design with Predictions. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, Lud De Raedt (Ed.). International Joint Conferences on Artificial Intelligence Organization, 571–577.