ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

PROJECT: ALGOLINC – Algorithmic Learning Theory and Incentives: Synergies in Optimization and Mechanism Design

Project ID: 15877

Greece 2.0 NATIONAL RECOVERY AND RESILIENCE PLAN "BASIC RESEARCH FINANCING" (Horizontal support for all Sciences)

WP3: Literature review and identification of main challenges

Deliverable D.3.1: Report on the relevant literature, main challenges and research questions

Prepared by the team members:

Michail Fasoulakis, Evangelos Markakis,

Panagiotis Patsilinakos, Panagiotis Tsamopoulos

ΕΛ.Ι.Δ.Ε.Κ.

Ελληνικό Ίδρυμα Έρευνας & Καινοτομίας

Deliverable D.3.1: Report on the Relevant Literature, Main Challenges and Research Questions

Michail Fasoulakis, Evangelos Markakis, Panagiotis Patsilinakos, and Panagiotis Tsamopoulos

Abstract

In recent years, we have seen increasing synergies between game theory, optimization, and machine learning. The goal of this deliverable is to provide an overview of the current approaches and mathematical models that have been adopted. At the same time, we also identify challenges and important questions for future research. The analysis is split into two domains, as described in the ALGOLINC proposal. The first one concerns the design of learning algorithms for finding Nash equilibria in games, whereas the second domain evolves around the design of mechanisms with the use of predictions (learning-augmented mechanism design).

1 Introduction

The goal of the proposed project is to contribute to the ongoing interplay between machine learning and algorithmic game theory. The field of algorithmic game theory lies at the intersection of computer science, game theory, and economics, and has evolved over the past two decades out of the need for efficient algorithms in game-theoretic models. In recent years, the advent of AI has created new challenges within algorithmic game theory. Essentially, several game-theoretic problems can be revisited through the lenses of machine learning, and new paradigms have emerged that deserve further investigation. Our project aims to focus on two important and to some extent complementary research components. The first one concerns the interaction between learning, optimization, and equilibrium computation in games and is centered on optimization problems that form the backbone of building efficient learning (training) algorithms. The second one moves in a reverse direction and concerns contributions of machine learning to mechanism design.

Optimization and learning in games Over the years, there have been several fruitful connections between optimization questions that arise in the context of multi-agent learning and game-theoretic solution concepts. As a representative starting point, we can consider the fundamental class of zero-sum games. Zero-sum games have played a fundamental role in both game theory, being among the first classes of games formally studied, and in optimization, as it is easily seen that their equilibrium solutions correspond to solving a min-max optimization problem of the form: $\min_x \max_y f(x, y)$, where x and y are vectors of variables, constrained to be probability distributions (i.e., mixed strategies for the two players). Even further, solving zero-sum games is in fact equivalent to solving linear programs, as properly demonstrated in [1]. Despite the fact that a single linear program suffices to find a Nash equilibrium, there has been a surge of interest in recent years for faster algorithms, motivated in part by applications in machine learning. One reason for this is that we may have very large games to solve, corresponding to LPs with thousands of variables and constraints. A second reason could be that, e.g., in learning environments, the players may be using iterative algorithms that can only observe limited information, hence it would be impossible to run a single linear program for the entire game. As an additional motivation, finding new algorithms for such a fundamental problem can provide insights that could be of further value and interest.

Learning-oriented mechanism design Mechanism design forms one of the main pillars of algorithmic game theory. It is often referred to as *inverse game theory* as the goal is to design the rules of a game to enforce certain desirable behaviors on the players. Auction mechanisms, in particular, constitute by far the most common subject in mechanism design research and range from high-stake governmental license auctions (among others, for telecommunications spectrum or carbon allowance) to procurement auctions for hiring sub-contractors in online marketplaces. Modern applications also involve richer bidding interfaces, where multiple items can be for sale simultaneously and the bidders can express preferences for combinations of items, resulting in combinatorial auctions. Regardless of the auction format, the general goal for an auction designer is typically two-fold: design the rules of the mechanism so as to constrain potential strategic behavior of the bidders and at the same time optimize economic performance. The main reason that machine learning can naturally come into play when designing auctions is the fact that the auctioneer may have access to data from previous auctions or may have access to a prediction algorithm for various parameters of the problem. This gives rise to a new paradigm of data-driven mechanism design, and one could essentially revisit all mechanism design problems from economic theory under this framework.

The rest of the survey is structured as follows: In Section 2, we focus on methods that have been proposed for computing Nash equilibria in games. In Section 3, our goal is to focus on the design of truthful mechanisms with the use of predictions. We conclude in Section 4 with a discussion of the directions of future work.

2 Learning in Games

We start with the first pillar of our project, which concerns learning algorithms for finding Nash equilibria in games.

2.1 Basic notation

Our project focuses mostly on bilinear zero-sum games. We will also mention extensions to more general problems in the sequel, but we will use zero-sum games as our basic setup.

A bilinear zero-sum game is defined by a $n \times n$ payoff matrix R with n pure strategies per player, and assume $R \in [0,1]^{n \times n}$, without loss of generality[1]. The payoff matrix R represents the utility of the row player, whereas the payoff matrix of the column player is equal to $-R$. Therefore, the game is often also denoted by the tuple $(R, -R)$. We consider mixed strategies $\boldsymbol{x} \in \Delta^{n-1}$ as a probability distribution (column vector) on the pure strategies of a player, with Δ^{n-1} be the $(n-1)$-dimensional simplex. We also denote by $\boldsymbol{e}_i$ the distribution corresponding to a pure strategy i, with 1 in the index i and zero elsewhere. A strategy profile is a pair $(\boldsymbol{x}, \boldsymbol{y})$, where $\boldsymbol{x}$ is the strategy of the row player and $\boldsymbol{y}$ is the strategy of the column player. Under a profile $(\boldsymbol{x}, \boldsymbol{y})$, the expected payoff of the row player is $\boldsymbol{x}^\top R \boldsymbol{y}$ and the expected payoff of the column player is $-\boldsymbol{x}^\top R \boldsymbol{y}$.

[1] We can easily see that we can do scaling for any $R \in \mathbb{R}^{n \times n}$ s.t. $R \in [0,1]^{n \times n}$ keeping exactly the same Nash equilibria.

A pure strategy i is a best-response strategy against $\boldsymbol{y}$ for the row player, if and only if, $\boldsymbol{e}_i^\top R\boldsymbol{y} \geq \boldsymbol{e}_j^\top R\boldsymbol{y}$, for any j. Similarly, a pure strategy j for the column player is a best-response strategy against some strategy $\boldsymbol{x}$ of the row player, if and only if, $\boldsymbol{x}^T R\boldsymbol{e}_j \leq \boldsymbol{x}^T R\boldsymbol{e}_i$, for any i. We will use $BR_r(\boldsymbol{y})$ and $BR_c(\boldsymbol{x})$ for the best-response sets.

Definition 2.1 (Nash equilibrium [40, 47])**.** *A strategy profile $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is a Nash equilibrium in the game $(R, -R)$, if and only if, for any i, j,*

$$v = \boldsymbol{x}^{*\top} R\boldsymbol{y}^* \geq \boldsymbol{e}_i^\top R\boldsymbol{y}^*, \text{ and, } v = \boldsymbol{x}^{*\top} R\boldsymbol{y}^* \leq \boldsymbol{x}^{*\top} R\boldsymbol{e}_j,$$

where v is the value of the row player (value of the game).

Definition 2.2 (δ-Nash equilibrium)**.** *A strategy profile $(\boldsymbol{x}, \boldsymbol{y})$ is a δ-Nash equilibrium (in short, δ-NE) in the game $(R, -R)$, with $\delta \in [0, 1]$, if and only if, for any i, j,*

$$\boldsymbol{x}^\top R\boldsymbol{y} + \delta \geq \boldsymbol{e}_i^\top R\boldsymbol{y}, \text{ and, } \boldsymbol{x}^\top R\boldsymbol{y} - \delta \leq \boldsymbol{x}^\top R\boldsymbol{e}_j.$$

It is easy to see that any 0-NE is an exact Nash equilibrium. With these at hand, we can now define the regret functions of the players as follows.

Definition 2.3 (Regret of a player)**.** *For a game $(R, -R)$, the regret function $f_R : \Delta^{n-1} \times \Delta^{n-1} \to [0, 1]$ of the row player under a strategy profile $(\boldsymbol{x}, \boldsymbol{y})$ is*

$$f_R(\boldsymbol{x}, \boldsymbol{y}) = \max_i \boldsymbol{e}_i^\top R\boldsymbol{y} - \boldsymbol{x}^\top R\boldsymbol{y}.$$

Similarly, for the column player the regret function is

$$\begin{aligned} f_{-R}(\boldsymbol{x}, \boldsymbol{y}) &= \max_j \boldsymbol{x}^\top (-R)\boldsymbol{e}_j + \boldsymbol{x}^\top R\boldsymbol{y} \\ &= -\min_j \boldsymbol{x}^\top R\boldsymbol{e}_j + \boldsymbol{x}^\top R\boldsymbol{y}. \end{aligned}$$

An important quantity for evaluating the performance or convergence of algorithms is the sum of regrets, that is, the function $V(\boldsymbol{x}, \boldsymbol{y}) = f_R(\boldsymbol{x}, \boldsymbol{y}) + f_{-R}(\boldsymbol{x}, \boldsymbol{y}) = \max_i \boldsymbol{e}_i^\top R\boldsymbol{y} - \min_j \boldsymbol{x}^\top R\boldsymbol{e}_j$. This is referred to in the bibliography as the *duality gap* in the case of zero-sum games.

Having these formulations, we are interested in deriving learning algorithms that the players achieve an (approximate) equilibrium. These algorithms can be classified, in terms of convergence, into two categories: *average convergence* and *last-iterate convergence.*

2.2 Average convergence

Designing learning algorithms for zero-sum games is a direction that started already several decades ago, e.g. with the fictitious play algorithm [12, 44] and the Multiplicative Weights Update (MWU) method [5, 24]. These two fundamental algorithms have the property that converge on average, in other words, the average of the strategies that are played until round t converges to an (approximate) Nash equilibrium as $t \to \infty$.

More formally, suppose that we have an algorithm that generates a sequence of strategy profiles $(\boldsymbol{x}^t, \boldsymbol{y}^t)$, for any iteration $t \geq 0$, where $(\boldsymbol{x}^0, \boldsymbol{y}^0)$ is the initial strategy profile. Then, the algorithm converges on average to a strategy profile $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ as time $t \to \infty$, if and only if, $\frac{1}{\tau} \sum_{0 \leq \tau \leq t} (\boldsymbol{x}^t, \boldsymbol{y}^t)$ converges to $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ as $t \to \infty$.

We now discuss two of the main algorithms that converge on average in zero-sum bimatrix games, the Gradient Descent/Ascent (GDA) and the Multiplicative Weights Update (MWU) method.

2.2.1 The GDA method

The GDA method is based on the gradient of a differentiable scalar function and the movement in this direction. If we wanted to optimize a single convex function, it would suffice to perform the standard gradient descent method. Since, we are now at a zero-sum game, where one player wants to maximize her payoff while the other player wants to decrease her payment, the extension of gradient descent is precisely the gradient descent/ascent method.

Formally, suppose that the payoff function $f(\boldsymbol{x}, \boldsymbol{y})$, is differentiable. Then, if we apply the GDA method the update for any i, j is

$$\begin{aligned} \boldsymbol{x}_i^{t+1} &= \boldsymbol{x}_i^t + \eta \frac{\partial f(\boldsymbol{x}^t, \boldsymbol{y}^t)}{\partial \boldsymbol{x}_i^t}, \\ \boldsymbol{y}_j^{t+1} &= \boldsymbol{y}_j^t - \eta \frac{\partial f(\boldsymbol{x}^t, \boldsymbol{y}^t)}{\partial \boldsymbol{y}_j^t}. \end{aligned}$$

Above, the notation $\boldsymbol{x}_i^{t+1}$ stands for the i-th coordinate of $\boldsymbol{x}^{t+1}$. Note that in the context of a zero sum bimatrix game $(R, -R)$ the function f is equal to the payoff matrix of the row player, namely $f(\boldsymbol{x}^t, \boldsymbol{y}^t) = (\boldsymbol{x}^t)^\top R \boldsymbol{y}^t$, with $\frac{\partial f(\boldsymbol{x}^t, \boldsymbol{y}^t)}{\partial \boldsymbol{x}_i^t} = \boldsymbol{e}_i^\top R \boldsymbol{y}^t$ and $\frac{\partial f(\boldsymbol{x}^t, \boldsymbol{y}^t)}{\partial \boldsymbol{y}_j^t} = (\boldsymbol{x}^t)^\top R \boldsymbol{e}_j$. Also note that since the strategies of the players are probability distributions, we need to normalize the update rule so that at the end the profile $(\boldsymbol{x}^{t+1}, \boldsymbol{y}^{t+1})$ is a valid strategy profile. Overall this results in the following update rule for the GDA method.

$$\begin{aligned} \boldsymbol{x}_i^{t+1} &= \frac{\boldsymbol{x}_i^t + \eta \cdot \boldsymbol{e}_i^\top R \boldsymbol{y}^t}{\sum_j (\boldsymbol{x}_j^t + \eta \cdot \boldsymbol{e}_j^\top R \boldsymbol{y}^t)}, \\ \boldsymbol{y}_j^{t+1} &= \frac{\boldsymbol{y}_j^t - \eta \cdot (\boldsymbol{x}^t)^\top R \boldsymbol{e}_j}{\sum_i (\boldsymbol{y}_i^t - \eta \cdot (\boldsymbol{x}^t)^\top R \boldsymbol{e}_i)}. \end{aligned} \tag{1}$$

2.2.2 The MWU method

Another method that converges on average for the zero-sum bimatrix games is the MWU method [5, 24]. In this method, at high level, we consider that we have n experts (as the number of pure strategies in games), and the row player divides her initial probability on these experts; in game-theoretic words she initially uses a full support strategy. In any round, depending on the advice of any expert, she decreases or increases her probability on this. In other words, if the advice is good, then there is an increase in the probability that is assigned to this, and in the other case there is a decrease. It was proven that this method converges on average to an ε-Nash equilibrium for zero-sum games, for $\varepsilon > 0$. This result is stated in the following theorem.

Theorem 2.4 (See [5]). *After $T = O\left(\frac{\ln n}{\delta^2}\right)$ rounds, the MWU method converges to a δ-Nash equilibrium, for any $\delta > 0$, of the bimatrix zero-sum game on average.*

A version of this method for zero-sum games is the following:

$$\begin{aligned} \boldsymbol{x}_i^{t+1} &= \frac{\boldsymbol{x}_i^t \cdot e^{\eta \cdot \boldsymbol{e}_i^\top R y^t}}{\sum_j \boldsymbol{x}_j^t \cdot e^{\eta \cdot \boldsymbol{e}_j^\top R y^t}}, \\ \boldsymbol{y}_j^{t+1} &= \frac{\boldsymbol{y}_j^t \cdot e^{-\eta \cdot (\boldsymbol{x}^t)^\top R e_j}}{\sum_i \boldsymbol{y}_i^t \cdot e^{-\eta \cdot (\boldsymbol{x}^t)^\top R e_i}}, \end{aligned}$$

with $\eta \in (0,1]$ be the step size. For arbitrarily small η and using Taylor's expansion of the exponent function e^x this is equivalent to:

$$\boldsymbol{x}_i^{t+1} = \frac{\boldsymbol{x}_i^t \cdot (1 + \eta \cdot \boldsymbol{e}_i^\top R y^t)}{\sum_j \boldsymbol{x}_j^t \cdot (1 + \eta \cdot \boldsymbol{e}_j^\top R y^t)},$$
$$\boldsymbol{y}_j^{t+1} = \frac{\boldsymbol{y}_j^t \cdot (1 - \eta \cdot (\boldsymbol{x}^t)^\top R e_j)}{\sum_i \boldsymbol{y}_i^t \cdot (1 - \eta \cdot (\boldsymbol{x}^t)^\top R e_i)},$$

which is referred as the linear version of the MWU method in the bibliography. As we can see, it is very close to the GDA method for zero-sum games, as was described above. Unfortunately, it has been proved that this method does not converge with last iterate in zero-sum games [7].

2.3 Last-iterate convergence

The previous subsection focused on algorithms that achieve only convergence on average. Within the last decade, there has also been a great interest in algorithms to attain the more robust notion of *last iterate convergence.*

Definition 2.5. *We say that an algorithm achieves last iterate convergence to an equilibrium if the limit of (x^t, y^t) as $t \to \infty$ is a Nash equilibrium of the underlying game, where (x^t, y^t) is the strategy profile at iteration t.*

Unfortunately, the methods that were presented earlier do not satisfy this condition. There are already negative results, as, e.g., in [7] and [38], showing that several no-regret algorithms such as many MWU as well as GDA variants, do not satisfy last-iterate convergence. Motivated by this, there has been a series of works on obtaining algorithms with provable last iterate convergence.

Most of the algorithms that have been obtained in the context of machine learing applications have emerged out of the seminal works of Korpelevich [33] and Popov [41], who defined the Extra Gradient (EG) and the Optimistic Gradient method respectively. To define these methods, recall first that $f(x, y) = \boldsymbol{x}^\top R \boldsymbol{y}$. The extra gradient method for zero-sum games proceeds in two phases in each iteration. Suppose that we have reached a strategy profile $(\boldsymbol{x}^t, \boldsymbol{y}^t)$ at the end of iteration t. The first stage of the next iteration is to produce an intermediate profile, which serves as a prediction for the opponent's move as follows:

$$\begin{aligned} \hat{\boldsymbol{x}}^t &= \boldsymbol{x}^t + \eta \nabla_{\boldsymbol{x}^t} f(\boldsymbol{x}^t, \boldsymbol{y}^t), \\ \hat{\boldsymbol{y}}^t &= \boldsymbol{y}^t - \eta \nabla_{\boldsymbol{y}^t} f(\boldsymbol{x}^t, \boldsymbol{y}^t). \end{aligned} \tag{2}$$

Then, at the second stage of iteration $t+1$, the strategy profile $(\boldsymbol{x}^t, \boldsymbol{y}^t)$ is created using an additional gradient step as follows:

$$\begin{aligned} \boldsymbol{x}^{t+1} &= \boldsymbol{x}^t + \eta \nabla_{\hat{\boldsymbol{x}}^t} f(\hat{\boldsymbol{x}}^t, \hat{\boldsymbol{y}}^t), \\ \boldsymbol{y}^{t+1} &= \boldsymbol{y}^t - \eta \nabla_{\hat{\boldsymbol{y}}^t} f(\hat{\boldsymbol{x}}^t, \hat{\boldsymbol{y}}^t). \end{aligned} \tag{3}$$

The optimistic gradient method utilizes a slightly different idea where we look at two iterations back and add a negative momentum in order to correct the dynamics. In particular the update rule in the unconstrained case is as follows, and we will refer to it as the OGDA method (Optimistic Gradient Descent Ascent):

$$
\begin{aligned}
\boldsymbol{x}^{t+1} &= \boldsymbol{x}^t + 2\eta \nabla_{\boldsymbol{x}^t} f(\boldsymbol{x}^t, \boldsymbol{y}^t) - \eta \nabla_{\boldsymbol{x}^{t-1}} f(\boldsymbol{x}^{t-1}, \boldsymbol{y}^{t-1}), \\
\boldsymbol{y}^{t+1} &= \boldsymbol{y}^t - 2\eta \nabla_{\boldsymbol{y}^t} f(\boldsymbol{x}^t, \boldsymbol{y}^t) + \eta \nabla_{\boldsymbol{y}^{t-1}} f(\boldsymbol{x}^{t-1}, \boldsymbol{y}^{t-1}).
\end{aligned}
\tag{4}
$$

The above rules apply to the unconstrained case regarding the profile $(\boldsymbol{x}, \boldsymbol{y})$. In our setting however, since we are mostly interested in the constrained case, where $\boldsymbol{x}$ and $\boldsymbol{y}$ are mixed strategies, the update rules both for EG anf for OGDA have to be modified in a manner analogous to equation 1.

For the unconstrained case of bilinear games, the works of [18] and [34] studied rates of convergence for OGDA. However, for the constrained case, even though it has been known that both EG and OGDA converge asymptotically (e.g. proved in [33] for EG and in [41, 29] for OGDA), establishing concrete rates of convergence was stated as an open problem in some recent works, as e.g., in [26]. This was recently resolved in [13], establishing the following more general result.

Theorem 2.6 ([13])**.** *The rate of convergence of both the OGDA and the EG method is $O(\frac{1}{\sqrt{t}})$ in terms of the duality gap, for monotone and Lipschitz Variational Inequality problems (which include the case of bilinear zero-sum games).*

The main innovation in the proof of the above theorem is the use of the *tangent residual*, which is a non-standard performance measure, essentially viewed as an adaptation of the norm of the operator that takes the local constraints into account. The tangent residual is used as a potential function in the analysis, where it is proved that it is non-increasing between two consecutive iterates.

This result was later improved in some cases to $O(1/t)$ in [14]. Although we do not expect significant improvements for the more general cases that this theorem applies to, it is still conceivable that for bilinear zero-sum games, the rate of convergence can be much faster.

2.3.1 Variants of MWU

In a spirit analogous to the way that the basic gradient descent methodology was enhanced by defining the OGDA and EG methods, we could apply the same intuition to the family of MWU algorithms. The optimistic version of MWU, referred to as OMWU was studied first in [19].

The idea of optimism, as before, is to take into account two previous iterations in order to compute the next update, where the extra term can be seen as a negative momentum, correcting the behaviour of MWU dynamics. The dynamics of OMWU are described below, where by $\boldsymbol{x}_i^t$, we denote the i-th coordinate of the mixed strategy $\boldsymbol{x}^t$, with $i \in [n]$.

$$
\begin{aligned}
\boldsymbol{x}_i^t &= \boldsymbol{x}_i^{t-1} \cdot \frac{e^{2\eta \boldsymbol{e}_i^\top R \boldsymbol{y}^{t-1} - \eta \boldsymbol{e}_i^\top R \boldsymbol{y}^{t-2}}}{\sum_{j=1}^n \boldsymbol{x}_j^{t-1} e^{2\eta \boldsymbol{e}_j^\top R \boldsymbol{y}^{t-1} - \eta \boldsymbol{e}_j^\top R \boldsymbol{y}^{t-2}}}, \\
\boldsymbol{y}_j^t &= \boldsymbol{y}_j^{t-1} \cdot \frac{e^{-2\eta \boldsymbol{e}_j^\top R^\top \boldsymbol{x}^{t-1} + \eta \boldsymbol{e}_j^\top R^\top \boldsymbol{x}^{t-2}}}{\sum_{i=1}^n \boldsymbol{y}_i^{t-1} e^{-2\eta \boldsymbol{e}_i^\top R^\top \boldsymbol{x}^{t-1} + \eta \boldsymbol{e}_i^\top R^\top \boldsymbol{x}^{t-2}}}.
\end{aligned}
\tag{5}
$$

Theorem 2.7 ([19])**.** *For zero-sum games that have a unique equilibrium, and for sufficiently small learning rate η, OMWU attains last iterate convergence.*

Proof Outline. We only highlight the main steps that are used, which form a technique that can be useful for analyzing other methods as well. The proof utilizes the following main arguments

- First, it is shown that the KL divergence of $(\boldsymbol{x}^t, \boldsymbol{y}^t)$ to the equilibrium reduces monotonically at a certain rate.
- This implies that at some step t, as long as η is sufficiently small, the profile $(\boldsymbol{x}^t, \boldsymbol{y}^t)$ enters a ball centered at the equilibrium.
- The final step is to show that the dynamics induce a contraction map, by proving that the eigenvalues of the system are bounded by one. This directly implies that eventually OMWU converges to the actual equilibrium.

□

This method was further analyzed in [48], where a convergence rate was obtained. However this rate is instance-dependent, unlike the analysis for OGDA that we presented earlier. It is still open if OMWU can achieve better rates of convergence.

We now present a different method, which was proposed by members of our research team in [23]. We refer to is as Forward Looking Best-Response Multiplicative Weights Update method (FLBR-MWU). We provide first a short description of the main idea behind the dynamics. This is an adaptation of the extra gradient method but applied to MWU, and each iteration has an intermediate and a final step. Suppose that starting from some initial profile, we reach the profile $(\boldsymbol{x}^{t-1}, \boldsymbol{y}^{t-1})$ by the end of iteration $t-1$. In the intermediate step of iteration t, we compute a strategy $\hat{\boldsymbol{x}}^t$ for the row player (resp. $\hat{\boldsymbol{y}}^t$ for the column player), which is an approximate best-response strategy to $\boldsymbol{y}^{t-1}$ (resp. to $\boldsymbol{x}^{t-1}$). This serves as a look ahead step of what would be the currently optimal choices. In the final step of iteration t, we compute the new mixed strategy $\boldsymbol{x}^t$ for the row player, by performing MWU updates, but after assuming that the opponent was playing $\hat{\boldsymbol{y}}^t$.

Formally, the first step of the dynamics is defined below, at iteration t, and for all $i, j \in [n]$, given a non-negative parameter $\xi \in \mathbb{R}^+$ (ξ is chosen sufficiently large).

$$\begin{aligned}
\hat{\boldsymbol{x}}_i^t &= \boldsymbol{x}_i^{t-1} \cdot \frac{e^{\xi \boldsymbol{e}_i^\top R \boldsymbol{y}^{t-1}}}{\sum_{j=1}^n \boldsymbol{x}_j^{t-1} e^{\xi \boldsymbol{e}_j^\top R \boldsymbol{y}^{t-1}}}, \\
\hat{\boldsymbol{y}}_j^t &= \boldsymbol{y}_j^{t-1} \cdot \frac{e^{-\xi \boldsymbol{e}_j^\top R^\top \boldsymbol{x}^{t-1}}}{\sum_{i=1}^n \boldsymbol{y}_i^{t-1} e^{-\xi \boldsymbol{e}_i^\top R^\top \boldsymbol{x}^{t-1}}}.
\end{aligned} \tag{6}$$

The second step, which updates the profile $(\boldsymbol{x}^{t-1}, \boldsymbol{y}^{t-1})$ to $(\boldsymbol{x}^t, \boldsymbol{y}^t)$ is below, given the learning rate parameter $\eta \in (0, 1)$. We assume that we use the same fixed constants η and ξ in all iterations.

$$\begin{aligned}
\boldsymbol{x}_i^t &= \boldsymbol{x}_i^{t-1} \cdot \frac{e^{\eta \boldsymbol{e}_i^\top R \hat{\boldsymbol{y}}^t}}{\sum_{j=1}^n \boldsymbol{x}_j^{t-1} e^{\eta \boldsymbol{e}_j^\top R \hat{\boldsymbol{y}}^t}}, \\
\boldsymbol{y}_j^t &= \boldsymbol{y}_j^{t-1} \cdot \frac{e^{-\eta \boldsymbol{e}_j^\top R^\top \hat{\boldsymbol{x}}^t}}{\sum_{i=1}^n \boldsymbol{y}_i^{t-1} e^{-\eta \boldsymbol{e}_i^\top R^\top \hat{\boldsymbol{x}}^t}}.
\end{aligned} \tag{7}$$

Remark 2.8. By setting $\xi = \eta$ in Equation equation 6 above, the proposed method becomes the same as OMD with entropic regularization [37]. In FLBR-MWU however, η and ξ differ substantially across both theoretical and experimental results.

Theorem 2.9 ([23]). *FLBR-MWU attains last-iterate convergence, as long as* $\eta \cdot \xi < 1$ *and* η *is sufficiently small.*

The proof of this result follows the same technique as the one described in the proof of Theorem 2.7.

2.3.2 Beyond Bilinear Zero-sum Games

There are several interesting generalizations of solving min-max problems beyond the class of bilinear zero-sum games. We have already alluded to some of these generalizations and we summarize below the scenarios that have attracted the attention of the community.

- The payoff function can be a general convex-concave function instead of a bilinear one (convex in the variables corresponding to the first player and concave for the variables corresponding to the second player).
- Even more generally we can have a monotone Variational Inequality problem [22]. These problems are defined by a convex domain D and a monotone Lipschitz operator F. The problem to solve is to find a point $\boldsymbol{z}^*$ so that

  $$< F(\boldsymbol{z}^*), \boldsymbol{z}^* - \boldsymbol{z} > \, \leq \, 0, \quad \forall \boldsymbol{z} \in D$$

  If we take F to be the gradient vector of the payoff function, one can deduce that bilinear zero-sum games fall into this framework.
- Going further, one can also define min-max problems of the form $\min_{\boldsymbol{x}} \max_{\boldsymbol{y}} f(\boldsymbol{x}, \boldsymbol{y})$, where the function f is not convex-concave and does not belong to the previous scenarios.
- Moreover, the profile $(\boldsymbol{x}, \boldsymbol{y})$ could also not be restricted to correspond to mixed strategies of the players. The domain for these variables could be completely unrestricted.
- Finally, one can go beyond two-player zero-sum games to general multi-player games.

Regarding the first two of the points above, several results that have been established for bilinear zero-sum games are easily extended to hold for the more general problem of monotone variational inequality problems. This is demonstrated in the convergence results that have been proved for the OGDA and the EG method in [13]. At the same time, there also exist results for bilinear zero-sum games, for which it is not known where they extend to more general games, such as [19, 23] that concern the methods OMWU and FLBR-MWU presented earlier.

Regarding the remaining cases mentioned above, it has been generally easier to establish convergence results for the unconstrained case, which can also be generalized to classes beyond bilinear games. This does not come as a surprise, and as an example, the work of [49] uses the anchored gradient method in order to show tight convergence results for convex-concave unconstrained games. In [21] further convergence results are established for a class of non-concave problems, using an adaptation of the Extra Gradient method (referred to as EG+). For multi-player games, the Optimistic Gradient is analyzed obtaining positive results in [26]. The picture however is generally more complex for general games and negative results have already been proved in [20].

2.4 Methods based on stochastic gradients

So far in this survey, we have described algorithms that have a deterministic nature. In particular, all the previous results assume that we have *perfect* gradient information. However, this is not always the case for learning algorithms, since there are applications, for instance the training of Generative Adversarial Networks (GANS) [27], where it is impossible, due to the volume of the datasets, to have access to the exact gradient information in every step. The practical solution here is that the algorithms compute an approximation/estimation of the gradient by using a randomly selected sample of the data (i.e., a mini-batch of data points). This is referred to as *stochastic gradient* and one can adapt all the algorithms we have seen so far to this stochastic nature. For example, the analog of Gradient Descent is now the Stochastic Gradient Descent (SGD) method, which is used quite often in practice. In the same spirit, there are stochastic versions of the Extra-Gradient algorithms, the stochastic Extra-Gradient [29], and the same applies to the Optimistic Gradient methods.

To define the problem in more detail, let $F(\boldsymbol{x}, \boldsymbol{y}) = (\nabla_{\boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{y}), -\nabla_{\boldsymbol{y}} f(\boldsymbol{x}, \boldsymbol{y}))$. For algorithms based on stochastic gradients, some of the usual assumptions are that the algorithm has access to an imperfect gradient oracle (referred to typically as a stochastic first order oracle). This means that at iteration t, the oracle returns a noisy gradient of the form $F(\boldsymbol{x}^t, \boldsymbol{y}^t) + Z_t$, where for the noise vector Z_t, we have the following conditions regarding the mean and the variance:

- $E[Z_t|Y_t] = 0$,
- $E[||Z_t||^2|Y_t] \leq (\sigma + k||(\boldsymbol{x}^t, \boldsymbol{y}^t) - (\boldsymbol{x}^*, \boldsymbol{y}^*)||)^2$.

The term Y_t above denotes the history of the profile $(\boldsymbol{x}^t, \boldsymbol{y}^t)$. Therefore, the noise has zero mean, and when $k = 0$ it means that we also have a bounded variance. On the contrary when $\sigma = 0$, it means that the variance is vanishing as we approach the solution set.

It turns out that when adapting the vanilla Extra Gradient method to this stochastic setting, last-iterate convergence is no longer guaranteed. In [16] it is shown that if the noise has unbounded variance then the EG method diverges at a geometric rate. Even further, the work of [30] exhibits new counterexamples, illustrating that we still have non-convergence for any error distribution with positive variance and any step size sequence. Hence, its trajectories remain non-convergent even if the noise is almost surely bounded and a vanishing step size schedule is employed.

To obtain positive convergence results, one needs to exploit additional ideas under the stochastic setting. Towards this direction, [31] showed convergence by using an increasing batch size, which achieves variance reduction. Furthermore, the works of [32, 37] have established convergence almost surely for variations of the EG method, albeit these results hold under stricter assumptions, like strict coherence or strongly monotone operators.

A significant progress on this front was achieved by [30], who showed almost sure convergence under milder assumptions, for the so called Double Stepsize Extra Gradient method. This method is inspired from ideas studied in [50], and is also similar in spirit to the FLBR-MWU method presented earlier. In particular, the intermediate step of the Extra Gradient method utilizes an aggressive learning rate ξ_t at iteration t, whereas the final update step performs a conservative update with learning rate η_t. Under the stochastic setting, the step sizes η and ξ have to vary over time, in contrast to the deterministic setting, where for example a constant η and ξ is employed for the FLBR-

MWU method that we saw before. In particular, the parameters that were used in [30] satisfy that $\sum_t \eta_t \xi_t = \infty$ and $\sum_t \xi_t^2 \eta_t < \infty$.

The latter condition implies that $\eta_t/\xi_t \to 0$, hence it verifies the idea of aggressive exploration and conservative updating. Given this progress, it is expected that such ideas could be very useful for obtaining even stronger results regarding convergence under imperfect gradients.

3 Learning-augmented Mechanism Design

In this section we switch to the second domain that is relevant to project ALGOLINC and concerns mechanism design. In many settings we need to design algorithms for environments with restricted or limited access to information, while still trying to approximate the optimal solution, with regard to the full information, as good as possible. Three main examples of such settings are *Online Algorithms*, *Algorithmic Mechanism Design* and *Computational Social Choice*. In online settings, we have information that is gradually revealed to the algorithm over time and the algorithm has to make instantaneous decisions. The algorithm's performance is then compared to the optimal performance attainable, should the full information and order of appearance was known a priori.

Algorithmic Mechanism Design refers to settings where information is private to strategic agents who may misreport it, should they obtain a better outcome by doing so. A Mechanism is essentially an algorithm that takes into account the strategic nature of the agents (i.e. the source of information) and tries to optimize an objective, either with regards to the agents or with regards to an authority using the mechanism, such as social welfare or revenue maximization. In many cases we are interested in mechanisms that can guarantee certain properties, such as *truthfulness* or *strategyproofness*, i.e. that no agent can gain a better outcome by misreporting their information to the mechanism. The mechanism's performance is then compared to the best attainable outcome, if we had access to the agents' full private information without them having the ability to strategize.

In Algorithmic Social Choice we usually deal with cases where agents do not act strategically but may still have limited information access. The main paradigm studied under this perspective is *Distortion in Voting*. Voting is one of the fundamental problems in Algorithmic Social Choice, where voters express preferences over candidates, and one (or multiple) winner(s) must be selected. Typically, voters provide limited information, such as rankings over candidates, despite having underlying utilities for each outcome, usually attributed to the high cognitive cost on users to fully quantify their preferences. This limited access to information affects the quality of the mechanism's outcome. This loss of efficiency is quantified by the notion of distortion in terms of approximation to the optimal solution [42].

Research problems in these areas have been studied very extensively from the perspective of worst-case analysis, the most well-established approach in algorithm analysis for over half a century. While worst case analysis provides airtight guarantees that our algorithms will never fall short of what's expected, it is a widely accepted fact that they are often unnecessarily pessimistic. This occurs because both lower bounds and impossibility results often only arise due to the inability to algorithmically treat some outlier "non-natural" instances. This key observation has driven a significant research direction of studying algorithms from a "beyond worst case analysis" perspective. The notion of "beyond worst case" has been studied from a bulk of different angles (for an extensive overview see [45]). A very recent one, comes from imposing new advances in artificial intelligence that has provided very efficient algorithms in practice, most of which seem to not be bounded in performance by non-trivial worst-case guarantees.

The seeming gap between worst-case analysis and the performance of machine learning algorithms motivated the agenda of machine learning augmented algorithms via predictions. The idea is that high quality predictions can come as a remedy to the lack of access to information. Essentially the agenda is to design algorithms that can make use of predictions provided by a machine learning system in order to enhance their performance.

Importantly, this line of work is not concerned with the machine learning aspect of producing the predictions, rather than with the design of algorithms that can benefit by using them. The perspective is motivated by the following concepts:

- Create algorithms that are robust, i.e. can still guarantee worst-case performance when predictions are inaccurate, while also achieving consistency guarantees regarding the given predictions, i.e. perform better whenever the predictions are accurate.
- Introduce machine learning power to critical applications for which full explainability of the decision process is required.

So far, most of related work has focused on augmenting online algorithms with predictions. The goal is to be able to guarantee consistency and robustness bounds, making sure that the algorithm will both have a better outcome when the predictions are good but not perform too poorly when the predictions are bad [36]. Usually, a trade-off between consistency and robustness can be established tuned by a factor capturing the degree of trust on the predictions [39]. This agenda received great attention from the community, rapidly creating a relatively large body of related bibliography [35].

3.1 Preliminaries

Before presenting the current state of the art for learning augmented mechanism design with predictions, we provide some basic definitions and notation regarding mechanisms and the modeling of predictions.

3.1.1 Mechanism Design

Mechanisms with money: An offline mechanism $M = (A, p)$ consists of an allocation function $A : \mathbb{R}^n_{\geq 0} \to 2^{[n]}$ and a payment rule $p : \mathbb{R}^n_{\geq 0} \to \mathbb{R}^n_{\geq 0}$. Given a profile of cost declarations $\vec{b} \in [0, B]^n$, where each b_i is the cost *reported* by agent i, the allocation function selects a set of agents $A(\mathbf{b}) \subseteq N$.

The payment rule returns a profile of payments $\vec{p}(\vec{b}) = (p_1, \ldots, p_n)$, where $p_i(\vec{b}) \geq 0$ is the payment to agent $i \in N$. Therefore, the utility of agent $i \in N$ is $u_i(\vec{b}) = p_i(\vec{b}) - c_i$ if $i \in A(\vec{b})$ and $u_i(\vec{b}) = 0$ otherwise, with the understanding that agents not in $A(\vec{b})$ are not paid.

An *online* mechanism, on the other hand, decides the allocation and payment of an agent at every time step without having seen the entire input. That is, given a permutation π on N, the mechanism decides about agent $\pi(t)$ at time step $t \in [n]$, as $\pi(t)$ appears and declares her cost, i.e., at the time the mechanism has only seen $\vec{b}^{\pi}_t = (b_{\pi(1)}, \ldots, b_{\pi(t)}) \in \mathbb{R}^t$. The mechanism, thus, maintains the pair $(A_t(\vec{b}^{\pi}_t), p_t(\vec{b}^{\pi}_t))$ of currently selected agents and their payments, which is consistent with the decisions being irrevocable.

For a *deterministic* mechanism $M = (A, p)$, we require that, for every fixed arrival order, any true cost profile $\vec{c}$ and any declared cost profile $\vec{b}$, the mechanism is:

- *individually rational*; no agent can be hired for less than they asked for, i.e., $p_i(\vec{b}) \geq b_i$, if $i \in A(\vec{b})$.
- *truthful*; no agent $i \in N$ has an incentive to misreport their true cost c_i, i.e., $u_i(c_i, \vec{b}_{-i}) \geq u_i(\vec{b})$.

A *randomized* mechanism $M = (A, p)$ can be thought of as a probability distribution over deterministic mechanisms for every fixed arrival order. Randomized mechanisms that are probability distributions over truthful deterministic mechanisms, are called *universally truthful.*

Mechanisms without money: While the community was initially mostly interested in mechanisms with monetary transfers, i.e. mechanisms that rely on payments to guarantee truthfulness, an array of domains has been identified where payments are not necessary (and maybe not even desirable) to guarantee such properties. Essentially, in such cases, truthfulness comes at the cost of approximation: the mechanism selects the best possible outcome that doesn't allow the agents to strategize. The *facility location problem* was introduced as an initial case study in [43].

The (single) facility location problem: In the single facility location problem we need to allocate a facility on a location that best serves a group of n agents, that lie on a metric space. For brevity we will define the problem for the two-dimensional Euclidean space, as we will also present the learning augmented version by [2].

Specifically, each agent i has a preferred location $p_i \in \mathbb{R}^2$, that is their private information. The goal is to allocate a facility at some location $f \in \mathbb{R}$ in such a way that some objective regarding the agents' welfare is optimized. We consider that, for facility location f, any agent i the agent suffers cost $d(f, p_i)$, which is the Euclidean distance between their preferred location and the allocated facility. For a preference profile $P = (p_1, ..., p_n)$, the most commonly studied social cost functions are:

- The egalitarian social cost: $\mathcal{C}^e(f, P) = \max_{p \in P} d(f, p)$, i.e. the distance between the facility and the location of the agent further away from it.
- The utilitarian social cost: $\mathcal{C}^u(f, P) = \sum_{p \in P} d(f, p)/n$, i.e. the average distance between agents and the facility.

In the strategic version of the problem the agents have the ability to misreport their private information, in order to manipulate the mechanism into allocating the facility closer to their real location. A mechanism for the facility location problem receives the reported (by the agents) preferred locations profile, $P \in \mathbb{R}^{2n}$, and allocates a facility $f(P) \in \mathbb{R}^2$.

A *mechanism* $f : \mathbb{R}^{2n} \longrightarrow \mathbb{R}^2$ is *strategyproof* if no agent can gain by reporting a false location to the mechanism. That is, for all instances $P \in \mathbb{R}^{2n}$ and every agent $i \in [n]$, for every deviation $p'_i \in \mathbb{R}^2$ we have $d(p_i, f(P)) \leq d(p_i, f(P_{-i}, p'_i))$, where (P_{-i}, p'_i) symbolizes the instance profile that occurs if we substitute p_i with p'_i in P.

3.1.2 Distortion in voting

Metric Distortion: A *metric space* is a pair $(\mathcal{M}, d(\cdot, \cdot))$, where $d : \mathcal{M} \times \mathcal{M} \mapsto \mathbb{R}$ is a *metric* on $\mathcal{M}$, i.e., (i) $\forall x, y \in \mathcal{M}, d(x, y) = 0 \iff x = y$ (identity of indiscernibles), (ii) $\forall x, y \in \mathcal{M}, d(x, y) = d(y, x)$ (symmetry), and (iii) $\forall x, y, z \in \mathcal{M}, d(x, y) \leq d(x, z) + d(z, y)$ (triangle inequality). Now consider a set of n voters $V = \{1, 2, \ldots, n\}$, and a set of m

candidates C; we will reference candidates with lowercase letters such as a, b, w, x. Voters and candidates are associated with points in a finite metric space $(\mathcal{M}, d)$, while it is assumed that $\mathcal{M}$ is the (finite) set induced by the set of voters and candidates. The goal is to select a candidate x who minimizes the *social cost*: $SC(x, d) = \sum_{i=1}^{n} d(i, x)$. This task would be trivial if we had access to the agents' distances from all the candidates. However, in the *metric distortion* framework every agent i provides only a *ranking* (a total order) σ_i over the points in C according to the *order* of i's distances from the candidates, with ties broken arbitrarily. We also define $\sigma := (\sigma_1, \ldots, \sigma_n)$, while we will sometimes use $top(i)$ to represent i's most preferred alternative.

A *deterministic social choice rule* is a function that maps an *election* in the form of a 3-tuple $\mathcal{E} = (V, C, \sigma)$ to a single candidate $a \in C$. We will measure the performance of f for a given input of preferences σ in terms of its *distortion*; namely, the worst-case approximation ratio it provides with respect to the social cost:

$$distortion(f; \sigma) = \sup \frac{SC(f(\sigma), d)}{\min_{a \in C} SC(a, d)}, \tag{8}$$

where the supremum is taken over all metrics consistent with the voting profile. The distortion of a social choice rule f is the maximum of $distortion(f; \sigma)$ over all possible input preferences σ:

$$distortion(f) = \sup_{\sigma} \sup_{d: d \rhd \sigma} \frac{SC(f(\sigma, \hat{p}), d)}{SC(c^*(d), d)},$$

Where $d \rhd \sigma$ symbolizes that the preference profile σ is consistent with metric d and $c^*(d)$ is the optimal candidate for d.

To put it differently, once the mechanism selects a candidate (or a distribution over candidates if the social choice rule is *randomized*) an adversary can select any metric space subject to being consistent with the input preferences.

When voters and candidates are embedded in a metric space, as described above, we say that we are studding a *metric distortion* problem. The notion of distortion can also be defined in a more generalized way, where the cost of each voter for any possible candidate can be any value, and the metric properties need not be satisfied. It has been shown that for non-metric distortion, no voting rule has a bounded distortion, while for metric distortion there exists a mechanism achieving distortion of 3, which is also the best possible [25].

3.1.3 Modeling of predictions

In learning augmented mechanisms (or algorithms) predictions are provided as *additional input*, along with the standard input the mechanisms receive for each specific problem in the non-augmented case. The predictions are modeled to provide insight regarding the missing information, that is, *private information* in the case of mechanism design. The specific type of information that predictions represent could vary, depending on what type of input the mechanism designer expects. Most commonly, predictions represent the *private (missing) information*, either in full or partially, or the *optimal solution*.

As mentioned above, predictions can be accurate or inaccurate. *Consistency* and *robustness* capture the two extreme cases in terms of approximation to the optimal solution [36]. Consistency represents the algorithm's performance when the predictions are perfect, while robustness it's performance when the predictions are arbitrarily incorrect.

Apart from the mechanism's behavior in the extreme cases, of perfect accuracy or complete inaccuracy, more refined analysis tries to understand the behavior of the mechanism in the intermediate cases. For this reason a prediction error η is defined, quantifying the disparity between the prediction value and the actual value of the private information it represents. The goal is to analyze the performance of mechanisms with regard to η as a parameter.

3.2 Deterministic, offline settings

3.2.1 Leveraging Predictions for Facility Location

One of the introductory and trend-setting works in learning augmented mechanism design is [2], where the problem of 1-facility location in Euclidean spaces as defined earlier is studied.

Predictions modeling: For this setting the prediction provided to the mechanism is considered to be a prediction of the *optimal solution.* Thus, given a set of (private) preferred optimal locations P, for which the optimal facility location is $o(P)$, the mechanism is provided with a prediction of the optimal solution, $\hat{o}$, before asking the agents to report P. The learning augmented mechanism, $f(P, \hat{o})$, must always still guarantee *strategyproofness* while trying to use $\hat{o}$ to enhance the quality of the facility it selects.

Given a cost function C, we say that a mechanism is α-*consistent* if it is α-approximate when $\hat{o} = o(P)$, i.e. when the prediction is fully correct:

$$\max_{P} \left\{ \frac{C(f(P, o(P)), P)}{C(o(P), P)} \right\} \leq \alpha.$$

The mechanism is said to be β-*robust* if it is β-approximate when the prediction is arbitrarily incorrect:

$$\max_{P,\hat{o}} \left\{ \frac{C(f(P, \hat{o}), P)}{C(o(P), P)} \right\} \leq \beta.$$

Finally, in order to provide a more refined analysis and provide the worst-case approximation ratios as a function of the accuracy of the prediction, the prediction error $\eta \geq$ is defined as follows:

$$\eta(\hat{o}, P) = \frac{d(\hat{o}, o(P))}{C(o(P), P)},$$

I.e. the distance between the predicted optimal solution, $\hat{o}$, and the true optimal solution, $o(P)$, normalized by the social cost in the optimal solution. Given a bound η on the prediction error, we say that our mechanism is $\gamma(\eta)$-approximate if:

$$\max_{P,\hat{o}:\eta(\hat{o},P)\leq\eta} \left\{ \frac{C(f(P, \hat{o}), P)}{C(o(P), P)} \right\} \leq \gamma(\eta).$$

If $\eta = 0$, this bound becomes equivalent to the consistency guarantee, while when $\eta \to \infty$ this bound corresponds to the robustness guarantee.

Egalitarian Social Cost Results: Regarding the egalitarian social cost, for the two-dimensional case, [2] provides a mechanism, called MINIMUM BOUNDING BOX that is 1-consistent and $1 + \sqrt{2}$ robust:

Theorem 3.1 ([2])**.** *The* MINIMUM BOUNDING BOX *mechanism is deterministic, strategyproof, and anonymous and it is* 1*-consistent and* $1+\sqrt{2}$*-robust for the egalitarian social cost.*

The mechanism essentially first determines the *minimum* axis-parallel bounding box that contains all preferred locations P. Then, it allocates the facility on the location within the bounding box that is as close as possible to the predicted optimal solution (that is, if the prediction falls within the bounding box the facility will be located on the prediction, otherwise it will be located on the prediction's projection on the box's borders):

The known (not learning augmented) *Coordinatewise median* mechanism, CM, achieves a 2-approximation for the egalitarian social cost. We see, that in order to guarantee a higher quality outcome when the prediction is good, MINIMUM BOUNDING BOX needs to compromise and provide a worst robustness guarantee than the non augmented algorithm. A matching lower bound shows that in order to guarantee a consistency better than 2, such a compromise is necessary:

Theorem 3.2 ([2])**.** *There is no deterministic, strategyproof, and anonymous mechanism that is* $(2-\epsilon)$*-consistent and* $(1+\sqrt{2}-\epsilon)$*-robust with respect to the egalitarian objective for any* $\epsilon > 0$*.*

Finally, regarding the egalitarian social cost, the results are extended to express the relation between the approximation ratio and the prediction error η:

Theorem 3.3 ([2])**.** *The* MINIMUM BOUNDING BOX *mechanism achieves a* $\min\{1+\eta, 1+\sqrt{2}\}$ *approximation for the egalitarian objective, where* η *is the prediction error.*

The key observation used in order to prove this, is that for a set of preferred location points P and two different predictions $\hat{o}$ and $\tilde{o}$, the distance between the allocated facilities by MINIMUM BOUNDING BOX for each of the predictions is no larger than the distance between the predictions:

$$d(f(P,\hat{o}), f(P,\tilde{o})) \leq d(\hat{o},\tilde{o}).$$

Utilitarian Social Cost Results: For the utilitarian social cost, another very popular form of learning augmented mechanisms via predictions has been used, where we also parameterize the mechanism with a confidence value $c \in [0,1)$ over the prediction. The confidence value is selected by the designer and quantifies how much she trusts the prediction.

The best deterministic non learning augmented mechanism for this problem is the coordinate-wise median mechanism, *CM*, achieving approximation ratio of $\sqrt{2}$. The mechanism simply places the facility at location (x_f, y_f), where x_f is the median of all the x-axis locations of the agents' preferences, $\{x_1, ..., x_n\}$, and y_f the median of all y-axis location.

In order to design a learning augmented algorithm via predictions, tuned by a confidence factor, the authors introduce the COORDINATEWISE MEDIAN WITH PREDICTIONS (CMP) mechanism. The mechanism creates multi-set P', containing cn copies of $\hat{o}$ and then selects as a solution the (generalized) coordinatewise median of the instance $(P \bigcup P')$. It is easy to see that the higher the degree confidence on the prediction, the more the prediction affects the location selected by the mechanism. The performance of the CMP mechanism is determined by the following theorem:

Theorem 3.4 ([2]). *The CMP mechanism with parameter $c \in [0, 1)$ is $\frac{\sqrt{2c^2+2}}{1+c}$-consistent and $\frac{\sqrt{2c^2+2}}{1-c}$-robust for the utilitarian objective.*

It is also proved that the trade-off achieved by *CMP* is optimal. Finally, a theorem bounding the approximation as a factor of the prediction error is also shown, for the utilitarian objective:

Theorem 3.5 ([2]). *The CMP mechanism with parameter $c \in [0, 1)$ achieves a min $\{\frac{\sqrt{2c^2+2}}{1+c} + \eta, \frac{\sqrt{2c^2+2}}{1-c}\}$-approximation, where η is the prediction error, for the utilitarian objective.*

3.2.2 Distortion in Voting

The agenda of learning augmented algorithms via predictions was also introduced to problems in Distortion in Voting in [11].

Predictions modeling: The predictions modeling for this case considers two types of settings:

- The prediction, $\hat{p}$, provided to the mechanism contains the *full missing information*, i.e. all pairwise distances between each voter and each candidate (strongest possible type of predictions).
- $\hat{p}$ is a prediction of who the optimal candidate is.

Consistency and robustness are once again defined in the typical way, described in terms of the predictions in the voting setting: Consider a preference profile σ, a prediction $\hat{p}$ and a metric d. We consider that the prediction is *consistent* with the metric (as the preference profile σ is, $d \rhd (\sigma, \hat{p})$). The algorithm selects a candidate based on σ and $\hat{p}$, $ALG(\sigma, \hat{p})$. Consistency is then the distortion of the algorithm when the prediction is correct:

$$consistency(ALG) = \sup_{\sigma} \sup_{d: d \rhd (\sigma, \hat{p})} \frac{SC(ALG(\sigma, \hat{p}), d)}{SC(c^*(d), d)},$$

Robustness is the distortion the algorithm guarantees when the predictions are arbitrarily incorrect:

$$robustness(ALG) = \sup_{\sigma, \hat{p}} \sup_{d: d \rhd \sigma} \frac{SC(ALG(\sigma, \hat{p}), d)}{SC(c^*(d), d)}.$$

Initially, an impossibility result is established, showing that even in the strongest prediction model, where the predictions represent the full missing information, improvement in consistency (compared to the guaranteed distortion of 3 of non augmented algorithms) can only be achieved with at least some related loss in robustness as follows:

Theorem 3.6 ([11]). *For any $\delta \in [0, 1)$, let ALG be a deterministic algorithm augmented with a prediction regarding the whole metric that is $\frac{3-\delta}{1+\delta}$-consistent. Then, for any $\beta < \frac{3+\delta}{1-\delta}$, ALG is not β-robust.*

Flowing that, the authors define a family of learning augmented algorithms via predictions, LA_δ, parameterized by a *confidence parameter* $\delta \in [0, 1)$, that the designer can chose depending on their confidence in the prediction. LA_δ algorithms receive a prediction of *only who the optimal candidate is*, $\hat{p} \in C$ - the second, much weaker prediction input described above.

Despite the much more limited information provided to the algorithm through the prediction, it is shown that LA_δ algorithms achieve the best possible trade-off between consistency and robustness for algorithms provided with predictions regarding the full information, as shown in 3.6[2]:

Theorem 3.7 ([11]). *For any $\delta \in [0, 1)$, LA_δ achieves $\frac{3-\delta}{1+\delta}$-consistency and $\frac{3+\delta}{1-\delta}$-robustness.*

3.2.3 Other problems in deterministic mechanism design

Although the facility location and the distortion settings have attracted significant attention from the community, there are also other problems that have been studied in terms of the design of deterministic mechanisms. Some prominent examples include *Strategic Scheduling* [9] and Auctions [8]. For an extensive list we refer to [35].

3.3 Online Mechanism Design with Predictions

In this section we will move to the setting of online mechanism design with predictions, which was recently introduced through [10], and brought together the classical setting of online algorithmic problems with predictions and mechanism design with predictions. The problem that was studied concerns selling goods to strategic bidders that arrive and depart over time.

More specifically, the problem considered was that of designing an auction for a single item to a set $N = 1, 2, \ldots, n$, of n bidders who arrive and depart over time. Each bidder i has an arriving time a_i, a departure time $d_i \geq a_i$ and a value v_i for the item that is auctioned. The interval $[a_i, d_i]$ corresponds to the time that agent i is active. These three, define the type $\theta_i = (a_i, d_i, v_i)$ of an agent i. For simplicity, the bidders are indexed based on their departure order (i.e., bidder i is the i-th bidder to depart). Moreover, π is an arbitrary total order over the set of bidders, which is used for tie-breaking, and let $i \succ j$ denote the fact that i is ranked before bidder j according to π. The goal is to maximize the revenue from the sale of the item. The difficulty of the problem, comes from the fact that the auctioneer does not have access to the private information of the agents (i.e. their arrival time a_i, their departure time d_i and their value v_i). The bidders can misreport their information and the auction must be designed in a way, so that they cannot benefit by doing so. Agents can misreport all the pieces of their private information, their value v_i and arrival and departure times. Based on the model firstly introduced by Hajiaghayi et al. [28], each bidder i can delay the announcement of her arrival (reporting a delayed arrival time $\hat{a}_i > a_i$), and she can report a false departure time $\hat{d}_i$ (either earlier or later than her true departure time, d_i). Upon arrival, each bidder i declares a type $\hat{\theta}_i$ (potentially different than θ_i) and the mechanism needs to decide, to which agent will the item be allocated, at what time t the item should be allocated to the winner, as well as the payment p that the winner will pay. Since the mechanism is online, this means that if it decides to allocate the item at some time t, then this decision is irrevocable, and both this allocation decision and the payment amount requested from the winner can depend only on information regarding bidders with arrival time $a_i \leq t$.

If the auction allocates the item to some bidder i^* at a time t for a price of p, then the utility of agent i^* is $v_{i^*} - p$, given that i^* is active at time t (i.e. $t \in [a_{i^*}, d_{i^*}]$). In the case that, at the time the item is allocated, agent i^* is inactive, then she receives no value from it and her utility is equal to $-p$. All the other agents do not receive the

[2]Notice that given a full information predictions profile we can deduce the optimal candidate prediction, but a specific optimal candidate could be deduced by infinite preference profiles.

item and contribute no payment, so their utility is 0. A mechanism is strategyproof if for every bidder i, it is a dominant strategy to report her true profile θ_i. This means that independently of what the other agents report, the utility of an agent i is maximized through truthfully reporting her type θ_i. The fact that in this setting, agents can also misreport the interval in which they are active, adds an extra difficulty of the problem compared to the offline counterpart, where agents can only misreport their value.

Given a set $I = \{[a_1, d_1], [a_2, d_2], \ldots, [a_n, d_n]\}$ of n arrival-departure intervals and a set V of n values generated adversarially, and the values of V are then matched to arrival-departure intervals from I uniformly at random, let $\mu(V, I)$ denote this random matching. Then, $\mathbb{E}_{\Theta\sim\mu(V,I)}[Rev(M(\Theta))]$ denotes the expected revenue of an auction M with respect to this random matching. Also, $v_{(1)}$ and $v_{(2)}$ denote the highest and second-highest values in V , the former is the highest feasible revenue and the latter corresponds to the amount of revenue that is actually achievable via the classic Vickrey auction in offline settings. Mechanisms are also equipped with a (possibly erroneous) prediction $\tilde{v}_{(1)}$ regarding the highest value, $v_{(1)}$, in V . The expected revenue of a mechanism, M, enhanced with a prediction, is denoted as $\mathbb{E}_{\Theta\sim\mu(V,I)}[Rev(M(\Theta, \tilde{v}_{(1)}))]$ and the performance of M is evaluated, with respect to its consistency and robustness. Consistency refers to the competitive ratio of the expected revenue achieved by the algorithm when the prediction it is provided with is accurate, i.e., whenever $\tilde{v}_{(1)} = v_{(1)}$. The benchmark we use for consistency is the highest value in V, often referred to as the first-best revenue. More formally:

$$consistency(M) = \min_{V,I} \frac{\mathbb{E}_{\Theta\sim\mu(V,I)}[Rev(M(\Theta, v_{(1)}))]}{v_{(1)}}.$$

The robustness of the mechanism, is the competitive ratio of the expected revenue given an adversarially chosen, inaccurate, prediction. The benchmark we use for robustness is the best revenue achievable via any (offline) strategyproof auction, which is the second highest value $v_{(2)}$. Below, the robustness ratio is formally defined:

$$robustness(M) = \min_{V,I,\tilde{v}_{(1)}} \frac{\mathbb{E}_{\Theta\sim\mu(V,I)}[Rev(M(\Theta, \tilde{v}_{(1)}))]}{v_{(2)}}.$$

3.3.1 The Three Phase Auction Mechanism

The mechanism proposed for the problem, in the work of Balkanski et al. [10], was a three phase auction, parameterized by a parameter $\alpha \in [0, 1]$, which corresponds to the confidence in the accuracy of the prediction (higher value means greater confidence). The THREE-PHASE auction mechanism, for any choice of α achieves α-consistency and $\frac{1-\alpha^2}{4}$-robustness and also is both value- strategyproof and time-strategyproof.

The Three-Phase auction considers the bidders based on the order of their departure and consists of three separate phases.

Phase 1: During the first phase, the auction samples the values of the first $\lceil \frac{1-\alpha}{2} n \rceil$ bidders to depart (without allocating the item to any of them). The aim of the sampling, is to "learn" an estimate of what a reasonable price for the item may be. If, during this first phase, the auction observes a value that exceeds the predicted maximum, $\tilde{v}_{(1)}$ (implying that the prediction is inaccurate), then, the first phase is completed, the auction skips the second phase and moves directly to the third one. When the first phase does not prove the prediction to be inaccurate, then the auction proceeds to the second phase.

Phase 2: During the second phase,which terminates after $\lfloor \alpha n \rfloor$ more bidders have departed, it asks all active bidders whether they would accept to pay a price equal to the prediction. If any active bidder is willing to pay this price, then they secure the item and they are guaranteed, to pay a price no more than that. However, the exact payment of bidders who secure the item during the second phase, may need to be lower than that, in order to guarantee strategyproofness. Finally, if none of the $\lfloor an \rfloor$ bidders is willing to pay a price equal to the prediction during the second phase, then the auction enters its third phase.

Phase 3: During the third phase, the mechanism offers a take-it-or-leave-it price equal to the highest value that has been observed during the two previous phases, and any active bidder can claim the item at that price.

After the termination of the THREE-PHASE auction, the payment rule is executed, in order to determine how much the winner, if any, should pay for the item. The price is initially set to be equal to the threshold τ at which the item was secured and the final price will be no more than that. As has already been mentioned, in some cases, the price is reduced in order to guarantee strategyproofness. Specifically, if the winner secured the item during the second phase and remains active during the third phase, they may receive a lower price. In this case, the price is determined by simulating the allocation process without the winning bidder, i^*. Should the new winner i', in the absence of i^*, either i) is not active during the transition into the third phase or ii) loses to i^* in tie-breaking, then the price p is lowered to the threshold τ at which i' would have secured the item. Intuitively, if neither of these two conditions holds and we did not offer i^* the reduced price, then i^* could report a value equal to τ' instead of her true value and secure the item at that lower price, right after the transition into the third phase.

The following theorem, summarizes the performance guarantees of the THREE-PHASE auction mechanism, with respect to consistency and robustness, and also with respect to value-strategyproofness and time-strategyproofness.

Theorem 3.8 ([10])**.** *The THREE-PHASE mechanism is a value-strategyproof and time-strategyproof online auction that, given any parameter $\alpha \in [0, 1]$, guarantees α-consistency and $\frac{1-\alpha^2}{4}$-robustness.*

3.4 Randomized Mechanism Design with Predictions

Now, we will move to the setting of randomized mechanisms with predictions, which was first introduced in [15]. As in the previous subsection, the problem studied was the design of truthful single-item offline auction and the objective is to get revenue as close as possible to the highest agent valuation. They study a class of randomized auctions, which they call intuitive auctions. An intuitive auction is anonymous, meaning that the outcome does not depend on the identifiers of the agents in any way and, has the property that only the highest bidder(s) get the item with positive probability.

In single-item auctions, as mentioned in the previous subsection, there is a set of n agents (or bidders) bidding for a single item. Each agent i has a private valuation v_i for that item. The valuations of the agents belong to the interval $[1, H]$ for $H > 1$. An auction mechanism receives bids from the agents as input and decides to which agent, the item will be allocated and the payment that will be received from each agent.

The term "revenue" refers to the total payment received by all agents. The goal is to design auctions that maximize revenue. Ideally, an auction applied to agents with highest valuation t, should extract a revenue as close to t as possible. We use the revenue-to-highest-valuation ratio as the primary objective. In this context, they assume access to prediction $\hat{u}$ for the highest valuation.

For parameters γ and ρ, the aim is to design auctions that achieve a revenue-to-highest-valuation ratio of at least γ when the prediction is accurate and ρ when it is not. Such auctions are said to have a consistency of γ and a robustness of ρ. Alternatively, robustness can be described as a function of the prediction error, $\max \frac{t}{\hat{u}}, \frac{\hat{u}}{t}$, which quantifies how far the highest valuation t is from the prediction $\hat{u}$. The robustness requirement can be defined by a non-increasing function $\rho : [1, H] \to [0, 1]$, ensuring a revenue of at least $\rho\left(\max \frac{t}{\hat{u}}, \frac{\hat{u}}{t}\right) \cdot t$.

In this context, they provide sufficient and necessary conditions for the existence of consistent and robust auctions, which are given through the following theorem.

Theorem 3.9 ([15]). *Let $0 \leq \rho \leq \gamma \leq 1$. There exists a γ-consistent and ρ-robust intuitive auction for bidders with valuations from $[1, H]$ and a prediction for the highest bid $\hat{u} \in [1, H]$, if and only if $\gamma + \rho \cdot \ln \max\{\hat{u}, \frac{H \cdot \rho}{\gamma}\} \leq 1$*

As a corollary of the above theorem they obtain that auctions with constant consistency and robustness $\Omega(\ln^{-1} H)$ exist, for the parameters $\gamma = \frac{1}{2}$ and $\rho = \frac{1}{2(1+\ln H)}$, which satisfy the conditions of the theorem.

Moreover, they also provide a sufficient and necessary condition for intuitive auctions,when robustness is expressed as a function ρ, with $\rho(\eta)$ indicating the required lower bound on the revenue-over-highest-valuation ratio extracted by the auction in all valuation profiles with prediction error η. This is formally stated in the following theorem.

Theorem 3.10 ([15]). *Let $\rho : [1, H] \to [0, 1]$ be a differentiable function so that $\rho(\eta)$ is non-increasing and $\eta \cdot \rho(\eta)$ is non-decreasing in η. There exists a ρ-robust intuitive auction for bidders with valuations from $[1, H]$ and a prediction for the highest bid $\hat{u} \in [1, H]$ if and only if:*

$$\rho(H/\hat{u}) + \int_1^{\hat{u}} \frac{\rho(z)}{z} dz + \int_1^{H/\hat{u}} \frac{\rho z}{z} dz \leq 1.$$

4 Conclusions and summary of open problems

We summarize here some of the research directions and open problems that also form the main source of inspiration for project ALGOLINC. We describe separately the problems pertaining to the material of Section 2 and the material of Section 3.

4.1 Open problems from learning in games

- For bilinear zero-sum games, it is still an open problem to determine the best rate of convergence. Can we have learning algorithms that attain last-iterate convergence and have a geometric rate w.r.t. the duality gap?
- For the method FLBR-MWU, which was presented earlier, it is an interesting topic for future work, to examine adaptive schemes for ξ and η throughout the iterations. We note that so far we only know asymptotic convergence for this method, and hence it is not yet clear if we can achieve any concrete convergence rates.
- A general direction that has also attracted a lot of attention is to be able to generalize convergence results beyond zero-sum games. A first approach here would be to focus on low-rank games. For general games we are not aware of any learning algorithm that converges to a Nash equilibrium.
- Most of the previous questions also apply to the stochastic setting. Can we have analogous progress when we have access to imperfect gradient information?

4.2 Open problems from learning-augmented mechanism design

- As we have seen, a line of work is emerging in learning augmented mechanism design, for the single item forward auction setting. An interesting future direction would be to see, if these results can be extended in the multi-unit auction setting.
- Additionally, another setting that can be enhanced with predictions, is that of reverse auctions, namely procurement auctions under a budget constraint both for the offline and online case, which was originally studied in [46, 17, 4, 6]. This concerns the design of mechanisms that are budget-feasible and truthful, and on top of that could achieve satisfactory consistency-robustness trade-offs under an appropriate prediction model.
- Distortion in voting is also another field where interesting questions can be addressed, regarding using learning augmented techniques. A first idea would be to try and use predictions for non-metric settings. As mentioned above, no deterministic algorithm has bounded distortion, when limited to only the ordinal information. For this reason some works focus on how many cardinal (i.e. utility) queries are required in order to guarantee bounded distortion [3]. The possible predictions-related questions can follow two directions here: a) what happens if the cardinal queries are predictions, i.e. have some uncertainty instead of providing always the correct value? b) Can we use predictions regarding the missing information to minimize the query complexity in order to achieve bounded distortion?
- Similar to the point above, multi-winner voting settings can be studied as well. For more than one winners even metric distortion is unbounded in most cases. Cardinal queries could also be used to overcome these negative results. Thus, questions a) and b) from the previous paragraph would also apply here.
- Another possible direction for future research could be expanding the results for 1-facility location to multiple facility location problems. k-facility location problems for $k > 2$ have very well known and restrictive negative results even for the simplest metrics. Can predictions help reach some better outcomes in some cases, maybe even in combination with some other beyond worst-case analysis techniques?

References

[1] Ilan Adler. The equivalence of linear programs and zero-sum games. *Int. J. Game Theory*, 42(1):165–177, 2013.

[2] Priyank Agrawal, Eric Balkanski, Vasilis Gkatzelis, Tingting Ou, and Xizhi Tan. Learning-augmented mechanism design: Leveraging predictions for facility location. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, EC '22, page 497–528, New York, NY, USA, 2022. Association for Computing Machinery.

[3] Georgios Amanatidis, Georgios Birmpas, Aris Filos-Ratsikas, and Alexandros A. Voudouris. Peeking behind the ordinal curtain: Improving distortion via cardinal queries. *Artificial Intelligence*, 296:103488, 2021.

[4] Georgios Amanatidis, Pieter Kleer, and Guido Schäfer. Budget-feasible mechanism design for non-monotone submodular objectives: Offline and online. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pages 901–919, 2019.

[5] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory Comput.*, 8(1):121–164, 2012.

[6] Ashwinkumar Badanidiyuru, Robert Kleinberg, and Yaron Singer. Learning on a budget: posted price mechanisms for online procurement. In *Proceedings of the 13th ACM conference on electronic commerce*, pages 128–145, 2012.

[7] James P. Bailey and Georgios Piliouras. Multiplicative weights update in zero-sum games. In *Proceedings of the Conference on Economics and Computation (EC'18)*, pages 321–338, 2018.

[8] Maria-Florina Balcan, Siddharth Prasad, and Tuomas Sandholm. Bicriteria multidimensional mechanism design with side information. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

[9] Eric Balkanski, Vasilis Gkatzelis, and Xizhi Tan. Strategyproof scheduling with predictions. 2023.

[10] Eric Balkanski, Vasilis Gkatzelis, Xizhi Tan, and Cherlin Zhu. Online mechanism design with predictions. *arXiv preprint arXiv:2310.02879*, 2023.

[11] Ben Berger, Michal Feldman, Vasilis Gkatzelis, and Xizhi Tan. Learning-augmented metric distortion via (p,q)-veto core. In *Proceedings of the 25th ACM Conference on Economics and Computation*, EC '24, page 984, New York, NY, USA, 2024. Association for Computing Machinery.

[12] George W. Brown. Iterative solution of games by fictitious play. In *Activity Analysis of Production and Allocation*. 1951.

[13] Y. Cai, A. Oikonomou, and W. Zheng. Finite-time last-iterate convergence for learning in multi-player games. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS'22)*, 2022.

[14] Yang Cai and Weiqiang Zheng. Doubly optimal no-regret learning in monotone games. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 3507–3524. PMLR, 2023.

[15] Ioannis Caragiannis and Georgios Kalantzis. Randomized learning-augmented auctions with revenue guarantees. *arXiv preprint arXiv:2401.13384*, 2024.

[16] Tatjana Chavdarova, Gauthier Gidel, François Fleuret, and Simon Lacoste-Julien. Reducing noise in GAN training with variance reduced extragradient. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, pages 391–401, 2019.

[17] Ning Chen, Nick Gravin, and Pinyan Lu. On the approximability of budget feasible mechanisms. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 685–699. SIAM, 2011.

[18] Constantinos Daskalakis, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng. Training GANs with optimism. In *Proceedings of the International Conference on Learning Representations (ICLR'18)*, 2018.

[19] Constantinos Daskalakis and Ioannis Panageas. Last-iterate convergence: Zero-sum games and constrained min-max optimization. In *Proceedings of the ITCS'19*, 2019.

[20] Constantinos Daskalakis, Stratis Skoulakis, and Manolis Zampetakis. The complexity of constrained min-max optimization. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 1466–1478. ACM, 2021.

[21] Jelena Diakonikolas, Constantinos Daskalakis, and Michael I. Jordan. Efficient methods for structured nonconvex-nonconcave min-max optimization. In Arindam Banerjee and Kenji Fukumizu, editors, *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pages 2746–2754. PMLR, 2021.

[22] Francisco Facchinei and Jong-Shi Pang. *Finite-dimensional variational inequalities and complementarity problems.* Springer,, 2003.

[23] M. Fasoulakis, E. Markakis, Y. Pantazis, and C. Varsos. Forward looking best-response multiplicative weights update methods for bilinear zero-sum games. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'22)*, pages 11096–11117, 2022.

[24] Yoav Freund and Robert E. Schapire. Adaptive Game Playing Using Multiplicative Weights. *Games and Economic Behavior*, 29(1-2):79–103, 1999.

[25] Vasilis Gkatzelis, Daniel Halpern, and Nisarg Shah. Resolving the optimal metric distortion conjecture. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1427–1438, 2020.

[26] Noah Golowich, Sarath Pattathil, and Constantinos Daskalakis. Tight last-iterate convergence rates for no-regret learning in multi-player games. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[27] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Proceedings of Annual Conference on Neural Information Processing Systems (NIPS '14)*, pages 2672–2680, 2014.

[28] Mohammad Taghi Hajiaghayi, Robert Kleinberg, and David C Parkes. Adaptive limited-supply online auctions. In *Proceedings of the 5th ACM Conference on Electronic Commerce*, pages 71–80, 2004.

[29] Yu-Guan Hsieh, Franck Iutzeler, Jérôme Malick, and Panayotis Mertikopoulos. On the convergence of single-call stochastic extra-gradient methods. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 6936–6946, 2019.

[30] Yu-Guan Hsieh, Franck Iutzeler, Jérôme Malick, and Panayotis Mertikopoulos. Explore aggressively, update conservatively: Stochastic extragradient methods with variable stepsize scaling. In *Proceedings of Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020.

[31] Alfredo N. Iusem, Alejandro Jofré, Roberto Imbuzeiro Oliveira, and Philip Thompson. Extragradient method with variance reduction for stochastic variational inequalities. *SIAM J. Optim.*, 27(2):686–724, 2017.

[32] Aswin Kannan and Uday V. Shanbhag. Optimal stochastic extragradient schemes for pseudomonotone stochastic variational inequality problems and their variants. *Comput. Optim. Appl.*, 74(3):779–820, 2019.

[33] G. Korpelevich. The extragradient method for finding saddle points and other problems. *Matecon*, 12:747–756, 1976.

[34] Tengyuan Liang and James Stokes. Interaction matters: A note on non-asymptotic local convergence of generative adversarial networks. In *Proceedings of The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS'19*, pages 907–915, 2019.

[35] Alexander Lindermayr, Nicole Megow, Bertrand Simon, Adam Polak, and Niklas Hahn. Algorithms with predictions site (ALPS). `https://algorithms-with-predictions.github.io/`.

[36] Thodoris Lykouris and Sergei Vassilvitskii. Competitive caching with machine learned advice. *J. ACM*, 68(4), July 2021.

[37] Panayotis Mertikopoulos, Bruno Lecouat, Houssam Zenati, Chuan-Sheng Foo, Vijay Chandrasekhar, and Georgios Piliouras. Optimistic mirror descent in saddle-point problems: Going the extra (gradient) mile. In *Proceedings of the 7th International Conference on Learning Representations, ICLR'19*, 2019.

[38] Panayotis Mertikopoulos, Christos H. Papadimitriou, and Georgios Piliouras. Cycles in adversarial regularized learning. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018*, pages 2703–2717. SIAM, 2018.

[39] Michael Mitzenmacher and Sergei Vassilvitskii. Algorithms with predictions. *CoRR*, abs/2006.09123, 2020.

[40] J. Nash. Non-cooperative games. *Annals of Mathematics*, 54 (2), 1951.

[41] L. Popov. A modification of the Arrow-Hurwicz method for search of saddle points. *Mathematical notes of the Academy of Sciences of the USSR*, 28:845–848, 1980.

[42] Ariel D. Procaccia and Jeffrey S. Rosenschein. The distortion of cardinal preferences in voting. In Matthias Klusch, Michael Rovatsos, and Terry R. Payne, editors, *Cooperative Information Agents X*, pages 317–331, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[43] Ariel D. Procaccia and Moshe Tennenholtz. Approximate mechanism design without money. In *Proceedings of the 10th ACM Conference on Electronic Commerce*, EC '09, page 177–186, New York, NY, USA, 2009. Association for Computing Machinery.

[44] J. Robinson. An iterative method of solving a game. *Annals of Mathematics*, pages 296–301, 1951.

[45] Tim Roughgarden. *Beyond the Worst-Case Analysis of Algorithms*. Cambridge University Press, 2021.

[46] Yaron Singer. Budget feasible mechanisms. In *2010 IEEE 51st Annual Symposium on foundations of computer science*, pages 765–774. IEEE, 2010.

[47] J. Von Neumann. Zur theorie der gesellschaftsspiele. *Math. Ann.*, 100:295–320, 1928.

[48] Chen-Yu Wei, Chung-Wei Lee, Mengxiao Zhang, and Haipeng Luo. Linear last-iterate convergence in constrained saddle-point optimization. In *Proceedings of the 9th International Conference on Learning Representations ICLR '21*, 2021.

[49] Taeho Yoon and Ernest K. Ryu. Accelerated algorithms for smooth convex-concave minimax problems with o(1/kˆ2) rate on squared gradient norm. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 12098–12109. PMLR, 2021.

[50] Guojun Zhang and Yaoliang Yu. Convergence of gradient methods on bilinear zero-sum games. In *8th International Conference on Learning Representations, ICLR 2020*. OpenReview.net, 2020.