Inducing Equilibria via Incentives: Simultaneous Design-and-Play Ensures Global Convergence

Anonymous Author(s)
Affiliation
Address
email

Abstract

To regulate a social system comprised of self-interested agents, economic incentives are often required to induce a desirable outcome. This incentive design problem naturally possesses a bilevel structure, in which a designer modifies the payoffs of the agents with incentives while anticipating the response of the agents, who play a non-cooperative game that converges to an equilibrium. The existing bilevel optimization algorithms raise a dilemma when applied to this problem: anticipating how incentives affect the agents at equilibrium requires solving the equilibrium problem repeatedly, which is computationally inefficient; bypassing the time-consuming step of equilibrium-finding can reduce the computational cost, but may lead the designer to a sub-optimal solution. To address such a dilemma, we propose a method that tackles the designer's and agents' problems simultaneously in a single loop. Specifically, at each iteration, both the designer and the agents only move one step. Nevertheless, we allow the designer to gradually learn the overall influence of the incentives on the agents, which guarantees optimality after convergence. The convergence rate of the proposed scheme is also established for a broad class of games.

1 Introduction

A common thread in human history is how to "properly" regulate a social system comprised of self-interested individuals. In a laissez-faire economy, for example, the competitive market itself is the primary regulatory mechanism [49, 16]. However, a laissez-faire economy may falter due to the existence of significant "externalities" [6, 18], which may arise wherever the self-interested agents do not bear the external cost of their behaviors in the entirety. The right response, many argue, is to introduce corrective policies in the form of economic incentives (e.g., tolls, taxes, and subsidies) [32]. By modifying the payoffs of the agents, these incentives can encourage (discourage) the agents to engage in activities that create positive (negative) side effects for the society, and thus guide the self-interests of the agents towards a socially desirable end. For example, carbon taxes can be levied on carbon emissions to protect the environment during the production of goods and services [35]. Surge pricing has been widely used to boost supply and dampen demand in volatile ride-hail markets [37]. Lately, subsidies and penalties were both introduced to overcome vaccine hesitancy in the world's hard-fought battle against the COVID-19 pandemic.

The study of such incentives can be traced back to Pigou's congestion toll for correcting the inefficient use of roads jammed by self-interested drivers [42]. Over the past decades, economic incentives have been investigated in various systems, e.g., energy [7], transportation [24], manufacturing [31], and education [13]. It has become a research theme in machine learning more recently. For example, Roth et al. [46] studied an incentive design scheme from revealed preferences feedback and applied it to the design of congestion tolls. Mguni et al. [37] and Zheng et al. [55] studied designing incentives in

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

multi-agent reinforcement learning systems and highlighted the potential applications to ride-sharing systems and economic systems, respectively. Li et al. [25] proposed an end-to-end optimization approach for solving incentive design problems cast as a mathematical program with equilibrium constraints (MPEC) [19]. The approach is tested for designing subsidies in public transit systems.

In this paper, we focus on designing incentives in games, hence we also formulate the incentive design problem as an MPEC. Such an MPEC naturally possesses a bilevel structure [3, 29, 9], in which an upper-level "designer" chooses the incentives while anticipating the response of the agents, who play a non-cooperative game at a lower level. The resulting problem is difficult to solve because even getting a first-order derivative through the bilevel structure is a challenge. To calculate the gradient of the designer's objective at equilibrium, which provides a principled direction for the designer to update the incentives, one must *anticipate* how the equilibrium is affected by the changes [17]. This is usually achieved by performing a sensitivity analysis, which in turn requires differentiation through the lower-level equilibrium problem, either implicitly or explicitly [25]. No matter how the sensitivity analysis is carried out, the equilibrium problem must be *solved* before updating the incentives. The resulting algorithm thus admits a double loop structure: in the outer loop, the designer iteratively moves along the gradient; but to find the gradient, it must allow the lower level game dynamics to run its course to arrive at the equilibrium given the current incentives.

Because of the inherent inefficiency of the double-loop structure, many heuristics methods have also been developed for bilevel programs in machine learning [26, 28, 14]. When applied to the incentive design problem, these methods assume that the designer *does not solve* the equilibrium exactly to evaluate the gradient. Instead, at each iteration, the game is allowed to run just a few rounds, enough for the designer to obtain a reasonable approximation of the gradient. Although such a method promises to reduce the computational cost significantly at each iteration, it may never converge to the same optimal solution obtained without the approximation.

Contribution. In a nutshell, correctly anticipating how incentives affect the agents at equilibrium requires solving the equilibrium problem repeatedly, which is computationally inefficient. On the other hand, simply bypassing the time-consuming step of equilibrium-finding may lead the designer to a sub-optimal solution. This dilemma prompts the following question that motivates this study: *can we obtain the optimal solution to an incentive design problem without repeatedly solving the equilibrium problem?*

In this paper, we propose an efficient principled method that tackles the designer's problem and agents' problem simultaneously in a single loop. At the lower level, we use the *mirror descent* method [40] to model the process through which the agents move towards equilibrium. At the upper level, we use the *gradient descent* method to update the incentives towards optimality. At each iteration, both the designer and the agents only move one step based on the first-order information. However, as discussed before, the upper gradient relies on the corresponding lower equilibrium, which is not available in the single-loop update. Hence, we propose to use the implicit differentiation formula—with equilibrium strategy replaced by the current strategy—to estimate the upper gradient, which might be biased at the beginning. Nevertheless, we prove that if we improve the lower-level solution with larger step sizes, the upper-level and lower-level problems may converge simultaneously at a fast rate. The proposed scheme hence guarantees optimality because it can anticipate the overall influence of the incentives on the agents eventually after convergence.

Organization. In Section2, we discuss related work. In Section 3, we provide the mathematical formulation of the incentive design problem. In Section 4, we design algorithms for solving the problem. In Section 5, we establish conditions under which the proposed scheme globally converges to the optimal solution and analyze the convergence rate. The convergence analysis is restricted to games with a unique equilibrium. In Section 6, we extend our work to games with multiple equilibria. Eventually, we conduct experiments to test our algorithms in Section 7.

Notation. We denote $\langle \cdot, \cdot \rangle$ as the inner product in vector spaces. For a vector form $a = (a^i)$, we denote $a^{-i} = (a^j)_{j \neq i}$. For a finite set $\mathcal{X} \in \mathbb{R}^n$, we denote $\Delta(\mathcal{X}) = \{\pi \in \mathbb{R}^n_+ : \sum_{x^i \in \mathcal{X}} \pi_{x^i} = 1\}$. For any vector norm $\| \cdot \|$, we denote $\| \cdot \|_* = \sup_{\|z\| \leq 1} \langle \cdot, z \rangle$ as its dual norm.

2 Related work

In this paper, we formulate the incentive design problem as an MPEC. MPEC is closely related to bilevel programs [9], which bind two mathematical programs together by treating one program as part of the constraints for the other. Below we would introduce both related bilevel optimization and MPEC algorithms.

Bilevel optimization. In the optimization literature, bilevel programming was first introduced to tackle resource allocation problems [5] and has since found applications in such diverse topics as revenue management, network design, traffic control, and energy systems. In the past decade, researchers have discovered numerous applications of bilevel programming in machine learning, including meta-learning (ML) [14], adversarial learning [22], hyperparameter optimization, [30] and neural architecture search [26]. These newly found bilevel programs in ML are often solved by gradient descent methods, which require differentiating through the (usually unconstrained) lower-level optimization problem [27]. The differentiation can be carried out either implicitly on the optimality conditions as in the conventional sensitivity analysis [see e.g., 2, 44, 4], or explicitly by unrolling the numerical procedure used to solve the lower-level problem [see e.g., 30, 15]. In the explicit approach, one may "partially" unroll the solution procedure (i.e., stop after just a few rounds, or even only one round) to reduce the computational cost. Although this popular heuristic has delivered satisfactory performance on many practical tasks [28, 36, 14, 26], it cannot guarantee optimality for bilevel programs under the general setting, as it cannot derive the accurate upper-level gradient at each iteration [54].

MPEC. Unlike bilevel programs, MPEC is relatively under-explored in the ML literature so far. Recently, Li et al. [25] extended the explicit differentiation method for bilevel programs to MPECs. Their algorithm unrolls an iterative projection algorithm for solving the lower-level problem, which is formulated as a variational inequality (VI) problem. Leveraging the recent advance in differentiable programming [2], they embedded each projection iteration as a differentiable layer in a computational graph, and accordingly, transform the explicit differentiation as standard backpropagation through the graph. Although backpropagation is efficient, constructing and storing such a graph—with potentially a large number of projection layers needed to find a good solution to the lower-level problem—is still demanding. To reduce this burden, partially unrolling the iterative projection algorithm is a solution. Yet, it still cannot guarantee optimality for MPECs due to the same reason as for bilevel programs.

The simultaneous design-and-play approach is proposed to address this dilemma. Our approach follows the algorithm of Hong et al. [21] and Chen et al. [8], which solves bilevel programs via single-loop update. Importantly, they solve both the upper- and the lower-level problem using a gradient descent algorithm and establish the relationship between the convergence rate of the single-loop algorithm and the step size used in gradient descent. However, their algorithms are limited to the cases where the lower-level optimization problem is *unconstrained*. Our work extends these single-loop algorithms to MPECs that have an equilibrium problem at the lower level. We choose mirror descent as the solution method to the lower-level problem because of its broad applicability to optimization problems with constraints [40] and generality in the behavioral interpretation of games [34, 23]. We show that the convergence of the proposed simultaneous design-and-play approach relies on the setting of the step size for both the upper- and lower-level updates, a finding that echos the key result in [21]. We first give the convergence rate under mirror descent and the unconstrained assumption and then extend the result to the constrained case. For the latter, we show that convergence cannot be guaranteed if the lower-level solution gets too close to the boundary early in the simultaneous solution process. To avoid this trap, the standard mirror descent method is revised to carefully steer the lower-level solution away from the boundary.

3 Problem formulation

We study incentive design in both atomic games [39] and nonatomic games [47], classified depending on whether the set of agents is endowed with an atomic or a nonatomic measure. In social systems, both types of games can be useful, although the application context varies. Atomic games are typically employed when each agent has a non-trivial influence on the payoffs of other agents. In a nonatomic game, on the contrary, a single agent's influence is negligible and the payoff could only be affected by the collective behavior of agents.

Atomic game. Consider a game played by a finite set of agents $\mathcal{N} = \{1, \ldots, n\}$, where each agent $i \in \mathcal{N}$ selects a strategy $a^i \in \mathcal{A}^i \subseteq \mathbb{R}^{d^i}$ to maximize the reward received, which is determined by a continuously differentiable function $u^i : \mathcal{A} = \prod_{i \in \mathcal{N}} \mathcal{A}^i \to \mathbb{R}$. Formally, a joint strategy $a_* \in \mathcal{A}$ is a Nash equilibrium if

$$u^i(a_*^i, a_*^{-i}) \geq u^i(a^i, a_*^{-i}), \quad \forall\, a^i \in \mathcal{A}^i, \quad \forall i \in \mathcal{N}.$$

Suppose that for all $i \in \mathcal{N}$, the strategy set $\mathcal{A}^i$ is closed and convex, and the reward function u^i is convex in a^i, then $a_* \in \mathcal{A}$ is a Nash equilibrium if and only if there exists $\lambda^i, \ldots, \lambda^n > 0$ such that [48]

$$\sum_{i=1}^{n} \lambda^i \cdot \left\langle \nabla_{a^i} u^i(a_*), a^i - a_*^i \right\rangle \leq 0, \quad \text{for all } a \in \mathcal{A}. \tag{3.1}$$

Example 3.1 (Oligopoly model)**.** In an oligopoly model, there is finite set $\mathcal{N} = \{1, \ldots, n\}$ of firms, each of which supplies the market with a quantity a^i ($a^i \geq 0$) of goods. Under this setting, we have $\mathcal{A} = \mathbb{R}_+^n$. The good is then priced as $p(q) = p_0 - \gamma \cdot q$, where $p_0, \gamma > 0$ and $q = \sum_{j \in \mathcal{N}} a^i$ it the total output. The profit and the marginal profit of the firm i are then given by

$$u^i(a) = a^i \cdot \left(p_0 - \gamma \cdot \sum_{j \in \mathcal{N}} a^j \right) - c^i, \quad \nabla_{a^i} u^i(a) = p_0 - \gamma \cdot \left(a^i + \sum_{j \in \mathcal{N}} a^j \right),$$

respectively, where c^i is the constant marginal cost for firm i.

Nonatomic game. Consider a game played by a continuous set of agents, which can be divided into a finite set of classes $\mathcal{N} = \{1, \ldots, n\}$. We assume that each $i \in \mathcal{N}$ represents a class of infinitesimal and homogeneous agents sharing the finite strategy set $\mathcal{A}^i$ with $|\mathcal{A}^i| = d^i$. The mass distribution for the class i is defined as a vector $q^i \in \Delta(\mathcal{A}^i)$ that gives the proportion of agents using each strategy. Let the cost of an agent in class i to select a strategy $a \in \mathcal{A}^i$ given $q = (q^1, \ldots, q^n)$ be $c^{ia}(q)$. Formally, a joint mass distribution $q \in \Delta(\mathcal{A}) = \prod_{i \in \mathcal{N}} \Delta(\mathcal{A}^i)$ is a Nash equilibrium, also known as a Wardrop equilibrium [52], if for all $i \in \mathcal{N}$, there exists b^i such that

$$\begin{cases} c^{ia}(q_*) = b^i, & \text{if } q_*^{ia} > 0, \\ c^{ia}(q_*) \geq b^i, & \text{if } q_*^{ia} = 0. \end{cases}$$

The following result extends the VI formulation to Nash equilibrium in nonatomic game: denote $c^i(q) = (c^{ia}(q))_{a \in \mathcal{A}^i}$, then q_* is a Nash equilibrium if and only if [10]

$$\sum_{i \in \mathcal{N}} \lambda^i \cdot \left\langle c^i(q_*), q^i - q_*^i \right\rangle \geq 0, \quad \text{for all } q \in \Delta(\mathcal{A}). \tag{3.2}$$

Example 3.2 (Routing game)**.** Consider a set of agents traveling from source nodes to sink nodes in a directed graph with nodes $\mathcal{V}$ and edges $\mathcal{E}$. Denote $\mathcal{N} \subseteq \mathcal{V} \times \mathcal{V}$ as the set of source-sink pairs, $\mathcal{A}^i \subseteq 2^{\mathcal{E}}$ as the set of paths connecting $i \in \mathcal{N}$ and $\mathcal{E}^{ia} \subseteq \mathcal{E}$ as the set of all edges on the path $a \in \mathcal{A}^i$. Suppose that each source-sink pair $i \in \mathcal{N}$ is associated with ρ^i nonatomic agents aiming to choose a route from $\mathcal{A}^i$ to minimize the total cost incurred. Let $q^{ia} \in \Delta(\mathcal{A}^i)$ be the proportion of travelers using the path $a \in \mathcal{A}_w$, $x_e \in \mathbb{R}_+$ be the number of travelers using the edge e and $t^e(x^e) \in \mathbb{R}_+$ be the cost for using edge e. Then we have $x^e = \sum_{i \in \mathcal{N}} \sum_{a \in \mathcal{A}^i} \rho^i \cdot q^{ia} \cdot \delta^{eia}$, where δ^{eik} equals 1 if $e \in \mathcal{E}^{ia}$ and 0 otherwise. The total cost for a traveler selecting a path $a \in \mathcal{A}^i$ will then be $c^{ia}(q) = \sum_{e \in \mathcal{E}} t^e(x^e) \cdot \delta^{eia}$.

Incentive design. Despite the difference, we see that an equilibrium of both atomic and nonatomic games can be formulated as a solution to a corresponding VI problem in the following form

$$\sum_{i \in \mathcal{N}} \lambda^i \cdot \left\langle v^i(x_*), x^i - x_*^i \right\rangle \leq 0, \quad \text{for all } x \in \mathcal{X} = \prod_{i \in \mathcal{N}} \mathcal{X}^i, \tag{3.3}$$

where v^i and $\mathcal{X}^i$ denote different terms in the two types of games. Suppose that there exists an incentive designer aiming to induce a desired equilibrium. To this end, the designer can add incentives $\theta \in \Theta \subseteq \mathbb{R}^d$, which is assumed to enter the reward/cost functions and thus leads to a parameterized $v_\theta^i(x)$. We assume that the designer's objective is determined by a function $f : \Theta \times \mathcal{X} \to \mathbb{R}$. Denote

$S(\theta)$ as the solution set to (3.3). We then obtain the uniform formulation of the incentive design problem for both atomic games and nonatomic games

$$\min_{\theta\in\Theta} f_*(\theta) = f\big(\theta, x_*\big), \quad \text{s.t.} \quad x_* \in S(\theta). \tag{3.4}$$

If the equilibrium problem admits multiple solutions, the agents may converge to different ones and it would be difficult to determine which one can better predict the behaviors of the agents without additional information. In this paper, we *first* consider the case where the game admits a unique equilibrium. Sufficient conditions under which the game admits a unique equilibrium will also be provided later. We would consider the non-unique case later and show that our algorithms can still become applicable by adding an appropriate regularizer in the cost function.

Stochastic environment. In the aforementioned settings, $v_\theta^i(x)$ is a deterministic function. Although most MPEC algorithms in the optimization literature follow this deterministic setting, in this paper, we hope our algorithm can handle more realistic scenarios. Specifically, in the real world, the environment could be stochastic if some environment parameters that fluctuate over days. In a traffic system, for example, both worse weather and special events may affect the road condition, hence the travel time $v_\theta^i(x)$ experienced by the drivers. We expect our algorithm can still work in the face of such stochasticity. To this end, we assume that $v_\theta^i(x)$ represents the expected value of the cost function. On each day, however, the agents can only receive a noised feedback $\widehat{v}_\theta^i$ as an estimation. In the next section, we develop algorithms based on such noisy feedback.

4 Algorithm

We propose to update θ and x simultaneously to improve the computational efficiency. The game dynamics at the lower level is modeled using the mirror descent method. Specifically, at the stage k, given θ_k and x_k, the agent first receives $v_{\theta_k}^i(x_k)$ as the feedback. After receiving the feedback, the agents update their strategies via

$$x_{k+1}^i = \operatorname*{argmax}_{x^i\in\mathcal{X}^i}\big\{\langle v_{\theta_k}^i(x_k), x^i\rangle - 1/\beta_k^i \cdot D_{\psi^i}(x_k^i, x^i)\big\}, \tag{4.1}$$

where $D_{\psi^i}(x_k^i, x^i)$ is the Bregman divergence induced by a strongly convex function ψ^i. The accurate value of $\nabla f_*(\theta_k)$, the gradient of its objective function, equals

$$\nabla_\theta f\big(\theta_k, x_*(\theta_k)\big) + \big[\nabla_\theta x_*(\theta_k)\big]^\top \cdot \nabla_x f\big(\theta_k, x_*(\theta_k)\big),$$

which requires the exact lower-level equilibrium $x_*(\theta_k)$. However, at the stage k, we only have the current strategy x_k. Therefore, we also have to establish an estimator of $\nabla x_*(\theta_k)$ and $\nabla f_*(\theta_k)$ using x_k, the form of which will be specified later.

Remark 4.1. The standard gradient descent method is double-loop because at each θ_k it involves an inner loop for solving the exact value of $x_*(\theta_k)$ and then calculating the exact gradient.

4.1 Unconstrained lower level

We first consider unconstrained games with $\mathcal{X}^i = \mathbb{R}^{d^i}$, for all $i \in \mathcal{N}$. We select $\psi^i(\cdot)$ as smooth function, i.e., there exists a constant $H_\psi \geq 1$ such that for all $i \in \mathcal{N}$ and $x^i, x^{i\prime} \in \mathcal{X}^i$,

$$\big\|\nabla\psi^i(x^i) - \nabla\psi^i(x^{i\prime})\big\|_2 \leq H_\psi \cdot \|x^i - x^{i\prime}\|_2. \tag{4.2}$$

Examples of ψ^i satisfying this assumption include (but are not limited to) $\psi^i(x^i) = (x^i)^\top Q^i x^i/2$, where $Q^i \in \mathbb{R}^{d^i} \times \mathbb{R}^{d^i}$ is a positive definite matrix. It can be directly checked that we can set $H_\psi = \max_{i\in\mathcal{N}} \delta^i$, where δ^i is the largest singular value of Q^i. In this case, the corresponding Bregman divergence becomes $D_{\psi^i}(x^i, x^{i\prime}) = (x^i - x^{i\prime})^\top Q^i (x^i - x^{i\prime})/2$, which is known as the squared Mahalanobis distance. Before laying out the algorithm, we first give the following lemma characterizing $\nabla_\theta x_*(\theta)$.

Lemma 4.2. *When $\mathcal{X}^i = \mathbb{R}^{d^i}$ and $\nabla_x v_\theta(x_*(\theta))$ is non-singular, it holds that*

$$\nabla_\theta x_*(\theta) = -\Big[\nabla_x v_\theta\big(x_*(\theta)\big)\Big]^{-1} \cdot \nabla_\theta v_\theta\big(x_*(\theta)\big).$$

Proof. See Appendix A.2 for detailed proof. □

For any given $\theta \in \Theta$ and $x \in \mathcal{X}$, we define

$$\widetilde{\nabla} f(\theta, x) = \nabla_\theta f(\theta, x) - \big[\nabla_\theta v_\theta(x)\big]^\top \cdot \big[\nabla_x v_\theta(x)\big]^{-1} \cdot \nabla_x f(\theta, x). \tag{4.3}$$

Although we cannot obtain the exact value of $\nabla f_*(\theta_k)$, we may use $\widehat{\nabla} f(\theta_k, x_k)$ as a *surrogate* and update θ_k based on $\widehat{\nabla} f_*(\theta_k, x_k)$ instead. Now we are ready to present the following bilevel incentive design algorithm for unconstrained games (see Algorithm 1).

Algorithm 1 Bilevel incentive design for unconstrained games

Input: $\theta_0 \in \Theta$, $x_0 \in \mathcal{X} = \mathbb{R}^d$, where $d = \sum_{i\in\mathcal{N}} d^i$, sequence of step sizes $(\alpha_k, \{\beta_k^i\}_{i\in\mathcal{N}})$.
For $k = 0, 1, \ldots$ **do:**
Update strategy profile

$$x_{k+1}^i = \operatorname*{argmax}_{x^i \in \mathcal{X}^i} \big\{ \langle \widehat{v}_k^i, x^i \rangle - 1/\beta_k^i \cdot D_{\psi^i}(x_k^i, x^i) \big\}, \tag{4.4}$$

for all $i \in \mathcal{N}$, where $\widehat{v}_k^i$ is an estimator of $v_{\theta_k}^i(x_k)$.
Update incentive parameter

$$\theta_{k+1} = \operatorname*{argmax}_{\theta\in\Theta} \big\{ \langle \widetilde{\nabla} f(\theta_k, x_{k+1}), \theta \rangle - 1/\alpha_k \cdot \|\theta - \theta_k\|_2^2 \big\}.$$

End
Output: Last iteration incentive parameter θ_{k+1} and strategy profile x_{k+1}.

In Algorithm 1, if θ_k and x_k converge to fixed points $\bar{\theta}$ and $\bar{x}$, respectively, then $\bar{x} = x_*(\bar{\theta})$ is expected to be satisfied. Thus, we would also have $\widehat{\nabla} f(\bar{\theta}, \bar{x}) = \nabla f_*(\bar{\theta})$. Thus, the optimality of $\bar{\theta}$ can be then guaranteed. It implies that the algorithm would find the optimal solution if it converges. Instead, the difficult part is how to design appropriate step sizes that ensure convergence. In this paper, we provide such conditions in Section 5.1.

4.2 Simplex-constrained lower level

We then consider the case where for all $i \in \mathcal{N}$, x^i is constrained within the probability simplex

$$\Delta\big([d^i]\big) = \big\{ x^i \in \mathbb{R}^{d^i} \,\big|\, x^i \geq 0, (x^i)^\top \mathbf{1}_{d^i} = 1 \big\}.$$

Here $\mathbf{1}_{d^i} \in \mathbb{R}^{d^i}$ is the vector of all ones. In such a case, we naturally consider $\psi^i(x^i) = -\sum_{j\in[d^i]} [x^i]_j \cdot \log[x^i]_j$, which is the Shannon entropy. Such a choice gives the Bregman divergence $D_{\psi^i}(x^i, x^{i\prime}) = \sum_{j\in[d^i]} [x^i]_j \cdot \log([x^{i\prime}]_j / [x^i]_j)$, which is known as the KL divergence. In this case, we still first need to characterize $\nabla_\theta x_*(\theta)$, which also has an analytic form. Specifically, if we define a function $h_\theta(x) = (h_\theta^i(x))_{i\in\mathcal{N}}$ that satisfies

$$h_\theta^i(x) = \operatorname*{argmax}_{x'^i \in \mathcal{X}^i} \Big\{ \big\langle v_\theta^i(x), x'^i \big\rangle - 1/\beta_k^i \cdot D_{\psi^i}(x^i, x'^i) \Big\},$$

then for any θ, $x_*(\theta)$ would satisfies $x_*(\theta) = h_\theta(x_*(\theta))$ [11]. Implicitly differentiating through this fixed point equation then yields $\nabla x_*(\theta) = \nabla_\theta h_\theta(x_*(\theta)) \cdot (I - \nabla_x h_\theta(x_*(\theta)))^{-1}$. Then, similar to (4.3), we may use

$$\widetilde{\nabla} f(\theta, x) = \nabla_\theta f(\theta, x) - \nabla_\theta h_\theta(x) \cdot \big(I - \nabla_x h_\theta(x)\big)^{-1} \cdot \nabla_x f(\theta, x). \tag{4.5}$$

to approximate the actual gradient $\nabla f_*(\theta)$ and then update θ_k based on $\nabla f_*(\theta_k)$ instead.

Remark 4.3. The mapping $h_\theta(x)$ has an analytic expression, which reads

$$h_\theta^i(x) = x^i \cdot \exp(-\beta_k^i \cdot v_\theta^i(x)) / \|x^i \cdot \exp(-\beta_k^i \cdot v_\theta^i(x))\|_1.$$

Hence, both $\nabla_x h_\theta(x)$ and $\nabla_\theta h_\theta(x)$ can also be calculated analytically.

In addition to a different gradient estimate, we also modify Algorithm 1 to keep the iterations x_k from hitting the boundary at the early stage. The modification involves an additional step that mixes the strategy with a uniform strategy $\mathbf{1}_{d^i}/d^i$, i.e., imposing an additional step

$$\widetilde{x}^i_{k+1} = (1-\nu_{k+1}) \cdot x_{k+1} + \nu_{k+1} \cdot \mathbf{1}_{d^i}/d^i$$

upon finishing the update (4.4), where $\nu_{k+1} \in (0,1)$ is a the mixing parameter, decreasing to 0 when $k \to \infty$. In the following, we give the formal presentation of the modified bilevel incentive design algorithm for simplex-constrained games (see Algorithm 2).

Algorithm 2 Bilevel incentive design for simplex constrained games

Input: $\theta_0 \in \Theta$, $x_0 \in \mathcal{X}$, step sizes $(\alpha_k, \{\beta^i_k\}_{i\in\mathcal{N}})$, $k \geq 0$, and mixing parameters ν_k, $k \geq 0$.
For $k = 0, 1, \ldots$ **do:**
Update strategy profile

$$\begin{aligned} x^i_{k+1} &= \operatorname*{argmax}_{x^i \in \Delta([d^i])} \big\{ \langle \widehat{v}^i_k, x^i \rangle - 1/\beta^i_k \cdot D_{\psi^i}(\widetilde{x}^i_k, x^i) \big\}, \\ \widetilde{x}^i_{k+1} &= (1-\nu_k) \cdot x_{k+1} + \nu_k \cdot \mathbf{1}_{d^i}/d^i, \end{aligned} \tag{4.6}$$

for all $i \in \mathcal{N}$, where $\widehat{v}^i_k$ is an estimator of $v^i_{\theta_k}(\widetilde{x}_k)$.
Update incentive parameter

$$\theta_{k+1} \leftarrow \operatorname*{argmax}_{\theta\in\Theta} \big\{ \langle \widetilde{\nabla} f(\theta_k, \widetilde{x}_{k+1}), \theta \rangle - 1/\alpha_k \cdot \|\theta - \theta_k\|_2^2 \big\}.$$

End

Similar to Algorithm 1, at the core of the convergence of Algorithm 2 is still the step size. This case is even more complicated, as we need to design α_k, β_k, and ν_k at the same time. In this paper, a provably convergent scheme is provided in Section 5.2.

Why do we study simplex constraints? Many classic game-theoretic models are simplex-constrained. In fact, as long as the agent faces a finite number of choices (e.g., self-driving or calling a taxi, online shopping or on-site shopping) and adopts a mixed strategy, its decision space would be a probability simplex [39]. In addition, some other types of decisions may also be constrained by a simplex. For example, financial investment concerns how to split the money on different assets. In such a scenario, the budget constraint can also be represented by a probability simplex.

Can the algorithm work on other constraints? The algorithm can be adapted to other types of constraints by using another $h_\theta(x)$ to model the game dynamics. In particular, the projected gradient descent dynamics has very broad applicability. In this case, the algorithm for calculating $\nabla_\theta h_\theta(x)$ and $\nabla_x h_\theta(x)$ is given by, for example, Amos and Kolter [2]. The additional step (4.6) then becomes unnecessary as it is dedicated to simplex constraints.

5 Convergence analysis

In this section, we study the convergence of the proposed algorithms. For simplicity, define $\overline{D}_\psi(x, x') = \sum_{i\in\mathcal{N}} D_{\psi^i}(x^i, x^{i\prime})$. We make the following assumptions.

Assumption 5.1. The lower-level problem in (3.4) satisfies:

- The strategy set $\mathcal{X}^i$ of player i is a nonempty, compact, and convex subset of $\mathbb{R}^{d^i}$.
- For each $i \in \mathcal{N}$, the gradient $v^i_\theta(\cdot)$ is H_u-Lipschitz continuous with respect to $\overline{D}_\psi$, i.e., for all $i \in \mathcal{N}$ and $x, x' \in \mathcal{X}$, $\|v^i_\theta(x) - v^i_\theta(x')\|_*^2 \leq H_u^2 \cdot \overline{D}_\psi(x, x')$.
- There exist constants $\rho_\theta, \rho_x > 0$ such that for all $x \in \mathcal{X}$ and $\theta \in \Theta$, $\|\nabla_\theta v_\theta(x)\|_2 < \rho_\theta$, and $\|[\nabla_x v_\theta(x)]^{-1}\|_2 \leq 1/\rho_x$.
- For all $\theta \in \Theta$, the equilibrium $x_*(\theta)$ of the game is strongly stable with respect to $\overline{D}_\psi$, i.e., for all $x \in \mathcal{X}$, $\sum_{i\in\mathcal{N}} \lambda^i \cdot \langle v^i_\theta(x), x^i_*(\theta) - x^i \rangle \geq \overline{D}_\psi(x_*(\theta), x)$.

Assumption 5.2. The upper-level problem in (3.4) satisfies:

- The set Θ is compact and convex. The function $f_*(\theta)$ is μ-strongly convex and $\nabla f_*(\theta)$ has 2-norm uniformly bounded by M.
- The extended gradient $\widetilde{\nabla} f(\theta, x)$ is $\widetilde{H}$-Lipschitz continuous with respect to $\overline{D}_\psi$, i.e., for all $x, x' \in \mathcal{X}$, $\|\widetilde{\nabla} f(\theta, x) - \widetilde{\nabla} f(\theta, x')\|_2^2 \leq \widetilde{H}^2 \cdot \overline{D}_\psi(x, x')$.

Assumption 5.3. Define the filtration by $\mathcal{F}_0^\theta = \{\theta_0\}$, $\mathcal{F}_0^x = \emptyset$, $\mathcal{F}_k^\theta = \mathcal{F}_{k-1}^\theta \cup \{x_{k-1}, \theta_k\}$, and $\mathcal{F}_k^x = \mathcal{F}_{k-1}^x \cup \{\theta_k, x_k\}$. It satisfies:

- The feedback $\widehat{v}_k$ is an unbiased estimate, i.e., for all $i \in \mathcal{N}$, we have $\mathbb{E}[\widehat{v}_k^i \mid \mathcal{F}_k^x] = v_{\theta_k}^i(x_k)$.
- The feedback $\widehat{v}_k$ has bounded mean squared estimation errors, i.e., there exist $\delta_f, \delta_u > 0$ such that $\mathbb{E}\Big[\big\|\widehat{v}_k^i - v_{\theta_k}^i(x_k)\big\|_*^2 \,\Big|\, \mathcal{F}_k^x\Big] \leq \delta_u^2, \quad \forall i \in \mathcal{N}$.

Below we discuss when the proposed assumptions hold, and if they are violated, how would our algorithm works.

Remark 5.4. Assumption 5.1 includes the condition that $x_*(\theta)$ is strongly stable. In this case, it is the unique Nash equilibrium of the game [34]. It is also a common assumption in the analysis of the mirror descent dynamics itself [12]. We provide sufficient conditions for checking strong stability in Appendix A.1. We refer the readers to Section 6 for an explanation of how to extend our algorithm when this assumption is violated.

Remark 5.5. Assumption 5.2 includes the convexity of the upper-level problem, which is usually a necessary condition to ensure global convergence. Yet, without convexity, our algorithm can still converge to a local minimum.

Remark 5.6. Assumption 5.3 becomes unnecessary if we simply assume that the environment is deterministic. In this case, the accurate value of $v_\theta^i(x)$ is available. Yet, if the noises are added to the feedback, assuming that the noisy feedback is unbiased and bounded is still reasonable.

5.1 Unconstrained game

In this part, we establish the convergence guarantee of Algorithm 1 for unconstrained games. We define the optimality gap ϵ_k^θ and the equilibrium gap ϵ_{k+1}^x as

$$\epsilon_k^\theta := \mathbb{E}\big[\|\theta_k - \theta_*\|_2^2\big], \quad \epsilon_{k+1}^x := \mathbb{E}\Big[\overline{D}_\psi\big(x_*(\theta_k), x_{k+1}\big)\Big].$$

We track such two gaps as the convergence criteria in the subsequent results.

Theorem 5.7. *For Algorithm 1, set the step sizes $\alpha_k = \alpha/(k+1)$, $\beta_k = \beta/(k+1)^{2/3}$, and $\beta_k^i = \lambda^i \cdot \beta_k$ with constants*

$$\beta \leq 1/N \cdot H_u^2 \|\lambda\|_2^2, \quad \alpha/\beta^{3/2} \leq 1/12 \cdot H_\psi \widetilde{H} H_*,$$

where $H_* = \rho_\theta/\rho_x$. Suppose that Assumptions 5.1-5.3 hold, then we have

$$\epsilon_k^\theta = O(k^{-2/3}), \quad \epsilon_k^x = O(k^{-2/3}).$$

Proof. See Appendix B for detailed proof and a detailed expression of convergence rates. □

5.2 Simplex-constrained game

In this part, we establish the convergence guarantee of Algorithm 2 for simplex constrained games. We still define optimality gap ϵ_k^θ as $\epsilon_k^\theta = \mathbb{E}\big[\|\theta_k - \theta_*\|_2^2\big]$. Yet, corresponding to (4.6), we track $\widetilde{\epsilon}_{k+1}^x$ as a measure of convergence for the strategies of the agents, which is defined as

$$\widetilde{\epsilon}_{k+1}^x = \mathbb{E}\Big[\overline{D}_\psi\big(\widetilde{x}_*(\theta_k), \widetilde{x}_{k+1}\big)\Big],$$

where $\widetilde{x}_*(\theta_k) = (1 - \nu_k) \cdot x_*(\theta_k) + \nu_k \cdot \mathbf{1}/d^i$. We are now ready to give the convergence guarantee of Algorithm 2.

Theorem 5.8. *For Algorithm 2, set the step sizes* $\alpha_k = \alpha/(k+1)^{1/2}$, $\beta_k = \beta/(k+1)^{2/7}$, $\beta_k^i = \lambda^i \cdot \beta_k$, *and* $\nu_k = \nu/(k+1)^{4/7}$ *with constants* α *and* β *satisfying*

$$\beta \leq 1/6 \cdot N H_u^2 \|\lambda\|_2^2, \quad \alpha/\beta^{3/2} \leq 1/7 \cdot \widetilde{H}\widetilde{H}_*,$$

where $\widetilde{H}_* = (1+d)\rho_\theta/\rho_x$. *Suppose that Assumptions 5.1-5.3 hold. If there exists some constant* $V_* > 0$ *such that* $\|v_\theta(x_*(\theta))\|_\infty \leq V_*$ *for any* $\theta \in \Theta$, *we then have*

$$\epsilon_k^\theta = O(k^{-2/7}), \quad \widetilde{\epsilon}_k^x = O(k^{-2/7}).$$

Proof. See Appendix C for detailed proof and a detailed form of the convergence rates. □

Implication. The two theorems established in our paper provide conditions under which the simultaneous design-and-play scheme provably converges. The existence of such conditions verifies that the single-loop bilevel algorithms developed recently in machine learning can be further extended to handle equilibrium constraints with a theoretical guarantee.

6 Extensions to games with multiple equilibria

It has long been recognized that when a game has multiple equilibria, further assumptions have to be imposed to select a representative one [20]. Below we show that these assumptions often simultaneously make our algorithms applicable.

Case I: If the function $v_\theta(x) = (v_\theta^i(x))_i$ is strongly monotone in the neighbourhood of each equilibrium, then all equilibria are strongly stable in a neighbourhood and hence isolated [38]. In this case, our algorithms can be directly applied as $\nabla_x v_\theta(x)$ is non-singular in these neighborhoods. It is commonly believed that the most likely equilibrium is the one reached by the game dynamics [53]; our algorithm naturally converges to this one.

Case II: If the function $v_\theta(x) = (v_\theta^i(x))_i$ is monotone but *not* strongly monotone, then the equilibrium set is a convex and closed region [34]. This case is challenging, as the matrix $\nabla_x v_\theta(x)$ needed to be inverted would become singular. Nevertheless, we can simply assume the agents are *bounded rationale* [43, 1, 33]. The bounded rationality would result in a quantum equilibrium for predicting the agents' response. We refer the readers to Appendix E for a detailed explanation (with numerical examples for illustration). Here we briefly sum up the key takeaways: (1) it is equivalent to add a regularizer $\eta^i \cdot (\log(x^i + \epsilon) + 1)$ to $v_\theta^i(x)$ for some $\eta^i > 0$ and $\epsilon \geq 0$; (2) as long as $\theta > 0$, the strong stability condition in Assumption 5.1 would then be satisfied, hence a unique equilibrium would exist; (3) as long as $\epsilon > 0$, the Lipschitz continuous condition in Assumption 5.1 will also not be violated. In a nutshell, the bounded rationality assumption can simultaneously make our model more realistic and satisfy the assumptions in Section 5.

7 Numerical experiments

Pollution control via emission tax. Consider the oligopoly model introduced in Example 3.1. We assume that producing a_i units of output, firm i would generate $e_i = d^i a_i$ units of emissions. We consider the following social welfare function [45] $W(a) = \int_0^{\sum_{i=1}^n a^i} p(q)\,\mathrm{d}q - \sum_{i=1}^n c^i \cdot q^i - \tau \cdot \sum_{i=1}^n d^i a^i$, where the first term is the consumers' surplus, the second term is the total production cost, and the third term is the social damage caused by pollution. To maximize social welfare, an authority can impose emission taxes on the productions. Specifically, whenever producing a_i units of output, firm i could be charged $\pi^i \cdot a^i$, where π^i is specialized for the firm i.

Second-best congestion pricing. Consider the routing game model introduced in Example 3.2. To minimize the total travel delay, an authority on behalf of the public sector could impose appropriate tolls on selected roads [51]. This problem of determining tolls is commonly known as the congestion pricing problem. The second-best scheme assumes that only a subset of links can be charged [50].

We test our algorithm on these two incentive design problems and compare them with some existing methods in both optimization and machine learning literature. We refer the readers to Appendix F for the results due to space limitations.

References

[1] Ahmed, Z., Le Roux, N., Norouzi, M., and Schuurmans, D. (2019). Understanding the impact of entropy on policy optimization. In *International conference on machine learning*, pages 151–160. PMLR.

[2] Amos, B. and Kolter, J. Z. (2017). Optnet: Differentiable optimization as a layer in neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 136–145.

[3] Anandalingam, G. and Friesz, T. L. (1992). Hierarchical optimization: An introduction. *Annals of Operations Research*, 34(1):1–11.

[4] Bai, S., Kolter, J. Z., and Koltun, V. (2019). Deep equilibrium models. In *Advances in Neural Information Processing Systems*, pages 688–699.

[5] Bracken, J. and McGill, J. T. (1973). Mathematical programs with optimization problems in the constraints. *Operations Research*, 21(1):37–44.

[6] Buchanan, J. M. and Stubblebine, W. C. (1962). Externality. In *Classic papers in natural resource economics*, pages 138–154. Springer.

[7] Caron, S. and Kesidis, G. (2010). Incentive-based energy consumption scheduling algorithms for the smart grid. In *2010 First IEEE International Conference on Smart Grid Communications*, pages 391–396. IEEE.

[8] Chen, T., Sun, Y., and Yin, W. (2021). A single-timescale stochastic bilevel optimization method. *arXiv preprint arXiv:2102.04671*.

[9] Colson, B., Marcotte, P., and Savard, G. (2007). An overview of bilevel optimization. *Annals of operations research*, 153(1):235–256.

[10] Dafermos, S. (1980). Traffic equilibrium and variational inequalities. *Transportation Science*, 14(1):42–54.

[11] Dafermos, S. (1983). An iterative scheme for variational inequalities. *Mathematical Programming*, 26(1):40–47.

[12] D'Orazio, R., Loizou, N., Laradji, I., and Mitliagkas, I. (2021). Stochastic mirror descent: Convergence analysis and adaptive variants via the mirror stochastic polyak stepsize. *arXiv preprint arXiv:2110.15412*.

[13] Figlio, D. N. and Kenny, L. W. (2007). Individual teacher incentives and student performance. *Journal of public economics*, 91(5-6):901–914.

[14] Finn, C., Abbeel, P., and Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR.

[15] Franceschi, L., Donini, M., Frasconi, P., and Pontil, M. (2017). Forward and reverse gradient-based hyperparameter optimization. In *International Conference on Machine Learning*, pages 1165–1173. PMLR.

[16] Friedman, M. and Friedman, R. (1990). *Free to choose: A personal statement*. Houghton Mifflin Harcourt.

[17] Friesz, T. L., Tobin, R. L., Cho, H.-J., and Mehta, N. J. (1990). Sensitivity analysis based heuristic algorithms for mathematical programs with variational inequality constraints. *Mathematical Programming*, 48(1-3):265–284.

[18] Greenwald, B. C. and Stiglitz, J. E. (1986). Externalities in economies with imperfect information and incomplete markets. *The quarterly journal of economics*, 101(2):229–264.

[19] Harker, P. T. and Pang, J.-S. (1988). Existence of optimal solutions to mathematical programs with equilibrium constraints. *Operations research letters*, 7(2):61–64.

[20] Harsanyi, J. C., Selten, R., et al. (1988). A general theory of equilibrium selection in games. *MIT Press Books*, 1.

[21] Hong, M., Wai, H.-T., Wang, Z., and Yang, Z. (2020). A two-timescale framework for bilevel optimization: Complexity analysis and application to actor-critic. *arXiv preprint arXiv:2007.05170*.

[22] Jiang, H., Chen, Z., Shi, Y., Dai, B., and Zhao, T. (2021). Learning to defend by learning to attack. In *International Conference on Artificial Intelligence and Statistics*, pages 577–585. PMLR.

[23] Krichene, W., Krichene, S., and Bayen, A. (2015). Convergence of mirror descent dynamics in the routing game. In *2015 European Control Conference (ECC)*, pages 569–574. IEEE.

[24] Lawphongpanich, S. and Hearn, D. W. (2004). An mpec approach to second-best toll pricing. *Mathematical programming*, 101(1):33–55.

[25] Li, J., Yu, J., Nie, Y. M., and Wang, Z. (2020). End-to-end learning and intervention in games. *Advances in Neural Information Processing Systems*, 33.

[26] Liu, H., Simonyan, K., and Yang, Y. (2018). Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*.

[27] Liu, R., Gao, J., Zhang, J., Meng, D., and Lin, Z. (2021). Investigating bi-level optimization for learning and vision from a unified perspective: A survey and beyond. *arXiv preprint arXiv:2101.11517*.

[28] Luketina, J., Berglund, M., Greff, K., and Raiko, T. (2016). Scalable gradient-based tuning of continuous regularization hyperparameters. In *International conference on machine learning*, pages 2952–2960. PMLR.

[29] Luo, Z.-Q., Pang, J.-S., and Ralph, D. (1996). *Mathematical programs with equilibrium constraints*. Cambridge University Press.

[30] Maclaurin, D., Duvenaud, D., and Adams, R. (2015). Gradient-based hyperparameter optimization through reversible learning. In *International conference on machine learning*, pages 2113–2122. PMLR.

[31] Martin, R., De Preux, L. B., and Wagner, U. J. (2014). The impact of a carbon tax on manufacturing: Evidence from microdata. *Journal of Public Economics*, 117:1–14.

[32] Maskin, E. S. (1994). The invisible hand and externalities. *The American Economic Review*, 84(2):333–337.

[33] Melo, E. (2021). On the uniqueness of quantal response equilibria and its application to network games. *Economic Theory*, pages 1–45.

[34] Mertikopoulos, P. and Zhou, Z. (2019). Learning in games with continuous action sets and unknown payoff functions. *Mathematical Programming*, 173(1-2):465–507.

[35] Metcalf, G. E. and Weisbach, D. (2009). The design of a carbon tax. *Harv. Envtl. L. Rev.*, 33:499.

[36] Metz, L., Poole, B., Pfau, D., and Sohl-Dickstein, J. (2016). Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*.

[37] Mguni, D., Jennings, J., Sison, E., Valcarcel Macua, S., Ceppi, S., and Munoz de Cote, E. (2019). Coordinating the crowd: Inducing desirable equilibria in non-cooperative systems. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 386–394.

[38] Nagurney, A. (2013). *Network economics: A variational inequality approach*, volume 10. Springer Science & Business Media.

[39] Nash, J. (1951). Non-cooperative games. *Annals of mathematics*, pages 286–295.

[40] Nemirovskij, A. S. and Yudin, D. B. (1983). *Problem Complexity and Method Efficiency in Optimization*. A Wiley-Interscience publication. Wiley.

[41] Parise, F. and Ozdaglar, A. (2017). Sensitivity analysis for network aggregative games. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 3200–3205. IEEE.

[42] Pigou, A. C. (1920). *The economics of welfare*. Palgrave Macmillan.

[43] Prashker, J. N. and Bekhor, S. (2004). Route choice models used in the stochastic user equilibrium problem: a review. *Transport reviews*, 24(4):437–463.

[44] Rajeswaran, A., Finn, C., Kakade, S., and Levine, S. (2019). Meta-learning with implicit gradients. *arXiv preprint arXiv:1909.04630*.

[45] Requate, T. (1993). Pollution control in a cournot duopoly via taxes or permits. *Journal of Economics*, 58(3):255–291.

[46] Roth, A., Ullman, J., and Wu, Z. S. (2016). Watch and learn: Optimizing from revealed preferences feedback. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 949–962.

[47] Schmeidler, D. (1973). Equilibrium points of nonatomic games. *Journal of statistical Physics*, 7(4):295–300.

[48] Scutari, G., Palomar, D. P., Facchinei, F., and Pang, J.-S. (2010). Convex optimization, game theory, and variational inequality theory. *IEEE Signal Processing Magazine*, 27(3):35–49.

[49] Smith, A. (1791). *An Inquiry into the Nature and Causes of the Wealth of Nations*, volume 1. Librito Mondi.

[50] Verhoef, E. T. (2002). Second-best congestion pricing in general networks. heuristic algorithms for finding second-best optimal toll levels and toll points. *Transportation Research Part B: Methodological*, 36(8):707–729.

[51] Vickrey, W. S. (1969). Congestion theory and transport investment. *The American Economic Review*, 59(2):251–260.

[52] Wardrop, J. G. (1952). Some theoretical aspects of road traffic research. In *Proceedings of the Institution of Civil Engineers*, volume 1, pages 325–362.

[53] Weibull, J. W. (1997). *Evolutionary game theory*. MIT press.

[54] Wu, Y., Ren, M., Liao, R., and Grosse, R. (2018). Understanding short-horizon bias in stochastic meta-optimization. *arXiv preprint arXiv:1803.02021*.

[55] Zheng, S., Trott, A., Srinivasa, S., Naik, N., Gruesbeck, M., Parkes, D. C., and Socher, R. (2020). The ai economist: Improving equality and productivity with ai-driven tax policies. *arXiv preprint arXiv:2004.13332*.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] We clarified that our algorithm is dedicated to incentive design problems in games.
   (b) Did you describe the limitations of your work? [Yes] The convergence analysis is restricted to games with a unique equilibrium. However, we also discussed how to extend our work into more general settings.
   (c) Did you discuss any potential negative societal impacts of your work? [No] Our work studied algorithms for solving the incentive design problem. It may help to improve social welfare.

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] Assumptions 5.1-5.3
   (b) Did you include complete proofs of all theoretical results? [Yes] In Appendices B-C
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] The codes can be provided if necessary.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix F.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]