Inverse Matrix Games with Unique Nash Equilibrium

Yue Yu, Jonathan Salfity, David Fridovich-Keil, and Ufuk Topcu

Abstract—In an inverse game problem, one needs to infer the cost function of the players in a game such that a desired joint strategy is a Nash equilibrium. We study the inverse game problem for a class of multiplayer games, where the players’ cost functions are characterized by a matrix. We guarantee that the desired joint strategy is the unique Nash equilibrium of the game with the inferred cost matrix. We develop efficient optimization algorithms for inferring the cost matrix based on semidefinite programs and bilevel optimization. We demonstrate the application of these methods using examples where we infer the cost matrices that encourage noncooperative players to achieve collision avoidance in path-planning and fairness in resource allocation.

I. INTRODUCTION

In a multiplayer game, each player tries to find the strategies with the minimum cost, where the cost of each strategy depends on the other players’ strategies. The Nash equilibrium is a set of strategies where no player can benefit from unilaterally changing strategies. The Nash equilibrium generalizes minimax equilibrium in two-player zero-sum games [1] to multiplayer general-sum games [2], [3].

Given a desired joint strategy of the players in a game, the inverse game problem requires inferring the cost function such that the desired joint strategy is indeed a Nash equilibrium. There have been many studies on inverse games in different contexts, including specific games, such as matching [4], network formation [5], and auction [6]; and generic classes of games, such as succinct games [7] and noncooperative dynamic games [8].

One common drawback in the existing inverse game results is that the inferred cost can make multiple joint strategies become Nash equilibria simultaneously [9]. As a result, the knowledge of Nash equilibrium can be useless in each player’s decision-making, and undesired strategies can become Nash equilibria unintentionally. For example, let us consider the two-player game illustrated in Fig. 1, where each player is a car, each strategy is a turning direction, and the Nash equilibria are the two joint strategies that avoid collision. Since either strategy of each player is part of one Nash equilibrium, knowing the Nash equilibrium itself does not help either player choose a collision-avoiding strategy.

Furthermore, let us consider the extreme case where, based on the two Nash equilibria, we infer both players’ cost to be uniformly zero, regardless of their strategies. Then such an inference will make any joint strategy become a Nash equilibrium, even the undesired ones that cause collision.

We study a class of inverse matrix games with guaranteed unique Nash equilibrium. In these games, each player’s strategy is a probability distribution over a finite number of discrete actions, and the cost of a strategy is characterized by a cost matrix. Our contributions are as follows.

First, by adding entropy regularization in each player’s cost function, we provide sufficient conditions for the corresponding matrix games to have a unique Nash equilibrium. Furthermore, we show that one can efficiently compute this unique Nash equilibrium by solving a nonlinear least-squares problem. Second, we develop two numerical methods—one based on semidefinite programs, the other based on the projected gradient method—for inverse matrix games with unique Nash equilibrium. Finally, we demonstrate the application of these methods in inferring the cost matrices—which can be interpreted as tolls or subsidies—that encourage collision avoidance in path-planning and fairness in resource allocation.

Our results address a previously overlooked challenge in mechanism design: how to motivate desired behavior while not encouraging other undesired behavior unintentionally. This challenge is commonplace in competitive games: the rule of offensive foul in basketball games is intended for reducing illegal body contact, but also unintentionally encourage flopping; government subsidies intended for lowering fossil fuel prices can unintentionally discourage the investments in renewable energy.

Notation: We let \( \mathbb{R} \), \( \mathbb{R}_+ \), \( \mathbb{R}_{++} \), and \( \mathbb{N} \) denote the set of real, nonnegative real, positive real, and positive integer numbers, respectively. Given \( m, n \in \mathbb{N} \), we let \( \mathbb{R}^n \) and \( \mathbb{R}^{m \times n} \) denote the set of \( n \)-dimensional real vectors and \( m \times n \) real matrices; we let \( \mathbf{1}_n \) and \( I_n \) denote the \( n \)-dimensional vector of all 1’s and the \( n \times n \) identity matrix, respectively. Given
positive integer \( n \in \mathbb{N} \), we let \([n] := \{1, 2, \ldots, n\}\) denote the set of positive integers less or equal to \( n \). Given \( x \in \mathbb{R}^n \) and \( k \in [n] \), we let \([x]_k\) denote the \( k \)-th element of vector \( x \), and \( \|x\| \) denote the \( \ell_2 \)-norm of \( x \). Given a square real matrix \( A \in \mathbb{R}^{n \times n} \), we let \( A^T \), \( A^{-1} \), and \( A^{-T} \) denote the transpose, the inverse, and the transpose of the inverse of matrix \( A \), respectively; we say \( A \succeq 0 \) and \( A \succ 0 \) if \( A \) is symmetric positive semidefinite and symmetric positive definite, respectively; we let \( \|A\|_F \) denote the Frobenius norm of matrix \( A \). We let \( \text{blkdiag}(A_1, \ldots, A_k) \) denote the block diagonal matrix whose diagonal blocks are \( A_1, \ldots, A_k \in \mathbb{R}^{m \times m} \).

Given continuously differentiable functions \( f_i : \mathbb{R}^n \to \mathbb{R} \) and \( G : \mathbb{R}^n \to \mathbb{R}^m \), we let \( \nabla f_i(x) \in \mathbb{R}^n \) denote the gradient of \( f_i \) evaluated at \( x \in \mathbb{R}^n \); the \( k \)-th element of \( \nabla f_i(x) \) is \( \partial f_i(x)_k \). Furthermore, we let \( \partial_i G(x) \in \mathbb{R}^{m \times n} \) denote the Jacobian of function \( G \) evaluated at \( x \in \mathbb{R}^n \); the \( ij \)-th element of matrix \( \partial_i G(x) \) is \( \partial_i G_{ij}(x) \).

### II. ENTROPY-REGULARIZED MATRIX GAMES

We introduce our theoretical model, a multiplayer matrix game with entropy regularization. We provide sufficient conditions for this game to have a unique Nash equilibrium. Furthermore, we show that one can compute this unique Nash equilibrium by solving a nonlinear least-squares problem.

#### A. Multiplayer matrix games

We consider a game with \( n \in \mathbb{N} \) players. Each player \( i \in [n] \) has \( m_i \in \mathbb{N} \) actions. We let \( m := \sum_{i=1}^n m_i \) denote the total number of actions of all players. Player \( i \)'s strategy is an \( m_i \)-dimensional probability distribution over all possible actions, denoted by \( x_i \in \Delta_i \), where

\[
\Delta_i := \{ y \in \mathbb{R}^{m_i} | y^\top 1_{m_i} = 1, y \geq 0 \}. \tag{1}
\]

Each player’s optimal strategy \( x_i \) is one that minimizes the expected cost, which is determined by the strategies of all players. In particular, player \( i \)'s strategy \( x_i \) satisfies the following condition:

\[
x_i \in \underset{y \in \Delta_i}{\arg \min} \quad \left( b_i + \frac{1}{2} C_{ii} y + \sum_{j \neq i} C_{ij} x_j \right)^\top y \tag{2}
\]

where \( b_i \in \mathbb{R}^{m_i} \) and \( C_{ij} \in \mathbb{R}^{m_i \times m_j} \) for all \( i, j \in [n] \) are cost parameters. Notice that if \( C_{ij} = 0_{m_i \times m_j} \), then the condition in (2) simply says that distribution \( x_i \) only selects the actions that correspond to the smallest entries in \( b_i \).

We will also use the following notation:

\[
b := \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}, \quad C := \begin{bmatrix} C_{11} & C_{12} & \ldots & C_{1n} \\ C_{21} & C_{22} & \ldots & C_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ C_{n1} & C_{n2} & \ldots & C_{nn} \end{bmatrix} \tag{3}
\]

We denote the joint strategy of all players as

\[
x := [x_1^\top \ x_2^\top \ \ldots \ \ x_n^\top ]^\top \tag{4}
\]

We define the Nash equilibrium in the above multiplayer matrix game as follows.

**Definition 1.** A joint strategy \( x := [x_1^\top \ x_2^\top \ \ldots \ \ x_n^\top ]^\top \) is a Nash equilibrium if (2) holds for all \( i \in [n] \).

The particular form of Nash equilibrium in Definition 1 was first introduced in [10], along with its existence and uniqueness conditions. However, computing such a Nash equilibrium is computationally challenging, even for two-player games; see [11] and references therein for details.

#### B. Entropy-regularized multiplayer matrix games

In order to reduce the complexity of computing the Nash equilibrium in Definition 1 we now introduce an entropy-regularized matrix game. In such a game, player \( i \)'s strategy is characterized by, instead of (2), the following:

\[
x_i \in \underset{y \in \Delta_i}{\arg \min} \quad \left( b_i + \frac{1}{2} C_{ii} y + \sum_{j \neq i} C_{ij} x_j \right)^\top y + \lambda y^\top \ln(y) \tag{5}
\]

for all \( i \in [n] \), where \( \lambda \in \mathbb{R}_{++} \) is a regularization weight, and \( \ln(y) \in \mathbb{R}^i \) is the elementwise logarithm of vector \( y \). Intuitively, the entropy term in (5) encourages a diversified distribution. However, if the value of \( \lambda \) in (5) is sufficiently small—in practice, one order of magnitude smaller than the elements in \( b_i \)—the effect of this entropy term becomes numerically negligible; see Fig. 2 for an illustration for the case where \( n = 2 \), \( m_1 = m_2 = 2 \), \( b_1 = [1 \ 0]^\top \), \( b_2 = [2 \ 0]^\top \), \( C_{ij} = 0_{2 \times 2} \) for all \( i, j = 1, 2 \).

![Fig. 2: The effects of entropy regularization in (5).](image)

The following definition introduces a modified version of the Nash equilibrium in Definition 1.

**Definition 2.** A joint strategy \( x := [x_1^\top \ x_2^\top \ \ldots \ \ x_n^\top ]^\top \) is a *entropy-regularized* Nash equilibrium if (5) holds for all \( i \in [n] \).

The benefits of the extra entropy regularization in (5) is as follows. The condition in (2) is a variational inequality, which is generally difficult to solve. However, if \( \lambda \) is strictly positive, the condition in (5) is equivalent to a set of explicit nonlinear equations, as shown by the following lemma.

**Lemma 1.** If \( \lambda > 0 \), then (5) holds if and only if

\[
x_i = f_i \left(-\frac{1}{\lambda \left(b_i + \sum_{j=1}^n C_{ij} x_j\right)}\right) \tag{6}
\]

where \( f_i(z) := \frac{1}{\Gamma(m_i)} \exp(z) \) for all \( z \in \mathbb{R}^{m_i} \), and \( \exp(z) \in \mathbb{R}^{m_i} \) is the elementwise exponential of vector \( z \).

**Proof.** The proof is similar to the one in [12, Thm. 4].

**Remark 1.** Lemma 1 shows that the Nash equilibrium is characterized by the softmax function, commonly used in neural networks [12, Chp. 2]. A similar form of equilibrium is also known as the logit quantal response equilibrium in the literature [13], [14].
C. Computing the entropy-regularized Nash equilibrium via nonlinear least-squares

Thanks to Lemma 1, we can compute the Nash equilibrium in Definition 2 by solving the following nonlinear least-squares problem:

\[
\min_x \sum_{i=1}^n \left\| x_i - f_i \left( -\frac{1}{\lambda}(b_i + \sum_{j=1}^n C_{ij}x_j) \right) \right\|^2
\]

(7)

where function \( f_i \) is given by (6). If the optimal value of the objective function in optimization (7) is zero, then the corresponding solution \( x \) is indeed a Nash equilibrium that satisfies (6) for all \( i \in [n] \).

However, the question remains whether the solution of optimization (7) is unique, and whether this solution exactly satisfies equation (6) for all \( i \in [n] \). We will answer these questions next.

The following proposition provides sufficient conditions under which the Nash equilibrium in Definition 2 exists and is unique.

**Proposition 1.** If \( C + C^T \succeq 0 \) and \( \lambda > 0 \), then there exists a unique \( x = [x_1^T \ x_2^T \ \ldots \ x_n^T]^T \in \mathbb{R}^{m+n} \) such that (6) holds for all \( i \in [n] \).

**Proof.** Since (5) implies that \( x_i \) is elementwise strictly positive (due to the logarithm function), the proof is a direct application of [10, Thm. 1] and [10, Thm. 6]; see [10, p. 529] on the related discussions on bilinear games. \( \square \)

Under the assumptions in Proposition 1, we know that the solution of optimization (7) is indeed unique, and this solution actually satisfies (6) for all \( i \in [n] \).

To solve optimization (7), one can use any off-the-shelf numerical methods for nonlinear least-squares problems, such as the Gauss-Newton method and the Levenberg–Marquardt method. We refer the interested readers to [15, Ch. 10] for further details on these methods.

III. NUMERICAL METHODS FOR INVERSE MATRIX GAMES

Given a cost matrix \( C \), the previous section shows how to compute the Nash equilibrium in Definition 2 by solving a nonlinear least-squares problem. We now consider the reverse of this process: given a desired joint strategy \( x \), how to infer the cost matrix \( C \) that makes \( x \) the unique Nash equilibrium in Definition 2? Here we only consider the inferring of the matrix \( C \) rather than the vector \( b \), since the former captures the interaction among different players and may therefore be more difficult to infer. However, we note that one can seamlessly generalize the results in this section to the inference of vector \( b \).

In the following, we will introduce two different approaches for the aforementioned inverse matrix game: one based on semidefinite programs, the other based on the projected gradient method for bilevel optimization.

A. Semidefinite program approach

We first consider the case where the desired Nash equilibrium is a pure joint strategy, where each player \( i \) has a preferred action \( i^* \in [m_i] \). In particular, suppose there exists \( x^* \in \mathbb{R}^m \) and \( i^* \in [m_i] \) for all \( i \in [n] \) such that

\[
[x^*]_k = \begin{cases} 
1, & k = i^* \\
0, & \text{otherwise.}
\end{cases}
\]

(8)

In this case, perhaps the most direct way to ensure \( x^* \) is a Nash equilibrium is to simply make sure that the cost of action \( i^* \) is sufficiently lower than any alternatives for player \( i \). By combining these constraints together with the results with the definition in Proposition 1 we obtain the following semidefinite program:

\[
\begin{aligned}
\min_{\xi} & \quad \frac{1}{2} \| C \|_F^2 \\
\text{subject to} & \quad C + C^T \succeq 0, \\
& \quad [b_i]_i + \sum_{j=1}^n C_{ij}x_j + \varepsilon \leq [b_i]_k + \sum_{j=1}^n C_{kj}x^*_j, \\
& \quad \forall k \in [m_i] \setminus \{i^*\}, \ i \in [n].
\end{aligned}
\]

(9)

where the objective function penalizes large values of the elements in matrix \( C \), and \( \varepsilon \in \mathbb{R}_+ \) is a tuning parameter that separates the cost of the best action from the cost of the second best action. Intuitively, as the the value of \( \varepsilon \) increases, the entropy-regularized Nash equilibrium in Definition 2 is more likely to take a pure form like the one in (8).

The drawback of optimization (9) is that it only applies to the case where the desired Nash equilibrium is known and deterministic. If the desired Nash equilibrium is mixed, i.e., each player has a preferred probability distribution over all actions rather than one single preferred action, then the semidefinite program is no longer useful.

B. Bilevel optimization approach

We now consider the case where the desired Nash equilibrium is described by a performance function, rather than explicitly as a desired joint strategy. In particular, we consider the following continuously differentiable function that evaluates the quality of a joint strategy

\[
\psi : \mathbb{R}^m \to \mathbb{R}
\]

(10)

For example, if \( x^* = [(x^*_1)^T \ (x^*_2)^T \ \ldots \ (x^*_n)^T]^T \) is the desired joint Nash equilibrium, then a possible choice of function \( \psi \) is as follows:

\[
\psi(x) = D_{KL}(x, x^*) := \sum_{i=1}^n (x^*_i - \ln(x_i) - \ln(x^*_i)).
\]

(11)

The above choice of function \( \psi(x) \) measures the sum of the Kullback–Leibler (KL) divergence between each player’s strategy and the corresponding desired strategy.

In order to compute the value of matrix \( C \) such that the Nash equilibrium in Definition 2 is unique and minimizes the value of performance function \( \psi(x) \), we introduce the following bilevel optimization problem:

\[
\begin{aligned}
\min_{x \in \mathbb{R}^m} & \quad \psi(x) \\
\text{subject to} & \quad C + C^T \succeq 0, \quad \| C \|_F \leq \rho,
\end{aligned}
\]

\[
\]

(12)

Here \( \rho \in \mathbb{R}_+ \) is a tuning parameter that controls the maximum allowed Frobenius norm of matrix \( C \). Intuitively,
the larger the value of $\rho$, the more choices of matrix $C$ from which we can choose, and the more likely we can achieve a lower value of function $\psi(x)$.

The drawback of optimization (12) is that, unlike the semidefinite program in (9), it is nonconvex and, as a result, one can only hope to obtain a local optimal solution in general. However, we can compute such an local optimal solution efficiently using the projected gradient method, as we will show next.

1) Differentiating through the Nash equilibrium condition: The key to solve bivel optimization (12) is to compute the gradient of $\psi(x)$ with respect to matrix $C$. In particular, we let $\nabla C \psi(x) \in \mathbb{R}^{m \times m}$ be the matrix whose $hl$-th element, denoted by $[\nabla C \psi(x)]_{pq}$, is given by

$$[\nabla C \psi(x)]_{pq} := \frac{\partial \psi(x)}{\partial C}_{pq} \quad (13)$$

for all $p, q \in [m]$. Since function $\psi$ is continuously differentiable, the difficulty in evaluating $\nabla C \psi(x)$ is to compute the Jacobian of the Nash equilibrium $x$ with respect to matrix $C$. To this end, we introduce the following notation:

$$u := -\frac{1}{\lambda} (b + Cx) \quad (14a)$$

$$f(u) := [f_1(u_1)^T \ f_2(u_2)^T \ \cdots \ f_n(u_n)^T]^T \quad (14b)$$

where $u_i \in \mathbb{R}^{m_i}$, for all $i \in [n]$, and $f_i$ is given by Lemma 1.

The following result provides a formula to compute $\nabla C \psi(x)$ using the implicit function theorem [16].

**Proposition 2.** Suppose $C + C^T \succeq 0$ and $\lambda > 0$. Let $x := [x_1^T \ x_2^T \ \cdots \ x_n^T]^T$ be such that (5) holds for all $i \in [n]$, $\psi : \mathbb{R}^{m} \rightarrow \mathbb{R}$ be a continuously differentiable function, and $f(u)$ given by (14). If $I_m + \frac{1}{\lambda} \partial_u f(u)C$ is nonsingular, then

$$\nabla C \psi(x) = -\frac{1}{\lambda} \partial_u f(u)^T (I_m + \frac{1}{\lambda} \partial_u f(u)C)^{-T} \nabla_x \psi(x)x^T.$$  

**Proof.** Let $F(x, C) := x - f(u)$ and $C_q$ denote the $q$-th column of matrix $C$. Proposition 1 implies $x$ is the unique vector that satisfies $F(x, C) = 0_m$. Since $f$ is a continuously differentiable function, the implicit function theorem [16, Thm. 1B.1] implies the following: if $\partial_x F(x, C)$ is nonsingular, then $\frac{\partial x}{\partial C_q} = -\frac{1}{\partial_x F(x, C)}^{-1} \partial C_q F(x, C)$. Using the chain rule we can show $\partial_x F(x, C) = I_m + \frac{1}{\lambda} \partial_u f(u)C$ and $\partial C_q F(x, C) = \frac{1}{\lambda}[x_j] \partial_u f(u)$. The rest of the proof is due to the chain rule and the definition of $\nabla C \psi(x)$ in (13). \qed

The gradient formula in Proposition 2 requires computing matrix inverse, which can be numerically unstable. In practice, we use the following least-squares-based formula:

$$\hat{\nabla} C \psi(x) := -\frac{1}{\lambda} \partial_u f(u)^T M \quad (15)$$

where

$$M \in \arg \min_{X \in \mathbb{R}^{m \times m}} \left\| (I_m + \frac{1}{\lambda} \partial_u f(u)C)^T X - \nabla_x \psi(x)x^T \right\|_F^2. \quad (16)$$

Note that if $I_m + \frac{1}{\lambda} \partial_u f(u)C$ is nonsingular, then Proposition 2 implies $\hat{\nabla} C \psi(x) = \nabla C \psi(x)$; otherwise, the value of $\hat{\nabla} \psi(x)$ provides only an approximation of $\nabla C \psi(x)$.

2) Approximate projected gradient method: Equipped with Proposition 2 and the projection formula in (15), we are now ready to introduce the approximate projected gradient method for bilevel optimization (12). To this end, we define the following closed convex set:

$$D := \{ C \in \mathbb{R}^{m \times m} | C + C^T \succeq 0, \|C\|_F \leq \rho \}. \quad (17)$$

We summarize the approximate projected gradient method in Algorithm 1, where the projection map $\Pi_D : \mathbb{R}^{m \times m} \rightarrow \mathbb{R}^{m \times m}$ is given by

$$\Pi_D(C) = \arg \min_{X \in D} \|X - C\|_F \quad (18)$$

for all $C \in \mathbb{R}^{m \times m}$. At each iteration, this method first solve the nonlinear least-squares problem in (7), then update matrix $C$ using the approximate gradient in (15).

**Algorithm 1 Approximate projected gradient method.**

**Input:** Function $\psi : \mathbb{R}^{m} \rightarrow \mathbb{R}$, vector $b \in \mathbb{R}^{m}$, scalar weight $\lambda \in \mathbb{R}_{++}$, step size $\alpha \in \mathbb{R}_{++}$, stopping tolerance $\epsilon$.

1: Initialize $C = 0_{m \times m}$, $C^+ = 2\epsilon I_m$.
2: while $\|C^+ - C\|_F > \epsilon$ do
3: $C \leftarrow C^+$.
4: Solve optimization (7) for $x$.
5: $C^+ \leftarrow \Pi_D(C - \alpha \hat{\nabla} C \psi(x))$.
6: end while

**Output:** Nash equilibrium $x$ and cost matrix $C$.

A key step in Algorithm 1 is to compute the projection in (18). The following lemma provides the explicit computational formula for computing this projection via eigenvalue decomposition and matrix normalization.

**Lemma 2.** Let $\mathbb{D}$ be given by (17). Let $C \in \mathbb{R}^{m \times m}$, $U \in \mathbb{R}^{m \times m}$, and $s \in \mathbb{R}^{m}$ be such that $U \text{diag}(s)U^T = \frac{1}{2}(C + C^T)$. Then

$$\Pi_D(C) = \frac{\rho}{\max\{\rho, \|A\|_F\}} A, \quad (19)$$

where $A := \frac{1}{2}(C - C^T) + U \text{diag}(\max(s, 0))U^T$.

**Proof.** First, we prove that matrix $A \in \arg \min_{Z \in \mathbb{D}} \|Z - C\|_F$ where $\mathbb{D} := \{ C \in \mathbb{R}^{m \times m} | C + C^T \succeq 0 \}$. To this end, we let $X = Z - \frac{1}{2}(C + C^T)$. Then $A \in \arg \min_{Z \in \mathbb{D}} \|Z - C\|_F$ if and only if $X \in \text{minimize} \quad \|X - \frac{1}{2}(C + C^T)\|_F$, where $S_+$ is the set of positive semidefinite matrices. Therefore we conclude $A \in \arg \min_{Z \in \mathbb{D}} \|Z - C\|_F$, due to the results in [17, Ex. 29.32]. The rest of the proof is a direct application of [18, Thm. 7.1]. \qed

**IV. NUMERICAL EXAMPLES**

We demonstrate the application of the numerical methods in Section III using two examples. In these examples, we aim to infer the cost matrices—which can be interpreted as subsidies and tolls—that encourage desired behavior, such as collision avoidance in multi-robot path-planning and fairness...
in delivery service. Throughout, we compute the entropy regularized Nash equilibrium in Definition 2 by solving optimization (7) using the Gauss-Newton method with line search [15, Sec. 10.3].

A. Encouraging collision avoidance

We consider four ground rovers placed in a two-dimensional environment, at coordinate (0, 1), (0, -1), (1, 0), and (-1, 0), respectively. Each rover wants to reach the corresponding target position with coordinates (0, -1), (0, 1), (-1, 0) and (1, 0), respectively. Each rover can choose one of three candidate paths that connects its initial position to its target position: a beeline path of length 2; two semicircle paths, each of approximate length \(\pi\), one in clockwise direction, the other one in counterclockwise direction. We assume all rovers start moving at the same time and move at the same speed.

We model the decision-making of each rover using the entropy-regularized matrix game in Section II. In particular, we let \(\lambda = 0.1\), \(n = 4\), \(m = 12\), and \(b_i = \left[ \frac{2}{\pi} \quad \pi \right] \top\) for all \(i = 1, 2, 3, 4\). Here the elements in \(b_i\) denote the length of each candidate path. If \(C = 0_{12 \times 12}\), one can verify–by solving an instance of optimization (7)–that the Nash equilibrium in Definition 2 is approximately

\[
x_i = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \quad i = 1, 2, 3, 4.
\] (20)

In other words, all players tend to choose the beeline path since it has the minimum length. However, this causes collisions among the rovers at coordinate (0, 0).

By choosing a nonzero matrix \(C\), we aim to change the Nash equilibrium above to the following

\[
x_i^* = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}, \quad i = 1, 2, 3, 4.
\] (21)

In other words, we want all players to choose the counterclockwise semicircle path. See Fig. 3 for an illustration and https://www.youtube.com/watch?v=EvtPp DwqgU for an animation.

Since the Nash equilibrium in (21) is of the form in (8), we can compute matrix \(C\) using either the semidefinite program (9) or the bilevel optimization (12); in the latter case, we choose the performance function to be the KL-divergence in (11).

We solve the semidefinite program (9) using the off-the-shelf solver, and the bilevel optimization (12) using Algorithm 1. Fig. 4 shows the tradeoff between \(D_{KL}(x, x^*)\)–which measures the distance between the Nash equilibrium \(x^*\) and the desired Nash equilibrium \(x\)–and \(||C||_F\), of the computed matrix \(C\) when tuning the parameter in (9) and (12). These results confirm that both the semidefinite program approach and the bilevel optimization approach apply to the cases where the desired Nash is pure and known explicitly. Furthermore, both approaches require a careful tuning of algorithmic parameters to achieve a preferred trade-off between \(D_{KL}(x, x^*)\) and \(||C||_F\).

B. Encouraging fair resource allocation

We now consider a case where the desired Nash equilibrium is not of the explicit form in (8). Instead, we only have access to a performance function that implicitly describes the desired Nash equilibrium. To this end, we consider the following three-player game. Each player is a delivery drone company that provides package-delivery service, located at the southwest, southeast, and east area of Austin, respectively. Each strategy denotes the distribution of service allocated to the nine areas of Austin; we assume all three companies have the same amount of service to allocate. For each company, within its home area (where it is located), the operating cost of delivery service is one unit; outside the home area, the operating cost increases by 50% in an area adjacent to the home area, and 80% otherwise. See Fig. 5 for an illustration.3 We model the joint decision of the three companies using the matrix game in Section II where \(n = 3\), \(m_i = 9\) for \(i = 1, 2, 3\), and \(m = 27\); we set \(\lambda = 0.1\) and vector \(b\) according to the aforementioned operating cost.

If all companies only consider the operating cost, they will only allocate service to their respective home area. We aim to infer the value of matrix \(C\) using Algorithm 1 that encourages a fair allocation to other areas. In particular, we choose the performance function as follows:

\[
\psi(x) = 1^\top(x_1 + x_2 + x_3)^{-1},
\] (22)

where vector \((x_1 + x_2 + x_3)^{-1}\) denotes the elementwise reciprocal of vector \(x_1 + x_2 + x_3\). Function \(\psi(x)\) is based on the potential delay function from the resource allocation

3 Picture credit: https://en.wikipedia.org/wiki/List_of_Austin_neighborhoods
literature; the latter is a special case of the more general α-fairness function [19, Sec. 2.4]. Here function $\psi(x)$ measures the overall fairness of the delivery service allocation.

We compute the cost matrix using Algorithm 1 and illustrates the percentages of the delivery service allocated to each area at the Nash equilibrium in Fig. 6. The results show that, when $\rho \approx 0$, all the drone fleets will almost only serve their respective home areas. As we increases the value of $\rho$, the computed matrix encourages a more fair joint strategy where all nine areas receive almost equal amount of service.

V. CONCLUSION

We study the inverse game problem in the context of multiplayer matrix game. We provide sufficient conditions for the uniqueness of the Nash equilibrium, and efficient numerical methods that ensure these sufficient conditions. By guaranteeing the equilibrium uniqueness, our work ensures that mechanism design for multiplayer games do not result in unintended and undesired behavior. Future directions include extensions to games with other equilibrium concepts, continuous strategy spaces, and temporal dynamics.

ACKNOWLEDGMENT

The authors would like to thank Yigit E. Bayiz, Shenghui Chen, Dayou Luo, and Shahriar Talebi for their feedback.