

# Computing Nash Equilibria in Potential Games with Private Uncoupled Constraints

Nikolas Patris,<sup>1,2</sup> Stelios Stavroulakis,<sup>1</sup> Fivos Kalogiannis,<sup>1,2</sup> Rose Zhang,<sup>1</sup> Ioannis Panageas,<sup>1</sup>

<sup>1</sup>University of California, Irvine,

<sup>2</sup>Archimedes Research Unit

{npatris, sstavrou, fkalogia, rwzhang}@uci.edu, ipanagea@ics.uci.edu

## Abstract

We consider the problem of computing Nash equilibria in potential games where each player’s strategy set is subject to private uncoupled constraints. This scenario is frequently encountered in real-world applications like road network congestion games where individual drivers adhere to personal budget and fuel limitations. Despite the plethora of algorithms that efficiently compute Nash equilibria (NE) in potential games, the domain of constrained potential games remains largely unexplored. We introduce an algorithm that leverages the Lagrangian formulation of NE. The algorithm is implemented independently by each player and runs in polynomial time with respect to the approximation error, the sum of the size of the action-spaces, and the game’s inherent parameters.

## 1 Introduction

The modeling and studying of games with constraints has received a lot of attention in various areas including control (Hsiao and Lazar 1991), transportation and routing in traffic networks ((Jeff) Ban et al. 2019; Larsson and Patriksson 1998), telecommunications (Pang et al. 2008), (Zhu 2008a), markets (Arrow and Debreu 1954), cloud computing (Ardagna, Panicucci, and Passacantando 2011), multi-agent RL (Chen, Ma, and Zhou 2022), even in managing environmental pollution (Bretton, Zaccour, and Zahaf 2006). There are two main models in the literature: settings in which each agent has their own private constraints (also called orthogonal constraints) and settings in which the constraints are the same for all agents (common couple constraints). It is worth noting that having arbitrary constraints might lead to non-existence of a Nash equilibrium, i.e., John Nash’s theorem (Nash 1951) is not applicable, even for two player zero-sum games (Altman and Solan 2009). A Nash equilibrium in constrained games, also known as *Generalized Nash Equilibrium*, is defined to be a feasible strategy profile so that each agent does not have an incentive to unilaterally, *feasibly* deviate and decrease their cost. In case of  $\epsilon$ -approximate Nash equilibrium, we borrow the definition used in (Jordan, Lin, and Zampetakis 2023). Specifically, an  $\epsilon$ -approximate Nash equilibrium is a strategy profile that is approximately feasible (inequality constraints are violated by at most an additive  $\epsilon$ )

and moreover each agent does not have an incentive to unilaterally deviate among the approximately feasible strategies of their constraints and decrease their cost.

A common sufficient condition that guarantees existence of Nash equilibria in constrained games is Slater’s condition (Bertsekas 1997). Slater’s condition is a constraint qualification (CQ) stating that for every strategy profile  $\mathbf{x} := (\mathbf{x}_i, \mathbf{x}_{-i})$  of the agents (feasible or infeasible), each agent  $i$  can deviate and create a strategy profile  $(\mathbf{x}'_i, \mathbf{x}_{-i})$  that is strictly feasible (see Assumption 3.3) for their own constraints.

In this paper, our primary focus is on the computation of Nash equilibria in potential games, as defined in Section 2, with private constraints, specifically in the context of normal-form games. The constraints we consider in this study are assumed to be convex. The necessity of this assumption to establish arguments regarding approximate Nash equilibria is thoroughly examined in Section 3. While the constraints themselves are assumed to be convex, it is important to note that the potential is non-convex. This non-convexity poses a significant technical challenge that has yet to be addressed in the existing literature. However, in this paper, we tackle this challenge effectively.

**Our contributions** Our paper focuses on the problem of computing approximate Nash equilibrium in normal form potential games with private convex constraints. We present Algorithm  $\text{IGD}_\lambda$  that uses projected gradient descent on a carefully constructed Lagrange function that reaches an  $\epsilon$ -approximate Nash equilibrium after  $O(1/\epsilon^6)$  iterations. We note that our algorithm is implemented in a distributed manner in the sense that each agent performs projected gradient on their own cost function and uses information about their own constraints.

**Technical overview** We present a concise roadmap of the main contributions. In Section 4, we outline the steps taken to prove Theorem 4.12. Following the standard procedure for constraint optimization, we formulate a Lagrangian problem that incorporates the constraints. Our objective is to identify a stationary point of the Lagrangian. However, a challenge arises when dealing with multipliers  $\lambda$  in bounded domains, as the commonly used first-order methods are designed for unbounded domains. Consequently, there is uncertainty regarding whether the first-order stationary points obtained

using these methods are true stationary points in the unconstrained case or if they arise due to the imposed restrictions. To address this issue, we introduce a regularization term to restrict the domain of the lambdas effectively. This regularization introduces a new technical consideration: the choice of regularization parameter significantly impacts the solution quality. Nevertheless, we successfully establish bounds and determine the appropriate relationship between the solution quality and the regularization constant.

**Related work** A common technique in constrained optimization is to incorporate the constraints, i.e. define the Lagrange function and optimize over the domain of interest and the Lagrange multipliers (e.g., aim at computing an approximate Karush-Kuhn-Tucker (KKT) point). The KKT condition provides a general characterization of local optimality under various CQs (Kuhn and Tucker 1951; Karush 1939). Both the KKT condition and CQs can be extended to get a Generalized Nash equilibria.

The purpose of Lagrange multipliers is to penalize agents in situations where they deviate significantly from feasibility. Several studies have adopted a similar principle known as penalty-type algorithms. However, the selection of the penalty function may vary based on the specific constraints' structure. Notable representatives include exact and inexact penalty methods (Facchinei and Lampariello 2011; Fukushima 2011; Facchinei and Kanzow 2010; Kanzow and Steck 2018; Ba and Pang 2022) and exact and inexact augmented Lagrangian method (Pang and Fukushima 2005; Kanzow 2016; Kanzow and Steck 2016, 2018). For more information, see (Fischer, Herrich, and Schönefeld 2014) and references therein. Finally there have been a lot of works based on the Nikaido-Isoda function; these include gradient projection (Rosen 1965), relaxation method (Uryas'ev and Rubinstein 1994; Krawczyk and Uryasev 2000; von Heusinger and Kanzow 2009) and Newton methods (Facchinei, Fischer, and Piccialli 2009; von Heusinger, Kanzow, and Fukushima 2012; Dreves et al. 2013; Izmailov and Solodov 2014; Fischer et al. 2016). In the context of potential games the works of (Zhu 2008b; Scutari, Barbarossa, and Palomar 2006) are regarded as standard. Notably, this extensive body of literature has focused primarily on on convex functions with common, coupled, or uncoupled constraints. This assumption, however, does not align with real-world problems. Conversely, when the potential function is non-convex, the understanding of the constraint setting is limited. For coupled constraints, there are no guarantees beyond the simplest scenario where constraints are linear, see (Sagrattella 2017; Jordan, Lin, and Zampetakis 2023). The recent, concurrent, work of (Alatur et al. 2023) considers the more general problem of learning Nash policies in constrained Markov potential games. However, it fundamentally diverges from our approach. In contrast to our decentralized framework, enabling each agent to act independently, the solution proposed in (Alatur et al. 2023) is centralized. This implies that a central authority is responsible for determining which agent (only one) performs a best response update in each iteration.

## 2 Preliminaries

### Notation and Definitions

**Notation** Let  $\mathbb{R}$  be the set of real numbers, and  $[n] = \{1, 2, \dots, n\}$ . We define  $\Delta$  as the probability simplex, which is the set of  $n$ -dimensional probability vectors, i.e.,  $\Delta := \{\mathbf{x} \in \mathbb{R}^n : x_i \geq 0, \sum_{i=1}^n x_i = 1\}$ . We use  $\mathbf{e}_i$  to denote the  $i$ -th elementary basis vector, and to refer to a coordinate of a vector, we use either  $x_i$  or  $[x]_i$ . The superscripts are used to indicate the iterates of an algorithm. Lastly, it should be noted that all norms used correspond to the standard Euclidean norm,  $\|\cdot\| = \|\cdot\|_2$ .

**Normal-form Games** We consider  $n$  players, represented by the set  $\mathcal{N} := [n]$ , with each player  $i \in \mathcal{N}$  having a set of actions denoted by  $\mathcal{A}_i$ . The joint action profile is represented by  $\mathbf{a} := (a_1, a_2, \dots, a_n) \in \mathcal{A}$ , where  $\mathcal{A} := \times_{i \in \mathcal{N}} \mathcal{A}_i$  is the product of the action spaces. Players may also randomize their strategies by selecting a probability distribution over their set of actions. We use  $\mathbf{x}_i(a_i)$  to denote the probability that player  $i$  chooses action  $a_i \in \mathcal{A}_i$ . Since  $\mathbf{x}_i$  is a probability distribution, it must belong to the probability simplex, which we denote as  $\Delta(\mathcal{A}_i)$ . The set  $\Delta(\mathcal{A}_i)$  consists of all probability vectors  $\mathbf{x} \in \mathbb{R}_{\geq 0}^{|\mathcal{A}_i|}$  satisfying  $\sum_{a_i \in \mathcal{A}_i} \mathbf{x}(a_i) = 1$ . The product of simplices is denoted by  $\Delta^n := \times_{i \in \mathcal{N}} \Delta(\mathcal{A}_i)$ .

We consider the case where the players are trying to minimize their costs. For a given strategy profile  $\mathbf{a} \in \mathcal{A}$ , each player  $i$  receives a *cost*  $c_i(\mathbf{a})$ , where  $c_i : \mathcal{A} \rightarrow \mathbb{R}$ . In the case of a randomized strategy  $\mathbf{x}$ , i.e., a probability distribution over  $\Delta$ , we define the notion of *expected cost* as the expected value of the cost function  $C_i(\cdot)$  under the distribution  $\mathbf{x}$ . Specifically, the expected cost for player  $i$  is defined as  $\mathbb{E}_{\mathbf{a} \sim \mathbf{x}}[C_i(\mathbf{a})] = C_i(\mathbf{x})$ .

**Definition 2.1** (Approximate Nash equilibrium). A joint strategy profile  $(\mathbf{x}_1^*, \mathbf{x}_2^*, \dots, \mathbf{x}_n^*) \in \Delta^n$  is said to be an  $\epsilon$  approximate Nash equilibrium if for any player  $i \in \mathcal{N}$  and any possible unilateral deviation  $\mathbf{x}'_i \in \Delta(\mathcal{A}_i)$ , the resulting change in player  $i$ 's expected cost is no more than  $\epsilon$ .

$$C_i(\mathbf{x}'_i, \mathbf{x}_{-i}^*) \geq C_i(\mathbf{x}^*) - \epsilon$$

**Remark 2.2.** We note that the inequality in Definition 2.1 should be adjusted based on the objective of the problem, i.e., whether the players aim to maximize their utility or minimize their cost.

**Potential Games** A potential game is a type of game that has a single function  $\Phi(\mathbf{x}) : \Delta^n \rightarrow \mathbb{R}$ , referred to as a potential function, which captures the incentive of all players to modify their strategies. In other words, if a player deviates from their strategy, then the difference in payoffs is determined by a potential function  $\Phi$  evaluated at those two strategy profiles. We express this formally as follows:

**Definition 2.3** (Potential function). Consider a joint action profile  $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$ . For any player  $i \in \mathcal{N}$  and any unilateral deviation  $\mathbf{x}'_i$ , the difference in cost resulting from this deviation is reflected in the change in the potential function.

$$C_i(\mathbf{x}'_i, \mathbf{x}_{-i}) - C_i(\mathbf{x}) = \Phi(\mathbf{x}'_i, \mathbf{x}_{-i}) - \Phi(\mathbf{x})$$

### Problem Statement

This work addresses the problem of constrained potential games. In these games, multiple players are involved, and each player  $i \in \mathcal{N}$  has a private set of convex constraints. These constraints specify a feasibility set for each player, within which they aim to minimize their individual cost function  $C_i(\mathbf{x})$ .

As noted before, the potential function in constrained potential games captures the incentives of all players simultaneously. This allows us to formulate a single problem, referred to as the primal problem, that captures the individual players' problems collectively. By solving the primal problem, we are able to solve the constituent problems of each player simultaneously.

**Definition 2.4** (Primal problem). Let each player  $i \in \mathcal{N}$  have a set of  $d_i$  private convex constraints, denoted by  $g_{i,m}(\cdot)$  for any  $m \in [d_i]$ . Then, the players aim to solve the following optimization problem.

$$\begin{aligned} & \text{minimize} && \Phi(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n) \\ & \text{subject to} && g_{i,m}(\mathbf{x}_i) \leq 0, \quad m = 1, 2, \dots, d_i, \forall i \in \mathcal{N} \end{aligned} \quad (1)$$

The total number of constraints is denoted by  $d = \sum_{i \in \mathcal{N}} d_i$ , where  $\mathcal{N}$  represents the set of all players. Additionally, we refer to the feasibility set of player  $i$  as  $\mathcal{S}_i$ , which is defined by their respective constraints,  $g_{i,m}(\mathbf{x}_i) \leq 0$  for any  $m \in [d_i]$ .

$$\mathcal{S}_i = \{\mathbf{x}_i \in \Delta(\mathcal{A}_i) \mid g_{i,m}(\mathbf{x}_i) \leq 0 \text{ for all } m \in [d_i]\}$$

As expected, optimization in the constrained case presents additional difficulties. However, this is a well-studied class of optimization problems, and there are standard tools available, with the most prominent being the Lagrangian methods. The Lagrangian method involves defining a modified problem that takes the constraints into account. Specifically, we begin with the primal problem defined in Definition 2.4, and then define a new function that incorporates the constraints.

**Definition 2.5** (Lagrangian function). Consider a function  $\Phi(\cdot)$  to be minimized, subject to the constraints  $g_{i,m}(\mathbf{x}_i)$  for any player  $i \in \mathcal{N}$  and any constraint  $m \in [d_i]$ . For each inequality constraint, we introduce a non-negative multiplier  $\lambda_{i,m}$ , commonly known as a Lagrange multiplier.

$$\begin{aligned} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) &= \Phi(\mathbf{x}) + \sum_{i=1}^n \sum_{m=1}^{d_i} \lambda_{i,m} g_{i,m}(\mathbf{x}_i) \\ &= \Phi(\mathbf{x}) + \sum_{i=1}^n \boldsymbol{\lambda}_i^\top \mathbf{g}_i(\mathbf{x}_i) \\ &= \Phi(\mathbf{x}) + \boldsymbol{\lambda}^\top \mathbf{g} \end{aligned} \quad (2)$$

where  $\boldsymbol{\lambda}$  is a  $d$ -dimensional vector corresponding where  $d := (\sum_{i \in \mathcal{N}} d_i)$  is the overall number of constraints. The vector  $\boldsymbol{\lambda}$  is the concatenation of  $n$  vectors, where each vector has a length corresponding to the number of constraints for the corresponding player, i.e.,  $d_1, d_2, \dots, d_n$ , respectively.

$$\boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{\lambda}_1 \\ \vdots \\ \boldsymbol{\lambda}_n \end{bmatrix} \in \mathbb{R}^{d_1} \times \dots \times \mathbb{R}^{d_n} = \mathbb{R}^d \quad \text{and}$$

$$\mathbf{g}_i(\mathbf{x}_i) = \begin{bmatrix} g_{i,1}(\mathbf{x}_i) \\ \vdots \\ g_{i,d_i}(\mathbf{x}_i) \end{bmatrix} \in \mathbb{R}^{d_i} \quad \text{and} \quad \mathbf{g} = \begin{bmatrix} \mathbf{g}_1(\mathbf{x}_1) \\ \vdots \\ \mathbf{g}_n(\mathbf{x}_n) \end{bmatrix}$$

where  $\lambda_i \in \mathbb{R}_+^{d_i}$  for any  $i \in \mathcal{N}$  and  $\lambda_{i,m}$  correspond to the Lagrange multiplier of the  $m$ -th constraint of the  $i$ -th player.

**Lagrangian dual function** We define the Lagrangian dual function as the minimum value of the Lagrangian (2) over  $\mathbf{x} \in \Delta$ . It's important to note that this minimum can be achieved even if  $\mathbf{x}$  is not a feasible point, meaning it may not satisfy the constraints.

**Definition 2.6** (Lagrangian dual function). The Lagrangian dual function is defined as  $h(\boldsymbol{\lambda}) = \inf_{\mathbf{x} \in \Delta} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda})$  where  $\mathbf{x}$  belongs to the product of simplices of all players.

The Lagrangian dual function provides a lower bound on the optimal value of the primal problem defined in Definition 2.4, and this property is known as weak duality. The proof of this inequality is standard and can be found in many textbooks on optimization.

**Definition 2.7** (Weak duality). Consider the primal problem in Equation (1) and suppose  $p^*$  is its optimal value. Then, for any  $\boldsymbol{\lambda} \succeq \mathbf{0}$  we have  $h(\boldsymbol{\lambda}) \leq p^*$ .

**Dual problem** As we described above, the Lagrangian dual function provides a lower bound on the optimal value  $p^*$  of the primal problem for any value of  $\boldsymbol{\lambda}$ . It is natural to ask what the best possible lower bound for  $p^*$  would be. This leads to the formulation of another optimization problem commonly referred to as the *dual problem*.

**Definition 2.8** (Dual problem). Suppose the primal problem as defined in Definition 2.4. Then, the dual problem is defined as follows.

$$\text{maximize } h(\boldsymbol{\lambda}) \quad \text{subject to } \boldsymbol{\lambda} \succeq \mathbf{0} \quad (3)$$

There are also equivalent formulation of the primal, dual problems with respect to each player having the joint strategy profile of the other players  $\mathbf{x}_{-i}$  fixed. We defer those definitions along with some relevant lemmas to the Appendix A.

## 3 Nonconvex Constrained Games and Lagrangian Functions

In this section, we provide a concise overview of nonconvex games. We begin by introducing the problem and defining the appropriate solution concept for this class of games. Next, we delve into the importance of having convex constraints and analyze the implications when convexity is violated. Finally, we introduce the concept of a regularized Lagrangian, which plays a crucial role in our proposed solution.

### Nonconvex Games

It is evident that the goal is to find a Nash equilibrium  $\mathbf{x}$  for the constrained potential game. However, unlike the unconstrained case where the optimization is done over the entire space, in the constrained case, the optimization is restricted to a specific convex domain. In our case, the convex set is the product of the simplices, and so the notion of Nash equilibrium is defined as follows.

**Definition 3.1** (Approximate First Order Stationary Point). A joint strategy profile  $\mathbf{x} := (\mathbf{x}_1, \dots, \mathbf{x}_n) \in \Delta^n$  is called  $\epsilon$  approximate first order stationary point of function  $f$  as long as

$$\min_{(\mathbf{x}+\delta) \in \Delta^n, \|\delta\|^2 \leq 1} \delta^\top \nabla_{\mathbf{x}} f(\mathbf{x}) \leq \epsilon$$

**Solution Concept** The solution concept we consider in this work is commonly referred to as a nonlinear generalized Nash equilibrium (Jordan, Lin, and Zampetakis 2023). Unlike in the case where there are no constraints, in the constrained case, we extend the set to include unilateral deviations that are also approximately feasible. The formal definition of this concept is provided below.

**Definition 3.2** (Approximate feasible Approximate Nash equilibrium). A joint strategy profile  $(\mathbf{x}_1^*, \mathbf{x}_2^*, \dots, \mathbf{x}_n^*) \in \Delta^n$  is said to be an  $O(\epsilon)$  approximate feasible approximate Nash equilibrium if for any player  $i \in \mathcal{N}$  and any possible unilateral deviation  $\mathbf{x}'_i \in \{\mathbf{x}_i \in \Delta(\mathcal{A}_i) \mid g_i(\mathbf{x}_i) \leq \epsilon\}$ , the resulting change in player  $i$ 's expected cost is no more than  $O(\epsilon)$ .

$$C_i(\mathbf{x}'_i, \mathbf{x}_{-i}^*) \geq C_i(\mathbf{x}^*) - O(\epsilon)$$

**Techniques and Assumptions** To solve our problem, we leverage an important property of the potential function: convexity per player. This property, together with a standard assumption used in the literature (referred to as Slater's condition), yields a stronger condition (strong duality) for the relationship between the primal and dual optimal values.

**Assumption 3.3** (Slater's Condition). For any player  $i \in [n]$  and any constraint  $m \in [d_i]$ , there is strategy profile  $\tilde{\mathbf{x}}_i$  such that  $g_{i,m}(\tilde{\mathbf{x}}_i) < \xi_{i,m}$  for a strictly negative  $\xi_{i,m}$ .

Slater's condition requires the existence of a strictly feasible point under the constraints. Constraints qualifications, such as Slater's condition, are a standard way to obtain strong duality. It is worth noting that Slater's condition only requires the existence of a point that strictly satisfies the constraints, and does not require an a priori knowledge of any candidate optimal solution, such as regularity.

**Lemma 3.4** (Strong duality per player). For any player  $i \in [n]$  and for any joint strategy of the other players  $\mathbf{x}_{-i}$  along with their Lagrange multipliers  $\lambda_{-i}$ , strong duality holds.

$$\begin{aligned} & \min_{\mathbf{x}_i \in \Delta(\mathcal{A}_i)} \max_{\lambda_i \geq 0} \mathcal{L}_i(\mathbf{x}_i, \lambda_i; \mathbf{x}_{-i}, \lambda_{-i}) \\ &= \max_{\lambda_i \geq 0} \min_{\mathbf{x}_i \in \Delta(\mathcal{A}_i)} \mathcal{L}_i(\mathbf{x}_i, \lambda_i; \mathbf{x}_{-i}, \lambda_{-i}) \end{aligned} \quad (4)$$

where  $\mathcal{L}_i$  is the player-wise Lagrangian function defined in Appendix.

**Remark 3.5.** In the LHS of Equation (4), the minimizer  $\mathbf{x}$  selects their strategy first. It is crucial for  $\mathbf{x}$  to choose a feasible point that satisfies the constraints; otherwise, the Lagrange multipliers  $\lambda$  can be set arbitrarily large. Moreover, if  $\mathbf{x}$  is a feasible point, then it implies that the Lagrange multipliers  $\lambda$  must be zero, which results in an alternative expression for strong duality.

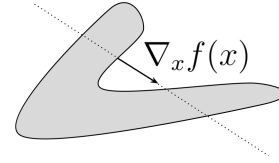


Figure 1: Nonconvex feasibility set

$$\min_{\mathbf{x}_i \in \mathcal{S}_i} \Phi(\mathbf{x}_i, \mathbf{x}_{-i}) = \max_{\lambda_i \geq 0} \min_{\mathbf{x}_i \in \Delta(\mathcal{A}_i)} \mathcal{L}_i(\mathbf{x}_i, \lambda_i; \mathbf{x}_{-i}, \lambda_{-i}) \quad (5)$$

where  $\mathcal{S}_i$  is the feasibility set of player  $i$ .

**Assumption 3.6.** For any player  $i \in \mathcal{N}$  and any constraint  $m \in [d_i]$ , the function  $g_{i,m}$  is convex and  $\gamma$ -smooth.

### Necessity of Convex Constraints

In this subsection, we provide a brief explanation of why the convexity of the constraints is a necessary condition for a first-order methods algorithm to find a Nash equilibrium.

In Figure 1, we encounter a problematic scenario that highlights a key challenge. Let's assume we have applied a first-order method and found a stationary point  $x$ . This stationary point suggests that in the neighborhood of  $x$ , further improvements in the objective function are not possible.

This scenario can also be interpreted within the context of zero-sum games. Consider the player  $x$ , who acts as the minimizer. In this situation,  $x$  does not have an incentive to move left as it would lead to being outside the feasible set. Hence, then the maximizer  $y$ , operating as the Lagrange multiplier in 2.5, can penalize  $x$  by setting  $y$  to infinity. Conversely, if  $x$  moves right, the objective performance would worsen. This phenomenon captures the notion of a local min-max point.

**Definition 3.7** (Local min-max point (Daskalakis and Panageas 2018)). A critical point  $(x^*, y^*)$  is a local min-max point if there exists a neighborhood  $U$  around  $(x^*, y^*)$  so that for all  $(x, y) \in U$  we have that  $f(x^*, y) \leq f(x^*, y^*) \leq f(x, y^*)$ .

However, relying solely on the concept of a local min-max point is insufficient for our objective. As depicted in Figure 1, although neither player has an incentive to change their strategy within a local neighborhood (i.e., for  $\epsilon$  deviations), we observe that there exists a direction along the extension of the gradient line, which, unfortunately, falls into the feasibility set.

### Regularized Lagrangian

In this subsection, we provide a brief explanation of why regularization is an essential component of our proposed solution. To understand this, let us discuss the method of Lagrange multipliers briefly. For a more detailed exposition, we refer to (Boyd and Vandenberghe 2004).

**Definition 3.8** (Regularized Lagrangian function). Suppose  $\mathcal{L}(\mathbf{x}, \lambda)$  is the Lagrangian function as defined in Definition 2.5. We can define the regularized version  $\tilde{\mathcal{L}}$  by adding

the regularization term  $-\mu\|\lambda\|^2$  as follows:

$$\tilde{\mathcal{L}}(\mathbf{x}, \lambda) := \mathcal{L}(\mathbf{x}, \lambda) - \mu\|\lambda\|^2$$

As previously mentioned, the intuition behind the Lagrange method is to introduce a new function that incorporates the constraints, transforming the problem into an unconstrained one. The Lagrange multipliers act as linear penalty terms on the objective. Consequently, whenever  $\mathbf{x}$  is an infeasible point, that means it does not satisfy the constraints, the maximization in the dual problem (as defined in Definition 2.8) can set  $\lambda$  arbitrarily large.

The key distinction lies in the domain of the Lagrange multipliers  $\lambda$ . In the original formulation, the domain of  $\lambda$  is unbounded, allowing for arbitrary values. Most widely-used methods for computing first-order stationary points are designed specifically for optimization problems with bounded domains. However, when we introduce bounded domains for the Lagrange multipliers  $\lambda$ , a potential issue arises: we cannot determine if the first-order stationary points obtained using these methods are genuine ( $\lambda \succeq \mathbf{0}$ ) stationary points in the unconstrained case or if they are introduced due to the restriction. In other words, these bounded domains may prevent the optimizer from reaching the true optimal point that would exist in the unconstrained case.

The regularization term in our formulation plays an important role for two reasons. Firstly, it indirectly limits the magnitude of the Lagrange multipliers. Although one might argue that this modifies the original game, it is important to note that as the regularization parameter  $\mu$  approaches zero, the regularized game converges to the original game. Moreover, for sufficiently small values of  $\mu$ , it governs the alternation of the game. In other words, it allows us to control the impact of the regularization based on how closely the solutions approach (in value) the unconstrained case. As we will demonstrate shortly, the parameter  $\mu$  also controls the approximation of the first-order stationary point. Another important aspect of regularization is that it makes the problem with respect to  $\lambda$  strongly concave. Strong concavity guarantees the uniqueness of a maximizer and assists the analysis.

## 4 Main Result

The main contribution of this section is divided into two parts. First, we present an algorithm specifically designed for computing first-order stationary points, as defined in Definition 3.1. This algorithm serves as the core computational tool of our approach. Then, we provide a complete set of statements and proofs that establish the correctness of our solution. These statements form the basis for understanding the theoretical underpinnings of our approach.

### Algorithm IGD $_{\lambda}$

We present here our proposed solution. Algorithm IGD $_{\lambda}$  is a natural and intuitive procedure that essentially performs (projected) gradient descent in a special function,  $\phi(\mathbf{x}) = \max_{\lambda} \tilde{\mathcal{L}}(\mathbf{x}, \lambda)$ . The first step of the algorithm involves a maximization step, which can be efficiently performed without the need for a max-oracle, thanks to the

---

**Algorithm 1:** IGD $_{\lambda}$ : Independent Gradient Descent on  $\phi(\cdot) = \max_{\lambda} \tilde{\mathcal{L}}(\cdot, \lambda)$  of the regularized Lagrangian  $\tilde{\mathcal{L}}$

---

```

1 Output:  $\hat{\mathbf{x}}$  first order stationary point.
2 Initialize:  $\mathbf{x}^{(0)} \in \Delta^n$ 
3 for  $t = 1, 2, \dots, T$  do
4    $\lambda^{(t)} \leftarrow \operatorname{argmax}_{\lambda} (\lambda^{\top} \mathbf{g}(\mathbf{x}^{(t)}) - \mu\|\lambda\|^2)$ 
5    $\mathbf{x}_i^{(t+1)} \leftarrow$ 
      $\Pi_i \left( \mathbf{x}_i^{(t)} - \eta \left( \nabla_{\mathbf{x}_i} C_i(\mathbf{x}^{(t)}) + \lambda_i^{\top} \nabla_{\mathbf{x}_i} \mathbf{g}_i(\mathbf{x}_i^{(t)}) \right) \right)$ 
      $\blacktriangleright$  for all players  $i \in \mathcal{N}$ 
6 Return  $\hat{\mathbf{x}} = (\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_n)$ 

```

---

structure of  $\phi(\mathbf{x})$  as explained in Section 4. Then, each agent independently performs a step of projected gradient descent on  $\tilde{\mathcal{L}}$ . However, despite its simplicity, Algorithm IGD $_{\lambda}$  provides strong guarantees, as we will demonstrate next.

### Analysis of Algorithm IGD $_{\lambda}$

In this section, we present a complete proof of our main theorem. We provide a clear roadmap outlining the steps we will take to establish the proof. First, we will argue about the smoothness of the regularized Lagrangian function. By doing so, we will be able to establish the smoothness of the function  $\phi(\cdot) = \max_{\lambda} \tilde{\mathcal{L}}(\cdot, \lambda)$ . Next, we will prove some boundedness results on the values of  $\lambda$ , both for the case of (approximately) optimal values and for arbitrary values. These bounds are important to ensure that the optimization process remains well-behaved.

**Lemma 4.1** (Strongly Concavity). *Let  $\hat{\mathbf{x}}$  be an arbitrary joint strategy profile. Then, the function  $\tilde{\mathcal{L}}(\hat{\mathbf{x}}, \lambda) = \mathcal{L}(\hat{\mathbf{x}}, \lambda) - \mu\|\lambda\|^2$  is strongly concave in  $\lambda$ , and so the maximizer  $\hat{\lambda} = \operatorname{argmax}_{\lambda} \tilde{\mathcal{L}}(\hat{\mathbf{x}}, \lambda)$  is unique.*

**Lemma 4.2** (Bounded norm of multipliers). *Let  $\hat{\mathbf{x}}$  be an arbitrary joint strategy profile. Then, the maximizer  $\hat{\lambda} = \operatorname{argmax}_{\lambda} \{\tilde{\mathcal{L}}(\hat{\mathbf{x}}, \lambda)\}$  has bounded norm, i.e.  $\|\hat{\lambda}\| \leq \frac{\sqrt{d}G_{\max}}{2\mu} = \Lambda_{\max}$ , where  $G_{\max} = \max_{i \in \mathcal{N}} \max_{m \in [d_i]} \max_{\mathbf{x}_i \in \Delta(\mathcal{A}_i)} g_{i,m}(\mathbf{x}_i)$ .*

Having established the strong concavity and boundedness of the Lagrange multipliers, we now continue on showing the smoothness of  $\tilde{\mathcal{L}}$ . We use  $A_{\max}$  to denote the maximum number of actions over all agents, i.e.  $A_{\max} := \max_{i \in \mathcal{N}} |\mathcal{A}_i|$  and  $\Phi_{\max} := \max_{\mathbf{x}} \Phi(\mathbf{x})$  for the maximum value of the potential function.

**Lemma 4.3** (Smoothness of  $\Phi$ ). *The potential function  $\Phi$  is  $(nA_{\max}\Phi_{\max})$ -smooth.*

**Lemma 4.4** (Smoothness of  $\mathcal{L}$ ). *The Lagrangian function  $\mathcal{L}$  is  $(nA_{\max}\Phi_{\max} + \Lambda_{\max}\gamma)$ -smooth, where  $\gamma$  is the smoothness of the constraints.*

By combining Lemma 4.3 and Lemma 4.4, and applying the triangle inequality, we can establish the boundedness of the spectral norm of  $\tilde{\mathcal{L}}$ .

**Lemma 4.5** (Smoothness of  $\tilde{\mathcal{L}}$ ). *The regularized Lagrangian  $\tilde{\mathcal{L}}$  is  $(nA_{\max}\Phi_{\max} + \Lambda_{\max}\gamma + 2\mu)$ -smooth.*

For clarity, let us define  $\ell$  as the smoothness parameter of the regularized Lagrangian, such that  $\ell = (nA_{\max}\Phi_{\max} + \Lambda_{\max}\gamma + 2\mu)$  from Lemma 4.5. The following proposition follows from the above statements.

**Proposition 4.6** (Properties of  $\tilde{\mathcal{L}}$ ). *The regularized Lagrangian  $\tilde{\mathcal{L}}$  satisfies the following properties:*

1.  $\tilde{\mathcal{L}}$  is  $\ell$ -smooth and  $\tilde{\mathcal{L}}(\mathbf{x}, \cdot)$  is  $\mu$ -strongly concave.
2. The domain of  $\lambda$  is bounded, i.e.  $\|\lambda\| \leq \Lambda_{\max}$ .

Let  $\kappa = \ell/\mu$  denote the condition number and define  $\phi(\cdot) = \max_{\lambda} \tilde{\mathcal{L}}(\cdot, \lambda)$  and  $\lambda^*(\cdot) = \operatorname{argmax}_{\lambda} \tilde{\mathcal{L}}(\cdot, \lambda)$ .

**Lemma 4.7** (Lemma 4.3 (Lin, Jin, and Jordan 2020)). *Under Proposition 4.6,  $\phi(\cdot)$  is  $(\ell + \kappa\ell)$ -smooth with  $\nabla\phi(\cdot) = \nabla_{\mathbf{x}}\tilde{\mathcal{L}}(\cdot, \lambda^*(\cdot))$ . Also  $\lambda^*(\cdot)$  is  $\kappa$ -Lipschitz.*

**Lemma 4.8** (Approximate Stationary Point of  $\phi$ ). *Let the learning rate  $\eta$  be  $1/\beta$ , where  $\beta = c/\mu$  and  $c = 4((nA_{\max})^2 + (\Lambda_{\max}\gamma)^2)$  is constant. If we run Algorithm IGD $_{\lambda}$  for  $T = \frac{32}{\epsilon^2\mu}(\Phi_{\max} + \Lambda_{\max}\sqrt{d}G_{\max})((nA_{\max})^2 + (\Lambda_{\max}\gamma)^2)$ , then there exists a timestep  $t \in \{1, 2, \dots, T\}$  such that  $\mathbf{x}^{(t)}$  is an  $\epsilon$ -approximate first order stationary point of  $\phi$ .*

*Proof.* The first step involves demonstrating that the function  $\phi(\cdot) = \max_{\lambda} \tilde{\mathcal{L}}(\cdot, \lambda)$  is a smooth function. Based on the result from Lemma 4.5, we know that  $\tilde{\mathcal{L}}$  is  $\ell$ -smooth, with  $\ell = (nA_{\max}\Phi_{\max} + \Lambda_{\max}\gamma + 2\mu)$ . Utilizing the smoothness result in Lemma 4.7, we can conclude that  $\phi(\cdot)$  is also smooth, with a parameter  $(\ell + \kappa\ell)$ , where  $\kappa = \ell/\mu$ .

Next, by applying the descent lemma (Lemma C.2) on  $\phi$ , we establish that the sequence of successive steps will invariably form a non-increasing sequence. This can be inferred from Lemma C.2, which ensures that  $\phi(\mathbf{x}^{(t+1)}) - \phi(\mathbf{x}^{(t)}) \leq -\frac{1}{2\beta}\|\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)}\|_2^2$ . Furthermore, Theorem C.4 guarantees the existence of at least one timestep  $t \in \{1, 2, \dots, T\}$  where  $\|\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)}\|_2^2 \leq \frac{2\beta\delta_{\phi}}{T}$ , with  $\delta_{\phi} = \phi(\mathbf{x}^{(0)}) - \phi(\mathbf{x}^*)$ . Bounding  $\delta_{\phi}$  by  $(\Phi_{\max} + \Lambda_{\max}\sqrt{d}G_{\max})$ , setting  $\beta = c/\mu$  and

$$T = \frac{32}{\epsilon^2\mu}(\Phi_{\max} + \Lambda_{\max}\sqrt{d}G_{\max})((nA_{\max})^2 + (\Lambda_{\max}\gamma)^2)$$

we get that  $\|\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)}\|_2 \leq \epsilon/2$ . Then, from Proposition A.4 and Remark A.5 we conclude that  $\mathbf{x}^{(t+1)}$  is an  $\epsilon$ -approximate first order stationary point of  $\phi$ .  $\square$

To establish the time complexity, we are free to choose the value of  $\mu$ . By setting  $\mu = O(\epsilon)$ , we determine the dependence of the running time of Algorithm IGD $_{\lambda}$  on  $\epsilon$ .

**Corollary 4.9.** *Algorithm IGD $_{\lambda}$  runs in  $T = O(1/\epsilon^6)$  iterations to find an  $\epsilon$  approximate first order stationary point of  $\phi$ .*

To proceed, we can utilize Lemma 4.7, which states that  $\nabla\phi(\cdot) = \nabla_{\mathbf{x}}\tilde{\mathcal{L}}(\cdot, \lambda^*(\cdot))$ , where  $\lambda^*(\cdot)$  represents the unique maximizer for a given  $\mathbf{x}$ , as indicated in Lemma 4.1. Therefore, this implies that the point  $\hat{\mathbf{x}}$  obtained from the algorithm is a first order stationary point of the regularized Lagrangian. Additionally, since the regularization term does not depend on the  $\mathbf{x}$  variable, it follows that  $\hat{\mathbf{x}}$  is a first order stationary point of the Lagrangian as defined in 2.5.

The next step involves demonstrating that  $\hat{\lambda} = \operatorname{argmax}_{\lambda} \mathcal{L}(\hat{\mathbf{x}}, \lambda)$  necessarily resides in the interior of its domain, meaning  $\|\lambda\| \leq \Lambda_{\max}$ . More precisely, we are looking for a bound that is independent of  $\mu = O(\epsilon)$ . To accomplish this, we utilize the Slater's condition.

**Lemma 4.10** (Bounded optimal multipliers). *Let  $\hat{\mathbf{x}}$  represent the  $\epsilon$ -approximate first order stationary point returned by Algorithm IGD $_{\lambda}$ . Then, for each set of Lagrange multipliers  $\hat{\lambda}_i$ , where  $\hat{\lambda}_i = \operatorname{argmax}_{\lambda_i} \mathcal{L}_i(\hat{\mathbf{x}}_i, \lambda_i; \hat{\mathbf{x}}_{-i}, \hat{\lambda}_{-i})$ , we can bound them by  $\frac{2(\Phi_{\max} - \Phi_{\min})}{\xi_{i,m}}$  component-wise, where  $\xi_{i,m}$  are defined in Assumption 3.3.*

Finally, we need to consider the (approximate) optimality with respect to the potential function  $\Phi$ . So far, our discussion has mainly focused on the Lagrangian, but we have not directly addressed the optimality of the potential function and as a result the individual players' value functions. In other words, we need a relation that associates the first order stationary point with a guarantee with respect to the value of  $\Phi$ .

**Lemma 4.11** (Approximate Optimality of  $\Phi$ ). *Given an  $\epsilon$  approximate stationary point  $(\hat{\mathbf{x}}, \hat{\lambda})$  of  $\tilde{\mathcal{L}}$ , we get that  $\hat{\mathbf{x}}$  is  $O(\epsilon)$  approximate feasible approximate Nash equilibrium, as in Definition 3.2.*

**Theorem 4.12** (Main Theorem). *Assuming that all agents perform Algorithm IGD $_{\lambda}$ , after  $T = O(1/\epsilon^6)$  steps, there exists an iterate  $\mathbf{x}^{(t)}$  for  $t \in [T]$ , so that  $\mathbf{x}^{(t)}$  is an  $O(\epsilon)$  approximate feasible approximate Nash equilibrium.*

## 5 Numerical Experiments

In this section, we empirically validate our theoretical results using constrained congestion games as a testing ground. Relying on the framework of congestion games is possible since every congestion game can be interpreted as a potential game (Rosenthal 1973).

**Experimental Setup** Our experimental setup involves a rooted directed acyclic graph (DAG), as illustrated in Figure 3. The graph comprises four paths connecting a source node  $s$  and a target node  $t$ . The length of each path is determined by the number of edges it contains, given that every edge in the graph is of unit distance. In addition, there are five players, each assigned the task of selecting a path and subject to an additional constraint on the amount of gas they are allowed to expend. The gas expenditure is proportional to the distance traveled, with one unit of gas allowing the traversal of one unit of distance. The congestion experienced on each path is influenced by the number of players selecting that particular route. Although the congestion functions for all paths

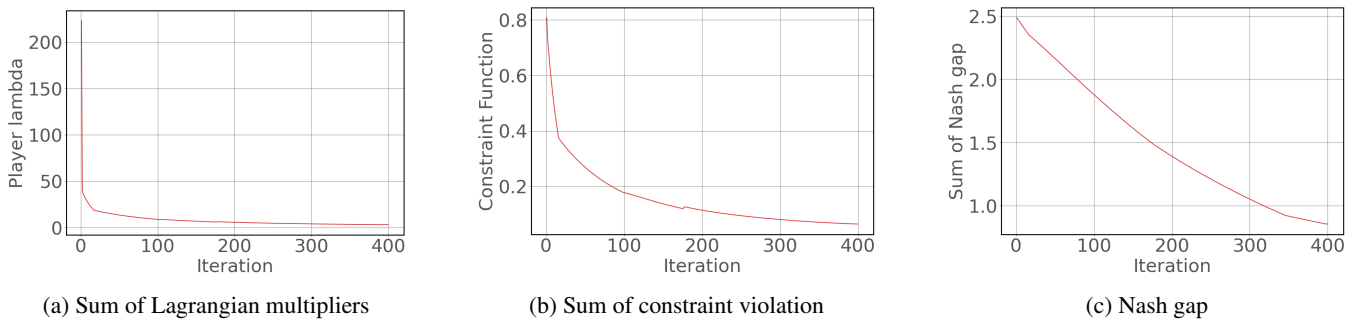


Figure 2: Collection of plots that are used to identify an approximate feasible approximate Nash Equilibrium.

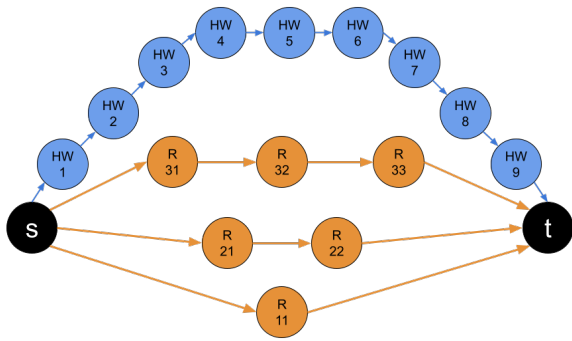


Figure 3: Players can pick out of four options: R1, R2, R3, HW by consuming {2, 3, 4, 10} from their gas allowance.

are linear, with the yellow paths sharing the same function, the highway’s function (blue path) has a smaller parameter, resulting in a lower cost experienced. This trade-off between congestion and gas consumption on the highway presents the important decision-making factor for the players.<sup>1</sup>

**Implementation & Results** Upon completing the iterations of Algorithm  $IGD_{\lambda}$ , we generate a spider chart to visually represent the final distributions of the players. Additionally, a table detailing the gas constraints for each player is presented alongside the spider chart. The agents select paths with lengths proportional to their gas constraints while simultaneously minimizing congestion.

**Evaluation** To assess the solution’s quality and the convergence speed of our algorithm, we employ several metrics of convergence: constraint violation, Lagrange multipliers, and Nash gap. Detailed calculations and information regarding the Nash Gap and hyperparameter selection can be found in the Appendix. In the final iteration, we anticipate the Nash gap to approach zero, constraint violation to be close to zero, and the Lagrange multipliers to have converged to a finite value. The accompanying plots are presented in Figure 2.

<sup>1</sup>The code is available at the GitHub repository: <https://github.com/steliostavroulakis/constrained-potential-games>

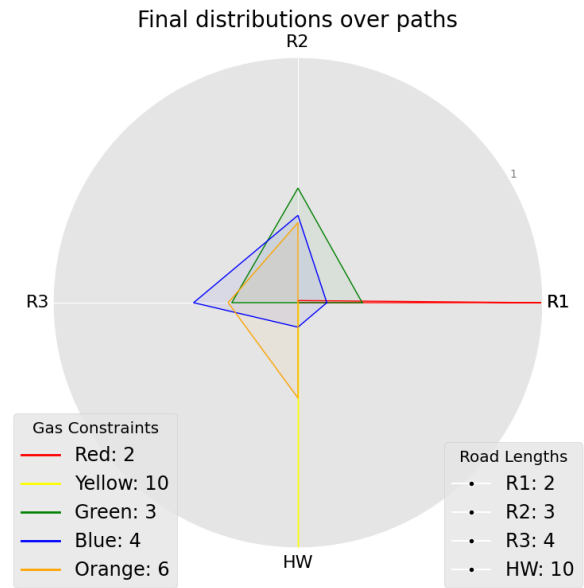


Figure 4: Spider chart capturing the final distributions over paths of all players. Given the initial gas constraints shown in the previous figure, this spider chart displays how players conform with constraints. The red player chooses his only available option, the yellow player freely chooses the highway to minimize congestion, and the other players distribute accordingly based on the relative ordering of their constraints.

## 6 Conclusions

In conclusion, this paper delves into decentralized computation of approximate Nash equilibria in constrained potential games, offering valuable insights and avenues for future research. We emphasize the importance of improving the convergence rates, based on our current findings. Furthermore, exploring different configurations of coupled constraints shows promise for advancing our understanding in this field. Additionally, extending the applicability of our results to scenarios beyond traditional potential games presents intriguing opportunities for further exploration.

## Acknowledgments

Ioannis Panageas would like to acknowledge startup grant from UCI. Part of this work was conducted while Nikolas, Fivos and Ioannis were visiting Archimedes Research Unit. This work has been partially supported by project MIS 5154714 of the National Recovery and Resilience Plan Greece 2.0 funded by the European Union under the NextGenerationEU Program.

## References

- Agarwal, A.; Kakade, S. M.; Lee, J. D.; and Mahajan, G. 2020. Optimality and approximation with policy gradient methods in markov decision processes. In *Conference on Learning Theory*, 64–66. PMLR.
- Alatur, P.; Ramponi, G.; He, N.; and Krause, A. 2023. Provably Learning Nash Policies in Constrained Markov Potential Games. *arXiv preprint arXiv:2306.07749*.
- Altman, E.; and Solan, E. 2009. Constrained Games: The Impact of the Attitude to Adversary’s Constraints. *IEEE Transactions on Automatic Control*, 54(10): 2435–2440.
- Ardagna, D.; Panicucci, B.; and Passacantando, M. 2011. A Game Theoretic Formulation of the Service Provisioning Problem in Cloud Systems. In *Proceedings of the 20th International Conference on World Wide Web, WWW ’11*, 177–186. New York, NY, USA: Association for Computing Machinery. ISBN 9781450306324.
- Arrow, K. J.; and Debreu, G. 1954. Existence of an equilibrium for a competitive economy. *Econometrica: Journal of the Econometric Society*, 265–290.
- Ba, Q.; and Pang, J.-S. 2022. Exact Penalization of Generalized Nash Equilibrium Problems. *Oper. Res.*, 70(3): 1448–1464.
- Beck, A. 2017. *First-order methods in optimization*. SIAM.
- Bertsekas, D. P. 1997. Nonlinear programming. *Journal of the Operational Research Society*, 48(3): 334–334.
- Boyd, S. P.; and Vandenberghe, L. 2004. *Convex optimization*. Cambridge university press.
- Breton, M.; Zaccour, G.; and Zahaf, M. 2006. A game-theoretic formulation of joint implementation of environmental projects. *European Journal of Operational Research*, 168(1): 221–239.
- Chen, Z.; Ma, S.; and Zhou, Y. 2022. Finding Correlated Equilibrium of Constrained Markov Game: A Primal-Dual Approach. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, volume 35, 25560–25572. Curran Associates, Inc.
- Daskalakis, C.; and Panageas, I. 2018. The limit points of (optimistic) gradient descent in min-max optimization. *Advances in neural information processing systems*, 31.
- Dreves, A.; von Heusinger, A.; Kanzow, C.; and Fukushima, M. 2013. A globalized Newton method for the computation of normalized Nash equilibria. *Journal of Global Optimization*, 56: 327–340.
- Facchinei, F.; Fischer, A.; and Piccialli, V. 2009. Generalized Nash equilibrium problems and Newton methods. *Mathematical Programming*, 117(1): 163–194.
- Facchinei, F.; and Kanzow, C. 2010. Penalty Methods for the Solution of Generalized Nash Equilibrium Problems. *SIAM Journal on Optimization*, 20(5): 2228–2253.
- Facchinei, F.; and Lampariello, L. 2011. Partial penalization for the solution of generalized Nash equilibrium problems. *Journal of Global Optimization*, 50(1): 39–57.
- Fischer, A.; Herrich, M.; Izmailov, A. F.; and Solodov, M. V. 2016. A globally convergent LP-Newton method. *SIAM Journal on Optimization*, 26(4): 2012–2033.
- Fischer, A.; Herrich, M.; and Schönefeld, K. 2014. Generalized nash equilibrium problems - Recent advances and challenges. *Pesquisa Operacional*, 34: 521–558.
- Fukushima, M. 2011. Restricted generalized Nash equilibria and controlled penalty algorithm. *Computational Management Science*, 8(3): 201–218.
- Ghadimi, S.; and Lan, G. 2016. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Mathematical Programming*, 156(1-2): 59–99.
- Hsiao, M.-T. T.; and Lazar, A. A. 1991. Optimal decentralized flow control of Markovian queueing networks with multiple controllers. *Performance Evaluation*, 13(3): 181–204.
- Izmailov, A. F.; and Solodov, M. V. 2014. On error bounds and Newton-type methods for generalized Nash equilibrium problems. *Computational Optimization and Applications*, 59(1-2): 201–218.
- (Jeff) Ban, X.; Dessouky, M.; Pang, J.-S.; and Fan, R. 2019. A general equilibrium model for transportation systems with e-hailing services and flow congestion. *Transportation Research Part B: Methodological*, 129: 273–304.
- Jordan, M. I.; Lin, T.; and Zampetakis, M. 2023. First-order algorithms for nonlinear generalized nash equilibrium problems. *Journal of Machine Learning Research*, 24(38): 1–46.
- Kanzow, C. 2016. On the multiplier-penalty-approach for quasi-variational inequalities. *Mathematical Programming*, 160: 33–63.
- Kanzow, C.; and Steck, D. 2016. Augmented Lagrangian Methods for the Solution of Generalized Nash Equilibrium Problems. *SIAM Journal on Optimization*, 26(4): 2034–2058.
- Kanzow, C.; and Steck, D. 2018. Augmented Lagrangian and Exact Penalty Methods for Quasi-Variational Inequalities. *Comput. Optim. Appl.*, 69(3): 801–824.
- Karush, W. 1939. *Minima of Functions of Several Variables with Inequalities as Side Conditions*. Master’s thesis, Department of Mathematics, University of Chicago, Chicago, IL, USA.
- Krawczyk, J. B.; and Uryasev, S. 2000. Relaxation algorithms to find Nash equilibria with economic applications. *Environmental Modeling & Assessment*, 5(1): 63–73.
- Kuhn, H. W.; and Tucker, A. W. 1951. Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, 1950*, 481–492. Berkeley and Los Angeles: University of California Press.

- Larsson, T.; and Patriksson, M. 1998. *Side Constrained Traffic Equilibrium Models—Traffic Management Through Link Tolls*, 125–151. Boston, MA: Springer US. ISBN 978-1-4615-5757-9.
- Leonardos, S.; Overman, W.; Panageas, I.; and Piliouras, G. 2021. Global convergence of multi-agent policy gradient in markov potential games. *arXiv preprint arXiv:2106.01969*.
- Lin, T.; Jin, C.; and Jordan, M. 2020. On gradient descent ascent for nonconvex-concave minimax problems. In *International Conference on Machine Learning*, 6083–6093. PMLR.
- Nash, J. 1951. Non-cooperative Games. *Annals of Mathematics*, 54(2): 286–295.
- Pang, J.-S.; and Fukushima, M. 2005. Quasi-variational inequalities, generalized Nash equilibria, and multi-leader-follower games. *Computational Management Science*, 2(1): 21–56.
- Pang, J.-S.; Scutari, G.; Facchinei, F.; and Wang, C. 2008. Distributed Power Allocation With Rate Constraints in Gaussian Parallel Interference Channels. *IEEE Transactions on Information Theory*, 54(8): 3471–3489.
- Rosen, J. B. 1965. Existence and Uniqueness of Equilibrium Points for Concave N-Person Games. *Econometrica*, 33(3): 520–534.
- Rosenthal, R. W. 1973. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2: 65–67.
- Sagratella, S. 2017. Algorithms for generalized potential games with mixed-integer variables. *Computational Optimization and Applications*, 68(3): 689–717.
- Scutari, G.; Barbarossa, S.; and Palomar, D. 2006. Potential Games: A Framework for Vector Power Control Problems With Coupled Constraints. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 4, IV–IV.
- Uryas’ev, S.; and Rubinstein, R. 1994. On relaxation algorithms in computation of noncooperative equilibria. *IEEE Transactions on Automatic Control*, 39(6): 1263–1267.
- von Heusinger, A.; and Kanzow, C. 2009. Relaxation Methods for Generalized Nash Equilibrium Problems with Inexact Line Search. *Journal of Optimization Theory and Applications*, 143(1): 159–183.
- von Heusinger, A.; Kanzow, C.; and Fukushima, M. 2012. Newton’s method for computing a normalized equilibrium in the generalized Nash game through fixed point formulation. *Mathematical Programming*, 132(1): 99–123.
- Zhu, Q. 2008a. A Lagrangian approach to constrained potential games: Theory and examples. In *2008 47th IEEE Conference on Decision and Control*, 2420–2425.
- Zhu, Q. 2008b. A Lagrangian approach to constrained potential games: Theory and examples. In *2008 47th IEEE Conference on Decision and Control*, 2420–2425. IEEE.