

A branch-and-prune algorithm for discrete Nash equilibrium problems

Stefan Schwarze* Oliver Stein#

March 21, 2022

Abstract

We present a branch-and-prune procedure for discrete Nash equilibrium problems with a convex description of each player's strategy set. The derived pruning criterion does not require player convexity, but only strict convexity of some player's objective function in a single variable. If satisfied, it prunes choices for this variable by stating activity of certain constraints. This results in a synchronous branching and pruning method. Numerical tests are performed on randomly generated instances with convex polyhedral strategy sets and convex quadratic as well as non-convex quadratic objective functions.

Key Words: Nash equilibrium problem, discrete game, branch and bound, branch and prune

Preliminary citation: Optimization Online, Preprint ID 2022-03-8836, 2022.

1 Introduction

The formulation of the Nash equilibrium for an n -person game by Nash in 1950 and 1951 was a landmark in the economic sciences and is still a key model in game theory [8,9]. In the setting of this game, finitely many players can choose their individual strategies independently, but their payoffs depend on the strategies of all players. In the absence of coalitions, each player aims to maximize her payoff given the other players' strategies. A situation in the game where no player has an incentive to unilaterally deviate from her strategy defines the famous Nash equilibrium, and finding such a situation is the so-called Nash equilibrium problem (NEP).

Over the last years, research gained interest in the numerical solution of NEPs, and there are a couple of algorithms tackling this issue. However, although integer optimization is applied in many fields and intensely studied, there are only few attempts to solve Nash

*Institute of Operations Research (IOR), Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, stefan.schwarze@kit.edu

#Institute of Operations Research (IOR), Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, stein@kit.edu

equilibrium problems with integer variables. Sagratella's publication [11] identifies this as an "important gap in literature". The latter paper proposes a branch-and-prune method to compute all solutions of NEPs with discrete strategy sets. Subsequently, the theory was extended to generalized NEPs with linear coupling constraints and mixed-integer variables [12]. We will rely on Sagratella's framework and propose a novel branch-and-prune approach for discrete NEPs which takes convexity of the strategy sets explicitly into account.

We introduce the problem and describe preliminary results in Section 2. In Section 3, we deliver a pruning criterion for NEPs with convex strategy sets. Section 4 provides an algorithmic application of the criterion for convex polyhedral strategy sets. In Section 5 we apply these findings numerically to discrete Nash equilibrium problems with convex polyhedral strategy sets and convex quadratic as well as non-convex quadratic objective functions. To the best of our knowledge, this is the first implemented and tested branch-and-prune procedure for this problem class. Finally, we wrap up our insights in Section 6.

2 Problem description and preliminary results

We study discrete Nash games with N players. In this setting, each player $\nu = 1, \dots, N$ aims to solve the optimization problem

$$Q^\nu(x^{-\nu}) : \min_{x^\nu} \theta_\nu(x^\nu, x^{-\nu}) \quad \text{s.t.} \quad x^\nu \in X_\nu.$$

The vector x^ν lies in \mathbb{R}^{n_ν} and represents all variables which are controlled by the ν -th player. The vector of all decision variables x is of dimension $n = \sum_{\nu=1}^N n_\nu$, and the notation $x := (x^\nu, x^{-\nu})$ emphasizes the ν -th player's decision variables, but does not reorder the entries of the vector $(x^1, \dots, x^\nu, \dots, x^N)$. The objective function $\theta_\nu : \Omega \rightarrow \mathbb{R}$ has the domain

$$\Omega := X_1 \times \dots \times X_N,$$

hence the player's objective function value depends on her own strategy as well as on the other players' strategies. The discrete feasible set

$$X_\nu := \{x^\nu \in \mathbb{Z}^{n_\nu} \mid g^\nu(x^\nu) \leq 0\}$$

is called the ν -th player's strategy set. It is defined by the function $g^\nu : \mathbb{R}^{n_\nu} \rightarrow \mathbb{R}^{m_\nu}$.

In this context, the *Nash equilibrium* is the most important and commonly used solution concept. A vector x^* is called Nash equilibrium of this game, if for each $\nu = 1, \dots, N$, the vector $x^{*,\nu}$ is an optimal point of $Q^\nu(x^{*,-\nu})$, i.e., $x^* \in \Omega$ and

$$\theta_\nu(x^*) = \theta_\nu(x^{*,\nu}, x^{*,-\nu}) \leq \theta_\nu(x^\nu, x^{*,-\nu}) \quad \forall x^\nu \in X_\nu$$

hold. The resulting Nash equilibrium problem may hence be formulated as

NEP : Find x^* such that $x^{*,\nu}$ is an optimal point of $Q^\nu(x^{*,-\nu})$ for all $\nu = 1, \dots, N$.

Our suggestion of a branch-and-prune approach for solving *NEP* will use its continuous relaxation \widehat{NEP} . There each player ν solves the continuous problem

$$\widehat{Q}^\nu(x^{-\nu}) : \min_{x^\nu} \theta_\nu(x^\nu, x^{-\nu}) \quad \text{s.t.} \quad x^\nu \in \widehat{X}_\nu$$

where the integrality condition is dropped in the strategy set

$$\widehat{X}_\nu := \{x^\nu \in \mathbb{R}^{n_\nu} \mid g^\nu(x^\nu) \leq 0\}.$$

The domain $\widehat{\Omega} := \widehat{X}_1 \times \dots \times \widehat{X}_N$ of the objective functions is defined analogously, and a vector \widehat{x}^* is a Nash equilibrium of \widehat{NEP} if $\widehat{x}^{*,\nu}$ solves $\widehat{Q}^\nu(\widehat{x}^{*,-\nu})$ for all $\nu = 1, \dots, N$.

From now on, we use the following assumption.

Assumption 2.1. *All entries of the function g^ν are convex for each player $\nu = 1, \dots, N$.*

Clearly, under Assumption 2.1, each player's relaxed strategy set \widehat{X}_ν is convex. If additionally each player's objective function θ_ν is convex with respect to x^ν , \widehat{NEP} is called *player convex*, which is a standard assumption for continuous NEPs. There are several possibilities to characterize and compute solutions of \widehat{NEP} under player convexity. For example, if $\widehat{\Omega}$ satisfies the Slater condition, a vector \widehat{x}^* is a Nash equilibrium of \widehat{NEP} if and only if $\widehat{x}^{*,\nu}$ is a Karush-Kuhn-Tucker (KKT) point of $\widehat{Q}^\nu(\widehat{x}^{*,-\nu})$ for each player (see [3, Prop. 1]). Other prominent solution techniques for \widehat{NEP} are the variational inequality (VI) and the Nikaido-Isoda (NI) approaches [4]. Unfortunately, none of these approaches carry over to the discrete problem NEP . For the KKT and VI approaches this is due to the missing convexity of the discrete strategy sets X_ν . The NI-function, on the other hand, may be defined for NEP , but it turns out to be structurally nonsmooth, nonconvex and discontinuous, and thus hard to treat algorithmically [12]. We mention that it follows from the findings in [6] that integer-feasible zero points of a "convexified" NI-function over $\widehat{\Omega}$ are solutions of NEP . However, we will not follow this approach, because we try to impose mild assumptions on the objective functions, which makes the computation of the NI-function's convex envelope rather impractical.

Instead, we will formulate an approach motivated by integer optimization techniques, where branch-and-bound algorithms are commonly used. Let us briefly recap the three key ingredients of branch-and-bound for integer optimization. Firstly, it exploits that it is easier to compute an optimal point of the continuous relaxation and that, if this point is integer, it also solves the integer optimization problem over this set. Secondly, if the obtained solution is not integer, it is removed by branching the feasible set. Thirdly, it is essential that the minimal value over some subset of the continuously relaxed feasible set serves as a lower bound for objective values of the integer feasible points in this subset. Thus one can discard subsets if the minimal value over their relaxation is larger than the objective value of the best known integer solution in the whole feasible set. This feature is called bounding.

The following example will help to illustrate how, if at all, these three steps carry over to the discrete problem NEP .

Example 2.2. *For two players, each one controlling a scalar variable, let us consider the NEP with objective functions*

$$\begin{aligned}\theta_1(x^1, x^2) &= \frac{3}{2}(x^1)^2 - 8x^1 + 4x^1x^2, \\ \theta_2(x^1, x^2) &= \frac{3}{2}(x^2)^2 - 6x^2 + 4x^1x^2\end{aligned}$$

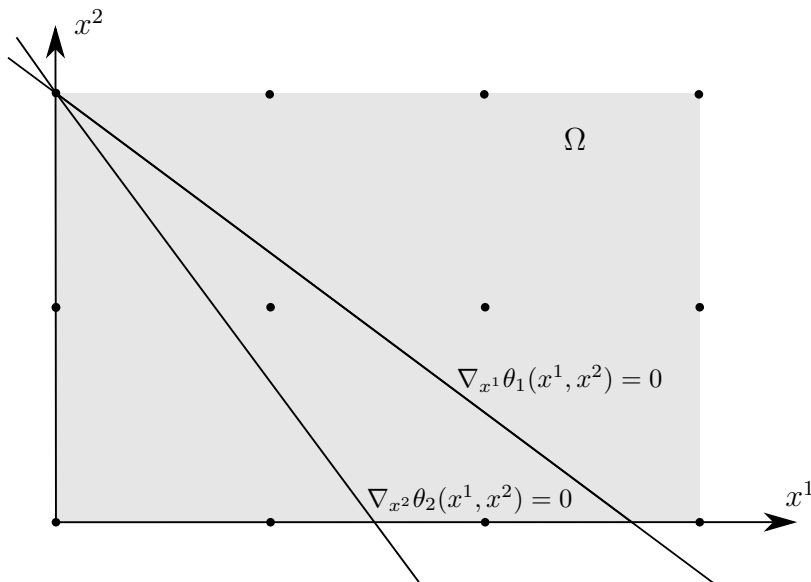


Figure 1: Feasible set of Example 2.2.

and strategy sets

$$\begin{aligned} X_1 &= \{x^1 \in \mathbb{Z} \mid 0 \leq x^1 \leq 3\}, \\ X_2 &= \{x^2 \in \mathbb{Z} \mid 0 \leq x^2 \leq 2\} \end{aligned}$$

as well as their continuous relaxations

$$\begin{aligned} \widehat{X}_1 &= \{x^1 \in \mathbb{R} \mid 0 \leq x^1 \leq 3\}, \\ \widehat{X}_2 &= \{x^2 \in \mathbb{R} \mid 0 \leq x^2 \leq 2\}. \end{aligned}$$

Since for any fixed $x^{-\nu}$, the loss function of each player ν is convex quadratic, the relaxed problem \widehat{NEP} is player convex.

For any $x^2 \in \widehat{X}_2$ the unconstrained minimal point of $\theta_1(\cdot, x^2)$ is given by $\nabla_{x^1} \theta_1(x^1, x^2) = 0$, lies in \widehat{X}_1 and is, thus, the best response of player 1 to x^2 . On the other hand, only for $x^1 \in [0, 3/2]$ the unconstrained minimal point of $\theta_2(x^1, \cdot)$, characterized by $\nabla_{x^2} \theta_2(x^1, x^2) = 0$, lies in \widehat{X}_2 , but for any $x^1 \in [3/2, 3]$ the boundary point $x^2 = 0$ is the best response to x^1 . Figure 1 illustrates that, thus, exactly the two points $(0, 2)$ and $(8/3, 0)$ solve \widehat{NEP} .

In contrast, the discrete problem NEP possesses exactly the three solutions $(0, 2)$, $(1, 1)$ and $(3, 0)$. In particular, although the point $(1, 1)$ lies close to the solution $(0, 2)$ of NEP as well as of \widehat{NEP} , it is an equilibrium. Moreover, as opposed to the point $(3, 0)$, it may not be obtained by rounding the entries of any of the solutions of \widehat{NEP} .

Regarding the first of the three above branch-and-bound ingredients, in Example 2.2 the continuously relaxed problem \widehat{NEP} is easy to solve, and the single discrete solution of \widehat{NEP} also solves NEP . Also in general, the KKT, VI or NI methods can be employed to solve a player convex problem \widehat{NEP} with differentiable defining functions, and the following result from [11, Prop. 2.1] guarantees that discrete solutions of the continuously relaxed problem solve the original discrete problem.

Proposition 2.3. *Any solution $x^* \in \mathbb{Z}^n$ of \widehat{NEP} also solves NEP .*

Note that this result also holds without player convexity, but that in this case \widehat{NEP} may not be easy to solve, even under Assumption 2.1. In Section 4 we explain how we deal with non-convex objective functions. This means that with regard to the first branch-and-bound ingredient we are in an analogous situation as in integer optimization. Concerning the second ingredient, we can also branch the strategy sets if the obtained solution is not integer, so that this situation is analogous as well.

In contrast, the bounding step as the third ingredient poses some difficulties. Firstly and most obviously, there are multiple objective functions. Equilibrium points are required to be minimal for each player's objective function with respect to the other players' decisions. However, we are interested in a single criterion telling us whether there may exist Nash equilibria on a given subset of the strategy space. More specifically, the bounding idea relies on some function p on the joint strategy set Ω whose minimal points coincide with the solutions of NEP . For the continuous problem \widehat{NEP} such functions can be obtained by the VI and NI approaches [2,10] but, as mentioned above, the latter are impossible or hard to apply in the discrete framework.

Under the additional assumption of NEP being a potential game [7] there exists a potential function $p : \mathbb{R}^n \rightarrow \mathbb{R}$ with

$$\theta_\nu(x^\nu, x^{-\nu}) - \theta_\nu(y^\nu, x^{-\nu}) = p(x^\nu, x^{-\nu}) - p(y^\nu, x^{-\nu}) \quad \forall x^\nu, y^\nu \in X_\nu$$

for all $\nu = 1, \dots, N$ and all $x^{-\nu} \in X_{-\nu}$. It is straightforward to show that then any optimal point of the integer program

$$\mathcal{P} : \min_x p(x) \quad \text{s.t.} \quad x \in \Omega$$

solves NEP . However, in general not all solutions of NEP are optimal for \mathcal{P} , as required for a bounding procedure relying on p . In fact, Example 2.2 provides a potential game with potential function

$$p(x^1, x^2) = \frac{3}{2}(x^1)^2 - 8x^1 + \frac{3}{2}(x^2)^2 - 6x^2 + 4x^1x^2,$$

but the potential values

$$p((0, 2)) = -6, \quad p((1, 1)) = -7, \quad p((3, 0)) = -10.5$$

of the three solutions of NEP are not identical. In any case, potential games form only a small subclass of NEPs, and their restrictive assumptions cover, e.g., cases where all players unconsciously minimize the same objective function. We, on the other hand, aim to handle a more general setting, only satisfying Assumption 2.1.

Since, if the solution of \widehat{NEP} does not happen to be integer, we do not seem to be able to draw any conclusions for discarding subsets of Ω by a *bounding* procedure, we will instead follow the *branch-and-prune* approach from [11,12]. There, relations between equilibria of NEP and \widehat{NEP} are exploited algorithmically. Regarding such relations, Example 2.2 illustrates that NEP may possess more solutions than \widehat{NEP} and that not every solution

of NEP may be obtained by rounding the fractional components of a solution of \widehat{NEP} . There are also examples where NEP possesses less solutions than \widehat{NEP} . In particular, the solvability of \widehat{NEP} does not entail the solvability of NEP (see [11, Ex. 2]). Additional requirements for the latter are given in [11, Cor. 4.4].

3 Theoretical foundation

The purpose of this section is to define a pruning criterion for discrete NEPs under Assumption 2.1. Moreover, for each player $\nu = 1, \dots, N$ we assume g^ν to be continuously differentiable and θ_ν to be twice continuously differentiable.

The theorem we present in this section employs a similar logic as Proposition 3.1 from [11]. In fact, we prove in Appendix A that it includes this proposition. It provides a set of verifiable conditions under which we are able to prune choices for values of single variables from some player's strategy set. We shall also motivate the underlying geometrical concept.

Our approach uses local approximations of the continuously relaxed problem \widehat{NEP} to infer properties of the discrete problem NEP . As opposed to [11,12] we use arbitrary continuous strategies, rather than only solutions of \widehat{NEP} , to obtain these approximations. This enables us to deal with non-convexities in the objective functions. For the approximations we employ the concept of the (outer) linearization cone

$$L_{\leq}(\bar{x}^\nu, \widehat{X}_\nu) := \left\{ d \in \mathbb{R}^{n_\nu} \mid \langle \nabla g_i^\nu(\bar{x}^\nu), d \rangle \leq 0, i \in I_0(\bar{x}^\nu, \widehat{X}_\nu) \right\}$$

of player ν 's continuously relaxed strategy set \widehat{X}_ν at the strategy $\bar{x}^\nu \in \widehat{X}_\nu$, where

$$I_0(\bar{x}^\nu, \widehat{X}_\nu) := \{i \in \{1, \dots, m_\nu\} \mid g_i^\nu(\bar{x}^\nu) = 0\}$$

denotes the active index set. Under the convexity property of the strategy constraints from Assumption 2.1 it is straightforward to prove that any linearization cone for player ν provides an outer approximation of her relaxed strategy set in the following sense.

Lemma 3.1. *Let g_i^ν be convex for $i = 1, \dots, m_\nu$ and let $\bar{x}^\nu \in \widehat{X}_\nu$. Then we have*

$$\widehat{X}_\nu \subseteq \bar{x}^\nu + L_{\leq}(\bar{x}^\nu, \widehat{X}_\nu).$$

Theorem 3.2. *Let Assumption 2.1 hold, let $\bar{x} \in \widehat{\Omega}$ and, for an arbitrary player ν , let there exist an index i such that θ_ν is strictly convex with respect to x_i^ν . Then the following two statements hold:*

(i) *Let $F_i^\nu = \nabla_{x_i^\nu} \theta_\nu$ be convex, let $F_i^\nu(\bar{x}) \geq 0$ and for each player $\mu = 1, \dots, N$ let*

$$\langle \nabla_{x^\mu} F_i^\nu(\bar{x}), d^\mu \rangle \geq 0 \quad \forall d^\mu \in L_{\leq}(\bar{x}^\mu, \widehat{X}_\mu). \quad (1)$$

Then any strategy $\tilde{x} \in \Omega$ for which \check{x} , defined by

$$\check{x}_j^\mu = \tilde{x}_j^\mu \text{ for all } (\mu, j) \neq (\nu, i), \text{ and } \check{x}_i^\nu = \tilde{x}_i^\nu - 1,$$

is also feasible cannot be a solution of NEP.

(ii) Let $F_i^\nu = \nabla_{x_i^\nu} \theta_\nu$ be concave, let $F_i^\nu(\bar{x}) \leq 0$ and for each player $\mu = 1, \dots, N$ let

$$\langle -\nabla_{x^\mu} F_i^\nu(\bar{x}), d^\mu \rangle \geq 0 \quad \forall d^\mu \in L_{\leq}(\bar{x}^\mu, \hat{X}_\mu). \quad (2)$$

Then any strategy $\tilde{x} \in \Omega$ for which \hat{x} , defined by

$$\hat{x}_j^\mu = \tilde{x}_j^\mu \text{ for all } (\mu, j) \neq (\nu, i), \text{ and } \hat{x}_i^\nu = \tilde{x}_i^\nu + 1,$$

is also feasible cannot be a solution of NEP.

Proof. In order to show that \tilde{x} is not a Nash equilibrium, we will show that player ν can choose a strictly better strategy.

On the one hand, if (i) holds and if we can show

$$\theta_\nu(\tilde{x}) = \theta_\nu(\tilde{x}^\nu, \tilde{x}^{-\nu}) < \theta_\nu(\tilde{x}^\nu, \tilde{x}^{-\nu}) = \theta_\nu(\tilde{x}) \quad (3)$$

the assertion follows by the feasibility of \tilde{x} for the discrete NEP. The strict inequality

$$\theta_\nu(\tilde{x}) + \langle \nabla_{x_i^\nu} \theta_\nu(\tilde{x}), \tilde{x}_i^\nu - \check{x}_i^\nu \rangle < \theta_\nu(\tilde{x}) \quad (4)$$

holds because of the strict convexity of θ_ν in the component x_i^ν , being the only value in which \check{x} and \tilde{x} differ. Hence, (3) follows, when

$$\langle \nabla_{x_i^\nu} \theta_\nu(\tilde{x}), \tilde{x}_i^\nu - \check{x}_i^\nu \rangle = F_i^\nu(\tilde{x}) \geq F_i^\nu(\bar{x}) + \langle \nabla_x F_i^\nu(\bar{x}), \tilde{x} - \bar{x} \rangle \geq 0$$

holds. Firstly, the equation comes from the defined notation and $\check{x}_i^\nu = \tilde{x}_i^\nu - 1$. Secondly, the left inequality follows from convexity of F_i^ν . Thirdly, the non-negativity comes from

- $F_i^\nu(\bar{x}) \geq 0$ by precondition,
- $\langle \nabla_x F_i^\nu(\bar{x}), \tilde{x} - \bar{x} \rangle = \sum_{\mu=1}^N \langle \nabla_{x^\mu} F_i^\nu(\bar{x}), \tilde{x}^\mu - \bar{x}^\mu \rangle \geq 0$, because from Lemma 3.1 follows $\tilde{x}^\mu - \bar{x}^\mu \in L_{\leq}(\bar{x}^\mu, \hat{X}_\mu)$ so that every summand is non-negative by (1).

On the other hand, if (ii) holds, the equations (3) and (4) can be stated for \hat{x} instead of \tilde{x} as well with all requirements fulfilled. It remains to show that the chain of inequalities

$$\langle \nabla_{x_i^\nu} \theta_\nu(\hat{x}), \hat{x}_i^\nu - \tilde{x}_i^\nu \rangle = -F_i^\nu(\hat{x}) \geq \langle -\nabla_x F_i^\nu(\bar{x}), \hat{x} - \bar{x} \rangle - F_i^\nu(\bar{x}) \geq 0$$

also holds. Firstly, the equation comes again from notation and $\hat{x}_i^\nu = \tilde{x}_i^\nu + 1$. Secondly, the left inequality is valid due to the concavity of F_i^ν , and the non-negativity comes from

- $-F_i^\nu(\bar{x}) \geq 0$ by precondition,
- $\langle -\nabla_x F_i^\nu(\bar{x}), \hat{x} - \bar{x} \rangle = \sum_{\mu=1}^N \langle -\nabla_{x^\mu} F_i^\nu(\bar{x}), \hat{x}^\mu - \bar{x}^\mu \rangle \geq 0$, because from Lemma 3.1 follows $\hat{x}^\mu - \bar{x}^\mu \in L_{\leq}(\bar{x}^\mu, \hat{X}_\mu)$ so that every summand is non-negative by (2).

□

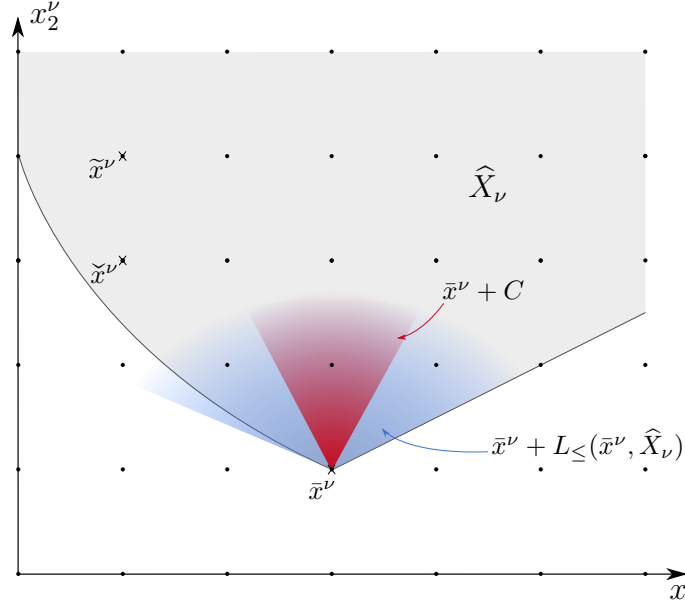


Figure 2: Pruning of \tilde{x} , when Theorem 3.2.(i) is fulfilled for $i = 2$. In particular $\nabla_{x^\nu} F_2^\nu(\bar{x})$ needs to lie in the dual cone C of $L_{\leq}(\bar{x}^\nu, \hat{X}_\nu)$.

To verbalize the statement of Theorem 3.2, we use the point \bar{x} to construct outer approximations of all players' complete strategy sets. This actually results in the outer approximation

$$\hat{\Omega} = \prod_{\nu=1}^N \hat{X}_\nu \subseteq \prod_{\nu=1}^N \left(\bar{x}^\nu + L_{\leq}(\bar{x}^\nu, \hat{X}_\nu) \right) = \bar{x} + L_{\leq}(\bar{x}, \hat{\Omega})$$

of $\hat{\Omega}$. If, on this whole set $\bar{x} + L_{\leq}(\bar{x}, \hat{\Omega})$, some player's variable x_i^ν has a favorable impact on the objective function θ_ν when it is increased or decreased without the new point becoming infeasible, then this player can deviate and realize this positive impact, which is impossible in a Nash equilibrium. In other words, under the given assumptions in a Nash equilibrium \tilde{x} the constructed deviation must result in an infeasible point. Figure 2 shows the two dimensional strategy set X_ν of a discrete N player game. Assume that for \bar{x} and $i = 2$ all requirements of Theorem 3.2.(i) hold. Then there is always a positive impact in the ν -th player's objective function, when she sets x_2^ν to a lower value. As a result, e.g. \tilde{x} cannot be a Nash equilibrium, because \tilde{x}^ν is feasible and a better answer for player ν . In X_ν , the set of possible best answers and thus the candidates for solutions of NEP shrinks to the pairs of "minimum feasible" x_2^ν -values for any given x_1^ν . Roughly speaking, only integer points for which at least one constraint is "active" in the sense that x_2^ν cannot be set to a lower value without changing other components of x^ν can be Nash equilibria. This criterion will enable us to reduce the search space significantly in Sections 4 and 5.

For the algorithmic exploitation of Theorem 3.2 we define linear optimization problems to check if (1) and (2) are satisfied. The statement (1) is clearly valid if and only if the optimization problem

$$F_{(1)} : \min_{d^\mu} \langle \nabla_{x^\mu} F_i^\nu(\bar{x}), d^\mu \rangle \quad s.t. \quad d^\mu \in L_{\leq}(\bar{x}^\mu, \hat{X}_\mu)$$

has a non-negative optimal value $v_{F(1)} \geq 0$. In the same way, (2) holds if and only if

$$F_{(2)} : \min_{d^\mu} \langle -\nabla_{x^\mu} F_i^\nu(\bar{x}), d^\mu \rangle \quad s.t. \quad d^\mu \in L_{\leq}(\bar{x}^\mu, \widehat{X}_\mu)$$

has a non-negative optimal value $v_{F(2)} \geq 0$. By definition of the linearization cone, $d^\mu = 0$ is always feasible for $F_{(1)}$ and $F_{(2)}$, so that the above optimal values are actually zero. If, on the other hand, there exists any direction d^μ with a negative objective value, the problems are unbounded, because the feasible set is a cone. Therefore, we only have to check if these linear optimization problems are bounded in order to verify the statements. We remark that (1) and (2) require that the vectors $\nabla_{x^\mu} F_i^\nu(\bar{x})$ and $-\nabla_{x^\mu} F_i^\nu(\bar{x})$, respectively, lie in the dual cone of $L_{\leq}(\bar{x}^\mu, \widehat{X}_\mu)$. In view of possibly non-unique cone coefficients in the absence of an appropriate constraint qualification it may, however, be algorithmically challenging to determine these cone coefficients explicitly, so that we rather work with the above optimization formulation.

We emphasize that we need the strict convexity of the objective function θ_ν in the component (ν, i) in order to apply Theorem 3.2. The (strict) convexity in *single* variables does not require convexity in all of player ν 's variables x^ν , as defined in player convexity. However, the additional assumption might be helpful in the sense that it increases the likelihood of finding (ν, i) -components in which θ_ν is strictly convex.

4 Algorithmic application

In this section, we define a branch-and-prune procedure for discrete NEPs by employing the pruning criterion from the previous section. This branching method is a modification of [11, Algorithm 1]. Resulting from the findings in Section 3 we define a method with synchronous branching and pruning.

4.1 Branching method

Algorithm 1 shows the high-level approach for discrete NEPs with convexly constrained, bounded strategy sets. In each iteration of the while-loop a joint strategy set $Y \subseteq \Omega$ is taken from the list \mathcal{L} . For the continuous relaxation \widehat{Y} of this set a feasible point \bar{x} is computed. Here, the feasibility of \bar{x} is a minimum requirement but, given the target of finding solutions of *NEP* as quickly as possible, in view of Proposition 2.3 computing a solution \bar{x} of the continuously relaxed problem \widehat{NEP} may be advantageous, depending on the effort for such a computation.

Afterwards, the first pruning and simultaneous branching can be started in line 6, resulting in the collection of sets \mathcal{B} which contain all equilibria of *NEP* in the set Y . Additionally, the sets in \mathcal{B} are pairwise disjoint. This is crucial in branching techniques to avoid that a point needs to be processed multiple times. Further, the partition is performed in a way that ensures $\bar{x} \in B_1$ if $\bar{x} \in \bigcup_{B \in \mathcal{B}} B$. The reason is that the pruning procedure could exclude \bar{x} . Otherwise we further process the point, so that it is useful to know which of the subsets contains it. Algorithm 2 in the next subsection presents such a procedure for convex polyhedral strategy sets.

Starting in line 7, the second branching process depends on whether $\bar{x} \in \mathbb{Z}^n$ or not. If so, the vector is potentially an equilibrium of the discrete problem *NEP* and can, after verification, be appended to E , the list of all Nash equilibria. The equilibrium property can be verified by checking if \bar{x}^ν is a solution of $Q^\nu(\bar{x}^{-\nu})$ for all $\nu = 1, \dots, N$. More specifically, we need to solve these N integer (non-)convex programs in the verification step (line 8). The appearing integer (non-)convex problems can be solved with techniques from mixed-integer (non-)convex optimization (see e.g. [1]). The efficient implementation of this step of course depends on the state-of-the-art of available solvers. After knowing whether \bar{x} is a solution, we can release it and search in the remaining feasible set for Nash equilibrium points. By Algorithm 3 ([11] and Appendix B), we obtain a partition of sets \mathcal{B}^+ which cover all other possible equilibria. Any integer point $\tilde{x} \in B_1 \cap \mathbb{Z}^n$ other than \bar{x} is in one of the sets from \mathcal{B}^+ . Additionally, these sets are again pairwise disjoint.

If otherwise $\bar{x} \notin \mathbb{Z}^n$, the branching step resembles the one common in integer optimization. One fractional component in \bar{x} is selected and two sets are added to the list \mathcal{L} . In the first one, the value of this component is bounded to be greater or equal the nearest larger integer. In the second one, it is bounded to be less or equal the nearest smaller integer.

When the strategy sets X_1, \dots, X_N are bounded, the termination of Algorithm 1 is ensured because there are finitely many integer strategies. This property can be established by setting finite upper and lower bounds for each variable.

Algorithm 1: Branching Method.

Input: Discrete Nash equilibrium problem NEP with bounded strategy sets defined by convex functions

Output: Solution set E of NEP

```
1 Initialize list of strategy subsets  $\mathcal{L} := \{\Omega\}$ 
2 initialize list of equilibria  $E := \{\}$ 
3 while  $\mathcal{L} \neq \{\}$  do
4   take a strategy set  $Y$  from  $\mathcal{L}$ 
5   compute a feasible strategy  $\bar{x} \in \hat{Y}$ 
6   apply pruning procedure, with  $Y$  and  $\bar{x}$ , to obtain a list of  $k$  sets
    $\mathcal{B} = \{B_1, \dots, B_k\}$ , then take  $B_1$  from  $\mathcal{B}$ 
7   if  $\bar{x} \in \mathbb{Z}^n$  then
8     if  $\bar{x}$  is a solution of  $NEP$  then put  $\bar{x}$  in  $E$ 
9     Obtain the list  $\mathcal{B}^+$  from Algorithm 3 with  $B_1$  and  $\bar{x}$  as Input
10     $\mathcal{B} \leftarrow \{B_2, \dots, B_k\} \cup \mathcal{B}^+$ 
11  else
12    find an index  $i \in \{1, \dots, n\}$  with  $\bar{x}_i \notin \mathbb{Z}$ 
13    if  $B_1 \cap \{x \in \mathbb{R}^n \mid x_i \geq \lceil \bar{x}_i \rceil\} \neq \emptyset$  then
14      put  $B_1 \cap \{x \in \mathbb{R}^n \mid x_i \geq \lceil \bar{x}_i \rceil\}$  in  $\mathcal{B}$ 
15    if  $B_1 \cap \{x \in \mathbb{R}^n \mid x_i \leq \lfloor \bar{x}_i \rfloor\} \neq \emptyset$  then
16      put  $B_1 \cap \{x \in \mathbb{R}^n \mid x_i \leq \lfloor \bar{x}_i \rfloor\}$  in  $\mathcal{B}$ 
17   $\mathcal{L} \leftarrow \mathcal{L} \cup \mathcal{B}$ 
```

4.2 Pruning procedure for convex polyhedral strategy sets

We now define a pruning procedure for discrete Nash games, where every player's strategy set X_ν can be characterized by linear inequalities:

$$X_\nu := \{x^\nu \in \mathbb{Z}^{n_\nu} \mid B^\nu x^\nu \leq b^\nu, l^\nu \leq x^\nu \leq u^\nu\}. \quad (5)$$

With each player having m_ν inequality constraints, B^ν is an integer valued $(m_\nu \times n_\nu)$ -matrix and $b^\nu \in \mathbb{R}^{m_\nu}$ a vector. We will refer to the k -th row of B^ν as $B_{k\star}^\nu$. The player's decision vector x^ν has explicitly defined lower and upper bounds $l^\nu \in \mathbb{Z}^{n_\nu}$ and $u^\nu \in \mathbb{Z}^{n_\nu}$, respectively. We call a discrete strategy set X_ν convex polyhedral, if the continuous relaxation of this strategy set \widehat{X}_ν is convex and polyhedral.

Previously, we determined conditions under which in an equilibrium of *NEP* it is not possible to increase or decrease the value of a variable to the next integer and remain feasible. Accordingly, at least one inequality (or the box restriction) must be active in a way that the next integer value in one direction is not feasible anymore. We formalize this kind of activity for a linear constraint.

Definition 4.1. *For a feasible point $\tilde{x} \in \Omega$ the inequality k is called*

- $(\nu, i)^-$ -active if for \check{x} , defined as $\check{x}_j^\mu = \tilde{x}_j^\mu$ for $(\mu, j) \neq (\nu, i)$ and $\check{x}_i^\nu = \tilde{x}_i^\nu - 1$, we have $B_{k\star}^\nu \check{x}^\nu > b_k^\nu$,
- $(\nu, i)^+$ -active if for \hat{x} , defined as $\hat{x}_j^\mu = \tilde{x}_j^\mu$ for $(\mu, j) \neq (\nu, i)$ and $\hat{x}_i^\nu = \tilde{x}_i^\nu + 1$, we have $B_{k\star}^\nu \hat{x}^\nu > b_k^\nu$.

We now investigate conditions that necessarily hold, when inequality k is $(\nu, i)^-$ - or $(\nu, i)^+$ -active. With these conditions we will be able to perform pruning steps. To begin with, by Definition 4.1 an inequality k is not $(\nu, i)^-$ -active for \tilde{x} if

$$b_k^\nu \geq B_{k\star}^\nu \tilde{x}^\nu = B_{k\star}^\nu \tilde{x}^\nu + B_{ki}^\nu \underbrace{(\tilde{x}_i^\nu - \tilde{x}_i^\nu)}_{=-1}$$

holds. Consequently we obtain

$$B_{k\star}^\nu \tilde{x}^\nu > b_k^\nu + B_{ki}^\nu \quad (6)$$

as necessary condition for inequality k being $(\nu, i)^-$ -active. Obviously this condition can only be satisfied if $B_{ki}^\nu < 0$ holds due to feasibility of \tilde{x} . Similarly, an inequality k is not $(\nu, i)^+$ -active for \tilde{x} if

$$b_k^\nu \geq B_{k\star}^\nu \tilde{x}^\nu = B_{k\star}^\nu \tilde{x}^\nu + B_{ki}^\nu \underbrace{(\tilde{x}_i^\nu - \tilde{x}_i^\nu)}_{=1}$$

holds. According to this we receive

$$B_{k\star}^\nu \tilde{x}^\nu > b_k^\nu - B_{ki}^\nu \quad (7)$$

as necessary condition for inequality k being $(\nu, i)^+$ -active, which can only hold when we have $B_{ki}^\nu > 0$. Note that we will use non-strict inequalities in the algorithmic implementation of (6) and (7) as we target adding inequalities to our model. This argumentation results in Corollary 4.2.

Corollary 4.2. *Let \tilde{x} be a solution of a problem NEP with strategy sets as defined in (5). Then the following two statements hold:*

- (i) *Suppose all requirements of Theorem 3.2.(i) hold for some $\bar{x} \in \widehat{\Omega}$ and an index pair (ν, i) . Then $\tilde{x}_i^\nu = l_i^\nu$ holds or there exists at least one inequality k with (6) and $B_{ki}^\nu < 0$.*
- (ii) *Suppose all requirements of Theorem 3.2.(ii) hold for some $\bar{x} \in \widehat{\Omega}$ and an index pair (ν, i) . Then $\tilde{x}_i^\nu = u_i^\nu$ holds or there exists at least one inequality k with (7) and $B_{ki}^\nu > 0$.*

Now we can state Algorithm 2. This procedure can be applied in an arbitrary point $\bar{x} \in \widehat{\Omega}$ in order to reduce the search space. Two outer for-loops, starting in line 2 and 3, iterate through all variables of the game. For each variable, the requirements of Theorem 3.2 are checked. If case (i) or (ii) is applicable, we perform a partition of the set according to Corollary 4.2.

Suppose that for (ν, i) the if-statement in line 4 is true. Then the first statement of Corollary 4.2 holds. Consequently in any Nash equilibrium x , x_i^ν is at its lower bound l_i^ν or at least one inequality from the index set $J^{\nu, i}$ must be $(\nu, i)^-$ -active. Integer points for which none of this holds are pruned in lines 5-12 by introducing new inequalities and splitting the set(s) up. In line 6, there is an inner for-loop which ensures that the subdivision is done for every set in the list \mathcal{B} . At first, \mathcal{B} only contains Y , but as soon as the if-statements hold true for more than one index pair, the branching step is performed for all sets from the previous subdivisions. Thus for every set from the current list \mathcal{B} the partition is added to a new list \mathcal{C} , which replaces \mathcal{B} afterwards. The subdivision itself for each set (lines 7-11) follows a similar logic like Algorithm 3. In line 7, all points for which x_i^ν is at its lower bound are added to \mathcal{C} and line 8 ensures that the next sets are disjoint. Then in lines 9-11, for all inequalities with $B_{ki}^\nu < 0$ the points for which firstly (6) holds and secondly are not contained in previous sets are added to \mathcal{C} . The latter is done in every iteration by stating the negation of (6) for all sets which will be added to \mathcal{C} successively in this for-loop.

Stating the negation of inequality (6) straightforwardly like

$$B_{k\star}^\nu \tilde{x}^\nu \not\geq b_k^\nu + B_{ki}^\nu \Leftrightarrow B_{k\star}^\nu \tilde{x}^\nu < b_k^\nu + B_{ki}^\nu$$

is not directly possible. The reason is that modeling strict inequalities can cause unsolvability of optimization problems, but with a basic result from number theory we can avoid this formulation: The scalar product of two integer valued vectors is integer. Thus $B_{k\star}^\nu x^\nu \geq \lceil b_k + B_{ki}^\nu \rceil$ holds if (6) is valid, and $B_{k\star}^\nu x^\nu \leq \lceil b_k + B_{ki}^\nu \rceil - 1$ is its negation. These inequalities form a split disjunction.

In lines 13-21 the analogous approach is implemented for the case when the requirements of Theorem 3.2.(ii) hold and Corollary 4.2.(ii) can be applied. We illustrate in Example 4.3 how the procedure works in detail.

Example 4.3. *Figure 3 depicts how the two inner for-loops in Algorithm 2 partition one player's strategy set into three sets C_1, C_2 and C_3 . The illustration shows the first player's strategy (sub-)set for an arbitrary game with two variables. The gray area is the*

Algorithm 2: Pruning procedure for convex polyhedral *NEP*

Input: Joint strategy set $Y \subseteq \Omega$ of *NEP*, strategy $\bar{x} \in \widehat{Y} = \widehat{Y}_1 \times \dots \times \widehat{Y}_N$

Output: A list of sets \mathcal{B} , such that $Y \setminus \{\bigcup_{B \in \mathcal{B}} B\}$ does not contain any solution of

NEP

```

1 Initialize  $\mathcal{B} := \{Y\}$ 
2 for  $\nu = 1$  to  $N$  do
3   for  $i = 1$  to  $n_\nu$  do
4     if  $\theta_i^\nu$  is strictly convex in  $x_i^\nu$ ,  $F_i^\nu$  is convex,  $F_i^\nu(\bar{x}) \geq 0$  and (1) holds for
       each player then
5       initialize  $\mathcal{C} := \{\}$ 
6       for  $B \in \mathcal{B}$  do
7         put the set  $B \cap \{x \in \mathbb{R}^n \mid x_i^\nu \leq l_i^\nu\}$  in  $\mathcal{C}$ , if non-empty
8          $B \leftarrow B \cap \{x \in \mathbb{R}^n \mid x_i^\nu \geq l_i^\nu + 1\}$ 
9         for  $j \in J^{\nu,i} := \{j \in \mathbb{N} \mid B_{ji}^\nu < 0\}$  do
10          put the set  $B \cap \{x \in \mathbb{R}^n \mid B_{j\star}^\nu x^\nu \geq \lceil b_j + B_{ji}^\nu \rceil\}$  in  $\mathcal{C}$ , if non-empty
11           $B \leftarrow B \cap \{x \in \mathbb{R}^n \mid B_{j\star}^\nu x^\nu \leq \lceil b_j + B_{ji}^\nu \rceil - 1\}$ 
12        $\mathcal{B} \leftarrow \mathcal{C}$ 
13     if  $\theta_i^\nu$  is strictly convex in  $x_i^\nu$ ,  $F_i^\nu$  is concave,  $F_i^\nu(\bar{x}) \leq 0$  and (2) holds for
       each player then
14       initialize  $\mathcal{C} := \{\}$ 
15       for  $B \in \mathcal{B}$  do
16         put the set  $B \cap \{x \in \mathbb{R}^n \mid x_i^\nu \geq u_i^\nu\}$  in  $\mathcal{C}$ , if non-empty
17          $B \leftarrow B \cap \{x \in \mathbb{R}^n \mid x_i^\nu \leq u_i^\nu - 1\}$ 
18         for  $k \in K^{\nu,i} := \{k \in \mathbb{N} \mid B_{ki}^\nu > 0\}$  do
19          put the set  $B \cap \{x \in \mathbb{R}^n \mid B_{k\star}^\nu x^\nu \geq \lceil b_k - B_{ki}^\nu \rceil\}$  in  $\mathcal{C}$ , if
           non-empty
20           $B \leftarrow B \cap \{x \in \mathbb{R}^n \mid B_{k\star}^\nu x^\nu \leq \lceil b_k - B_{ki}^\nu \rceil - 1\}$ 
21        $\mathcal{B} \leftarrow \mathcal{C}$ 

```

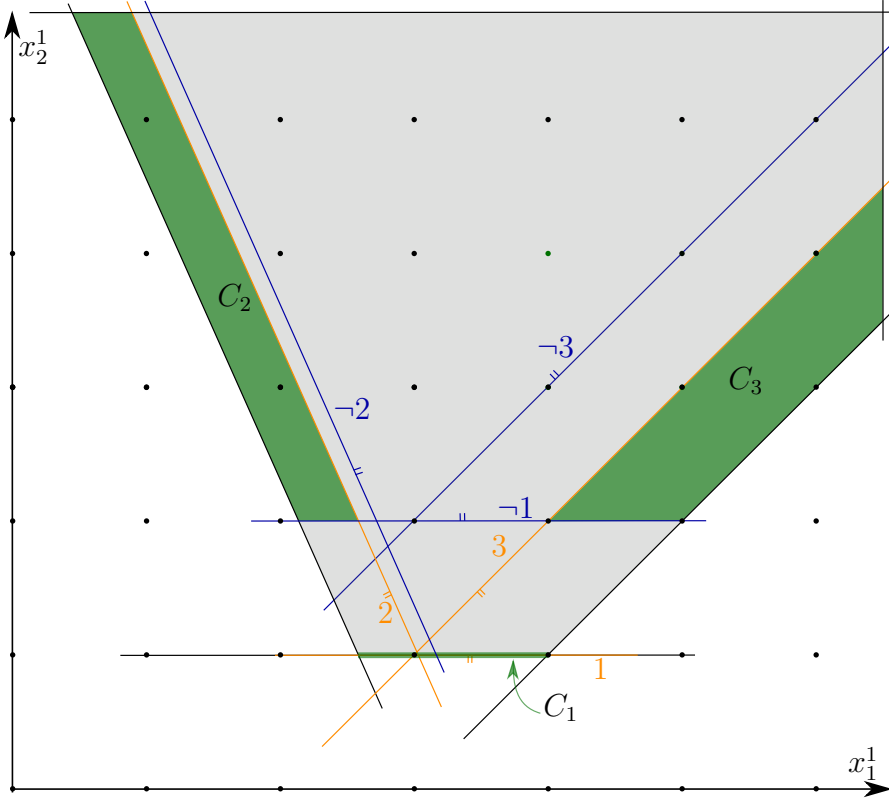


Figure 3: Pruning in a polyhedral strategy set.

continuously relaxed strategy set \widehat{Y}_1 . In this situation, Algorithm 2 was executed with Y and some $\bar{x} \in \widehat{Y}$, which we do not have to name explicitly. We just suppose that the if-statement in line 4 holds true for x_2^1 and $\mathcal{B} = \{Y\}$ still holds. Thus there is only one iteration of the for-loop starting in line 6 which we consecutively describe.

At first, in line 7 the algorithm puts the set $C_1 \times \widehat{Y}_2$ into \mathcal{C} . It contains all possible strategies with active lower bound $x_2^1 = l_2^1$. This happens by posing inequality 1 for this set. For all future sets of this iteration, inequality -1 , with $x_2^1 \geq l_2^1 + 1$, is stated in line 8 without excluding any feasible points.

At this point, the innermost for-loop starts in line 9. The index set $J^{1,2}$ contains only two elements as there are only two linear constraints with $B_{j_2}^1 < 0$ which can be $(1,2)^-$ -active. We can identify them because they are avoiding that x_2^1 is decreased at some point in \widehat{Y}_1 . Thus there are two iterations of this loop. The first one is performed with the left inequality. In line 10, we put $C_2 \times \widehat{Y}_2$ into \mathcal{C} , where inequality 2 coming from (6) holds. Afterwards, we state inequality -2 to prevent an overlap to all incoming sets. The second iteration is done with the right inequality. For this restriction, in addition to the inequalities -1 and -2 , 3 must hold in $C_3 \times \widehat{Y}_2$ which is put into \mathcal{C} in line 10. In line 11, the inequality -3 is stated for future sets which is not necessary anymore, as we exit the two for-loops and replace \mathcal{B} with the three sets in \mathcal{C} .

As a result, we have 3 strategy subsets and for player one there are overall 9 integer points remaining, which could be choices in a Nash equilibrium.

In Section 5 we will see that Example 4.3 is not an isolated case, but that often a considerable part of the feasible set can be pruned by Algorithm 2.

5 Numerical results

In this section, we will solve discrete Nash equilibrium problems with the branch-and-prune procedure presented in Section 4. All players' feasible sets are polyhedral, as defined in (5). The objective functions are defined as

$$\theta_\nu(x) := \frac{1}{2}(x^\nu)^\top Q^\nu x^\nu + (C^\nu x^{-\nu} + b^\nu)^\top x^\nu,$$

with a symmetric, but not necessarily positive semidefinite matrix Q^ν for each player $\nu = 1, \dots, N$. We will consider player convex games as well as games which satisfy Assumption 2.1, but in which the objective functions are only required to be strictly convex with respect to individual variables x_i^ν for $i = 1, \dots, n_\nu$.

In the following, we firstly give details on the concrete implementation of the algorithms. Secondly, we describe how test instances were generated. Lastly, we evaluate and discuss the results.

5.1 Implementation

All algorithms are implemented in Matlab R2020a. We solve all occurring optimization problems via the Matlab interface of Gurobi 9.5.0 which enables us to solve non-convex quadratic optimization problems. The script for the numerical test is executed on an Intel Core i7-9700K CPU @ 3.60GHz with Linux Mint 20 and 32 GB RAM.

Some details of the algorithms from Section 4 can be implemented in various ways and are specified below. The complete code is available in a Git repository¹.

Algorithm 1 At first, the feasible strategy in line 4 is computed with a Gauss-Seidel best response scheme (see [5, Algorithm 1]), because we favor \bar{x} to be already a continuous Nash equilibrium. Within this Gauss-Seidel procedure, we avoid exhaustive calculations if there is only slow or no convergence by executing the while-loop at most 10 times. Note that in this case our approach works as well, as it does not rely on \bar{x} being a continuous Nash equilibrium, but only a feasible point. Secondly, the if-statement in line 7 is verified by solving the optimization problem $Q^\nu(\bar{x}^{-\nu})$ and compare its optimal value to $\theta_\nu(\bar{x})$ for each player. Thirdly, for the considered games, we can use Algorithm 2 as pruning procedure in line 5.

Algorithm 2 The description in Section 4.2 is tailored for convex polyhedral strategy sets. Consecutively, we explain how the if-statements are checked. For each variable, the strict convexity of θ_i^ν in x_i^ν is fulfilled when $Q_{ii}^\nu > 0$. Further, F_i^ν is a linear function and therefore both, convex and concave. Naturally, we can calculate $F_i^\nu(\bar{x}) = Q_{i\star}^\nu \bar{x}^\nu + C_{i\star}^\nu \bar{x}^{-\nu} + b_i^\nu$.

¹https://github.com/schwarze-st/nep_pruning

Equations (1) and (2) are validated by checking the boundedness of $F_{(1)}$ and $F_{(2)}$. Lastly, in line 7-11 and 16-20, we ensure that \bar{x} is always in the first entry of \mathcal{B} (if it is not pruned) by some additional logical queries.

5.2 Generation of test instances

We randomly generate instances of Nash games and name them $C/N \star X \star Y_k$. The first letter is C , if the instance is player convex and N , if not. The second and third signs denote the number of players X and the number of variables Y which each player controls, and k is an index to distinguish instances with similar attributes. For example, the instance $C32_k$ consists of three players with two-dimensional strategy sets and player convexity holds. Table 1 lists all instances and their properties. In the next two paragraphs, we explain these properties and sketch how the instances were generated.

Strategy sets The aim is to generate an arbitrary convex polytope as strategy set for each player. Our approach is to start with a box which has equal side lengths d and its center in the origin. We set $d = 10$ and initiate X_ν from (5) with $l'_i = -5$, $u'_i = 5$, $i = 1, \dots, n_\nu$. Afterwards we sequentially add m linear inequalities. We employ randomness, but also ensure that they are neither redundant nor lead to inconsistency. Note that the complexity of Algorithm 2 significantly increases with the number of inequalities. For the purpose of later comparisons, we also compute the cardinality of Ω which is simply the number of integer points in the common feasible set.

Objective functions For each instance, we compute C^ν , b^ν and Q^ν filled with random values from the interval $(-1, 1)$ for each player. Each entry in C^ν and b^ν is set to zero with a probability of 0.5 in order to reduce density.

For generating a test bed of player convex problems, we update $Q^\nu = (Q^\nu)^\top Q^\nu$. For those instances, we list the minimum eigenvalue λ_{\min} of all Q^ν in Table 1.

In the non-convex test bed, we set $Q^\nu := 0.5 \cdot (Q^\nu)^\top + 0.5 \cdot Q^\nu$ and replace the diagonal entries by their absolute values. By this, we have a symmetric matrix and θ_ν is convex in *single* variables x'_i , $i = 1, \dots, n_\nu$. In this procedure, the matrices often turn out to be, by chance, positive definite. We discard instances where this happens for all players. In Table 1 we can see in the column N_{NC} , how many players have non-positive-semidefinite matrices Q^ν and have thus a non-convex objective function.

Test bed Besides convexity, we subdivide the instances according to their sizes. The **small test bed** are all instances of type $\star 22_\star$. The maximum number of integer points is bounded below 12.000. The **medium test bed** consists of all $\star 23_\star$ and $\star 32_\star$ instances. In $\star 23_{5-8}$, the complexity is increased by adding twice as many constraints. Finally the **large test bed** are all $\star 24_\star$, $\star 33_\star$ and $\star 25_\star$ problems. The number of integer points drastically increases due to exponential growth of the strategy sets in the number of variables. Here, we can only expect convergence in a reasonable time if the pruning procedure eliminates an enormous part of the feasible sets.

Name	λ_{\min}	m	$ \Omega $	Name	N_{NC}	m	$ \Omega $
$C22_1$	0.0943	4	10494	$N22_1$	1	4	10780
$C22_2$	0.2127	4	11227	$N22_2$	1	4	11021
$C22_3$	0.0261	4	11025	$N22_3$	1	4	11009
$C22_4$	0.1989	4	11544	$N22_4$	1	4	11330
$C23_1$	0.0010	4	1269840	$N23_1$	1	4	1420860
$C23_2$	0.0034	4	1355248	$N23_2$	1	4	1471288
$C23_3$	0.0007	4	1484436	$N23_3$	1	4	1485925
$C23_4$	0.1155	4	1441917	$N23_4$	1	4	1366560
$C23_5$	0.0039	8	1275317	$N23_5$	2	8	1229098
$C23_6$	0.0637	8	1167460	$N23_6$	1	8	1286908
$C23_7$	0.0210	8	1253250	$N23_7$	2	8	1358010
$C23_8$	0.0279	8	1338613	$N23_8$	2	8	1285368
$C32_1$	0.0059	4	1212640	$N32_1$	1	4	1236060
$C32_2$	0.0134	4	1190280	$N32_2$	2	4	1282930
$C32_3$	0.0208	4	1259604	$N32_3$	2	4	1102400
$C32_4$	0.1136	4	1178508	$N32_4$	2	4	1212952
$C24_1$	0.0424	8	160436484	$N24_1$	2	8	162311688
$C24_2$	0.0051	8	162716736	$N24_2$	2	8	172443875
$C24_3$	0.0283	8	165580470	$N24_3$	2	8	150670488
$C24_4$	0.0077	8	171940986	$N24_4$	2	8	177972145
$C25_1$	0.0087	10	19359762024	$N25_1$	2	10	18729644934
$C25_2$	0.0010	10	17727657540	$N25_2$	2	10	18526106112
$C25_3$	0.0016	10	19411274700	$N25_3$	2	10	21413827944
$C25_4$	0.0093	10	19329602700	$N25_4$	2	10	18001869885
$C33_1$	0.0077	6	1624593537	$N33_1$	2	6	1472255323
$C33_2$	0.1368	6	1532715320	$N33_2$	3	6	1519384096
$C33_3$	0.1261	6	1472177912	$N33_3$	2	6	1349843690
$C33_4$	0.0002	6	1537992225	$N33_4$	2	6	1421984016

Table 1: Properties of generated instances: Minimum eigenvalue of all Q^ν in the game (λ_{\min}), number of inequalities added per player (m), cardinality of common feasible set ($|\Omega|$), and number of players with θ_ν non-convex w.r.t. x^ν (N_{NC}).

5.3 Evaluation

Subsequently, we investigate if Algorithm 1 is able to compute some or even all equilibria of the test instances. Additionally, we examine how much of the feasible set can be pruned by Algorithm 2 and compare the performance on convex and non-convex instances.

In Table 2 we can see the statistics of the solving process for all player convex instances. We can see in column $|E|$, how many solutions of the NEP are found within the time limit of $t_{max} = 3600$ seconds. The column t_1 marks the run time in seconds, when the *first* equilibrium was found, t_2 when the *last* equilibrium was found and t_3 when the solving process ended. If, $t_3 = t_{max}$, the process did not finish and there is no guarantee that all solutions were found. The statistic $O(t_k)$ displays how often the if-statement in line 6 of Algorithm 1 held true, hence how many integer points were processed at these timestamps. Note that, if the algorithm finished, $O(t_3)/|\Omega|$ tells us the share of integer points that needed to be processed, the rest was pruned by Algorithm 2.

In the **small test bed** we report that the algorithm completed and all equilibria were found. The properties of the instances are quite different: While $C22_1$ and $C22_3$ have two solutions, $C22_4$ has none. In the **medium test bed** we report that for 11 of 12 instances we were able to find provably all solutions within one hour, $C32_3$ being an exception. For this instance, we found three equilibria but did not finish. Notably, there are four instances certified to have no equilibria. Lastly, in the **large test bed** there was no instance for which the procedure finished within an hour. Nevertheless, we found a solution for two instances.

If we were able to compute equilibria, the first one was mostly found in the first ten seconds of the run time. We now analyse the 15 instances for which $t_3 < t_{max}$ holds. In four cases less than 5% of the integer points in Ω were processed, in seven others this share is under 8%. We note that often a large proportion of feasible points could be pruned, the arithmetic mean is 92.5% and the standard deviation 3.6%. We point out that, if $t_3 = t_{max}$, the column $O(t_3)/|\Omega|$ has no similar interpretation. It only says how many integer points were processed in the given time.

For the non-convex test bed, displayed in Table 3, we see similar results. We report that all small and medium instances were solved completely. We have eight provably inconsistent instances, three more than before. Again, there was no large instance solved completely in the time limit, but we found a solution for three instances. For the instances with $t_3 < t_{max}$, the mean share of pruned points is 92.1% with a standard deviation of 4%. Hence, in our randomly generated test bed the convexity in individual variables is sufficient to be able to prune a large proportion of feasible points.

	$ E $	t_1	t_2	t_3	$O(t_1)$	$O(t_2)$	$O(t_3)$	$O(t_3)/ \Omega $
$C22_1$	2	0.09	19.98	21.10	1	1173	1249	0.11902
$C22_2$	1	0.05	-	8.58	1	-	542	0.04828
$C22_3$	2	0.11	0.31	8.68	5	18	700	0.06349
$C22_4$	0	-	-	13.41	-	-	640	0.05544
$C23_1$	1	10.37	-	923.22	498	-	60203	0.04741
$C23_2$	0	-	-	3328.80	-	-	228099	0.16831
$C23_3$	2	0.92	595.20	1345.72	39	42356	96392	0.06494
$C23_4$	0	-	-	2472.04	-	-	172608	0.11971
$C23_5$	0	-	-	1791.36	-	-	98942	0.07758
$C23_6$	1	179.63	-	1171.00	10872	-	73213	0.06271
$C23_7$	1	2.01	-	1561.84	95	-	85178	0.06797
$C23_8$	2	0.62	4.09	2065.83	18	158	136965	0.10232
$C32_1$	2	1.07	49.69	837.48	16	2044	43151	0.03558
$C32_2$	1	0.07	-	1525.47	1	-	74208	0.06234
$C32_3$	3	0.02	28.06	t_{\max}	1	1048	162447	0.12897
$C32_4$	0	-	-	678.37	-	-	33663	0.02856
$C24_1$	1	23.41	-	t_{\max}	970	-	195508	0.00122
$C24_2$	0	-	-	t_{\max}	-	-	186991	0.00115
$C24_3$	0	-	-	t_{\max}	-	-	169657	0.00102
$C24_4$	0	-	-	t_{\max}	-	-	173599	0.00101
$C25_1$	0	-	-	t_{\max}	-	-	125718	0.00001
$C25_2$	0	-	-	t_{\max}	-	-	131850	0.00001
$C25_3$	0	-	-	t_{\max}	-	-	169275	0.00001
$C25_4$	0	-	-	t_{\max}	-	-	142249	0.00001
$C33_1$	1	1.87	-	t_{\max}	19	-	145035	0.00009
$C33_2$	0	-	-	t_{\max}	-	-	141326	0.00009
$C33_3$	0	-	-	t_{\max}	-	-	143666	0.00010
$C33_4$	0	-	-	t_{\max}	-	-	140279	0.00009

Table 2: Results on convex test bed: Number of computed equilibria ($|E|$), timestamps when the first (t_1) and the last (t_2) equilibrium was found, timestamp at the end of computation (t_3), and number of processed integer points at timestamp t_i ($O(t_i)$).

	$ E $	t_1	t_2	t_3	$O(t_1)$	$O(t_2)$	$O(t_3)$	$O(t_3)/ \Omega $
$N22_1$	2	0.30	5.52	9.24	5	306	559	0.05186
$N22_2$	1	1.04	-	17.80	24	-	1060	0.09618
$N22_3$	0	-	-	9.05	-	-	458	0.04160
$N22_4$	0	-	-	31.45	-	-	1745	0.15402
$N23_1$	0	-	-	1533.53	-	-	112134	0.07892
$N23_2$	1	0.98	-	1733.78	34	-	100641	0.06840
$N23_3$	0	-	-	3600.01	-	-	232560	0.15651
$N23_4$	1	0.38	-	658.08	7	-	49078	0.03591
$N23_5$	2	0.07	120.68	1418.29	1	5591	73366	0.05969
$N23_6$	0	-	-	3600.01	-	-	174269	0.13542
$N23_7$	0	-	-	1258.18	-	-	62363	0.04592
$N23_8$	0	-	-	2214.10	-	-	120203	0.09352
$N32_1$	0	-	-	2777.24	-	-	131534	0.10641
$N32_2$	2	3.48	42.41	1757.10	80	1669	77605	0.06049
$N32_3$	1	1.33	-	327.68	15	-	14325	0.01299
$N32_4$	4	7.03	369.06	1662.22	259	16323	72359	0.05966
$N24_1$	1	1.13	-	t_{\max}	11	-	142222	0.00088
$N24_2$	0	-	-	t_{\max}	-	-	167211	0.00097
$N24_3$	0	-	-	t_{\max}	-	-	153328	0.00102
$N24_4$	0	-	-	t_{\max}	-	-	137245	0.00077
$N25_1$	0	-	-	t_{\max}	-	-	115337	0.00001
$N25_2$	1	0.34	-	t_{\max}	1	-	105781	0.00001
$N25_3$	0	-	-	t_{\max}	-	-	127607	0.00001
$N25_4$	0	-	-	t_{\max}	-	-	130861	0.00001
$N33_1$	0	-	-	t_{\max}	-	-	118451	0.00008
$N33_2$	1	563.56	-	t_{\max}	17733	-	103851	0.00007
$N33_3$	0	-	-	t_{\max}	-	-	117543	0.00009
$N33_4$	0	-	-	t_{\max}	-	-	127009	0.00009

Table 3: Results on non-convex test bed: Number of computed equilibria ($|E|$), timestamps when the first (t_1) and the last (t_2) equilibrium was found, timestamp at the end of computation (t_3), and number of processed integer points at timestamp t_i ($O(t_i)$).

Name	GT_{GS}	GT_{isNE}	GT_{bd}	GT_{con}	GT_{tot}	Name	GT_{GS}	GT_{isNE}	GT_{bd}	GT_{con}	GT_{tot}
$C22_1$	0.41	0.10	0.30	0.19	20.38	$N22_1$	0.38	0.21	0.25	0.17	8.92
$C22_2$	0.39	0.12	0.31	0.18	8.28	$N22_2$	0.48	0.15	0.24	0.13	17.22
$C22_3$	0.42	0.15	0.30	0.14	8.36	$N22_3$	0.57	0.10	0.21	0.13	8.76
$C22_4$	0.50	0.09	0.23	0.18	12.96	$N22_4$	0.48	0.15	0.22	0.15	30.29
$C23_1$	0.40	0.15	0.32	0.13	881.13	$N23_1$	0.42	0.14	0.27	0.17	1449.30
$C23_2$	0.41	0.14	0.32	0.13	3099.11	$N23_2$	0.50	0.11	0.27	0.12	1670.18
$C23_3$	0.41	0.14	0.33	0.11	1277.52	$N23_3$	0.49	0.13	0.29	0.09	3314.61
$C23_4$	0.41	0.15	0.29	0.15	2338.10	$N23_4$	0.46	0.15	0.32	0.07	626.65
$C23_5$	0.34	0.11	0.26	0.28	1717.23	$N23_5$	0.43	0.16	0.20	0.22	1361.99
$C23_6$	0.34	0.12	0.27	0.26	1123.41	$N23_6$	0.32	0.24	0.22	0.22	3427.55
$C23_7$	0.31	0.11	0.25	0.33	1504.25	$N23_7$	0.41	0.18	0.21	0.20	1212.60
$C23_8$	0.37	0.13	0.26	0.23	1946.25	$N23_8$	0.38	0.21	0.22	0.20	2117.23
$C32_1$	0.44	0.10	0.30	0.16	806.45	$N32_1$	0.42	0.12	0.29	0.16	2664.56
$C32_2$	0.41	0.09	0.32	0.18	1468.77	$N32_2$	0.44	0.10	0.25	0.20	1697.91
$C32_3$	0.44	0.09	0.31	0.17	3440.14	$N32_3$	0.49	0.12	0.24	0.14	315.61
$C32_4$	0.44	0.11	0.29	0.16	651.54	$N32_4$	0.44	0.08	0.27	0.21	1605.63
$C24_1$	0.34	0.17	0.27	0.22	3282.63	$N24_1$	0.34	0.30	0.20	0.15	3396.05
$C24_2$	0.36	0.13	0.32	0.19	3183.16	$N24_2$	0.40	0.15	0.27	0.18	3287.20
$C24_3$	0.35	0.11	0.30	0.24	3277.73	$N24_3$	0.37	0.19	0.26	0.18	3302.92
$C24_4$	0.35	0.11	0.32	0.22	3248.49	$N24_4$	0.35	0.29	0.21	0.15	3391.74
$C25_1$	0.31	0.13	0.40	0.16	3034.03	$N25_1$	0.31	0.30	0.22	0.16	3318.78
$C25_2$	0.36	0.12	0.42	0.11	2938.86	$N25_2$	0.41	0.23	0.27	0.09	3257.14
$C25_3$	0.36	0.14	0.36	0.13	3110.01	$N25_3$	0.38	0.25	0.26	0.10	3233.81
$C25_4$	0.33	0.11	0.37	0.18	3124.67	$N25_4$	0.38	0.25	0.27	0.10	3240.55
$C33_1$	0.38	0.08	0.33	0.21	3301.03	$N33_1$	0.45	0.12	0.30	0.13	3322.80
$C33_2$	0.38	0.09	0.36	0.17	3278.18	$N33_2$	0.45	0.10	0.29	0.16	3278.14
$C33_3$	0.39	0.09	0.35	0.17	3286.78	$N33_3$	0.47	0.12	0.31	0.09	3291.73
$C33_4$	0.41	0.08	0.36	0.15	3257.12	$N33_4$	0.43	0.12	0.30	0.15	3314.68

Table 4: Run time details on test beds: Total run time of Gurobi (GT_{tot}) and shares of run time for the Gauss-Seidel alg. (GT_{GS}), checking if \bar{x} is an equilibrium (GT_{isNE}), checking boundedness of $F_{(1)}$ and $F_{(2)}$ (GT_{bd}) and checking consistency (GT_{con}).

In contrast, we detect differences in the run time between the convex and the non-convex case. Table 4 reports how much of the total run time is caused by solving optimization problems and checking consistency with Gurobi (GT_{tot}). Of this time, we see on the left the fractions caused by different tasks. In the non-convex test bed, aiming to solve the continuously relaxed problems with the Gauss-Seidel method and checking whether \bar{x} solves the NEP takes on average a larger proportion of GT_{tot} (42% and 17% instead of 38% and 12%). For these two tasks, non-convex optimization problems need to be solved. The other two columns only report the time fractions needed for consistency checks. Overall, one can also see in the tables which parts of the algorithms require the most run time to assess where improvements are most beneficial.

6 Conclusion

This paper presents novel theoretical results on pruning for discrete Nash equilibrium problems. The required activity of particular constraints leads to synchronous branching and pruning of the strategy sets. Furthermore, we showed in a numerical study that a noteworthy part of the joint feasible set can be pruned by following this rationale. This was demonstrated for polyhedral strategy sets and (not necessarily convex) quadratic objective functions. It remains to be investigated if these results can also be applied to broader problem classes like, for example, generalized Nash equilibrium problems.

References

- [1] Pietro Belotti, Christian Kirches, Sven Leyffer, Jeff Linderoth, James Luedtke, and Ashutosh Mahajan. Mixed-integer nonlinear optimization. *Acta Numerica*, 22:1–131, 2013.
- [2] Axel Dreves, Christian Kanzow, and Oliver Stein. Nonsmooth optimization reformulations of player convex generalized Nash equilibrium problems. *Journal of Global Optimization*, 53(4):587–614, 2012.
- [3] Francisco Facchinei, Andreas Fischer, and Veronica Piccialli. Generalized Nash equilibrium problems and Newton methods. *Mathematical Programming*, 117(1-2):163–194, 2009.
- [4] Francisco Facchinei and Christian Kanzow. Generalized Nash equilibrium problems. *For*, 5(3):173–210, 2007.
- [5] Francisco Facchinei, Veronica Piccialli, and Marco Sciandrone. Decomposition algorithms for generalized potential games. *Computational Optimization and Applications*, 50(2):237–262, 2011.
- [6] Tobias Harks and Julian Schwarz. Generalized Nash Equilibrium Problems with Mixed-Integer Variables. Preprint 2107.13298, arXiv, 2021.

- [7] Dov Monderer and Lloyd S. Shapley. Potential Games. *Games and Economic Behavior*, 14(1):124–143, May 1996.
- [8] John F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36(1):48–49, 1950.
- [9] John F. Nash. Non-Cooperative Games. *The Annals of Mathematics*, 54(2):286–295, 1951.
- [10] Massimo Pappalardo, Giandomenico Mastroeni, and Mauro Passacantando. Merit functions: A bridge between optimization and equilibria. *For*, 12(1):1–33, 2014.
- [11] Simone Sagratella. Computing all solutions of Nash equilibrium problems with discrete strategy sets. *SIAM Journal on Optimization*, 26(4):2190–2218, 2016.
- [12] Simone Sagratella. On generalized Nash equilibrium problems with linear coupling constraints and mixed-integer variables. *Optimization*, 68(1):197–226, 2019.

A Theorem 3.2 generalizes [11, Prop. 3.1]

Subsequently, it will be shown that Theorem 3.2 is a generalization of Proposition 3.1 from [11]. This article considers box-constrained strategy sets X_ν with

$$g^\nu(x^\nu) := \begin{pmatrix} x^\nu - u^\nu \\ l^\nu - x^\nu \end{pmatrix} \leq 0, \quad l^\nu, u^\nu \in \mathbb{Z}^{n_\nu}. \quad (8)$$

Obviously, the gradient $\nabla_{x^\nu} g_j^\nu(x^\nu)$ of the constraint j equals the unit vector e_j for $j \in \{1, \dots, n_\nu\}$ and $-e_j$ for $j \in \{n_\nu + 1, \dots, 2n_\nu\}$, respectively.

Proposition A.1 ([11, Proposition 3.1]). *Suppose that Ω is defined by box constraints. Let $\bar{x} \in \Omega$ be an integer solution of the continuously relaxed problem \widehat{NEP} . Let us consider a generic player ν . Suppose that an index $i \in \{1, \dots, n_\nu\}$ exists, such that θ_ν is strictly convex with respect to x_i^ν and that one of the two possibilities holds:*

- (i) *Given that $\bar{x}_i^\nu = l_i^\nu$, F_i^ν is convex and for each player $\mu \in \{1, \dots, N\}$ and each index $j \in \{1, \dots, n_\mu\}$, such that $(\mu, j) \neq (\nu, i)$, it holds $\bar{x}_j^\mu = l_j^\mu$ if $\nabla_{x_j^\mu} F_i^\nu(\bar{x}) > 0$ and $\bar{x}_j^\mu = u_j^\mu$ if $\nabla_{x_j^\mu} F_i^\nu(\bar{x}) < 0$.*
- (ii) *Given that $\bar{x}_i^\nu = u_i^\nu$, F_i^ν is concave and for each player $\mu \in \{1, \dots, N\}$ and each index $j \in \{1, \dots, n_\mu\}$, such that $(\mu, j) \neq (\nu, i)$, it holds $\bar{x}_j^\mu = u_j^\mu$ if $\nabla_{x_j^\mu} F_i^\nu(\bar{x}) > 0$ and $\bar{x}_j^\mu = l_j^\mu$ if $\nabla_{x_j^\mu} F_i^\nu(\bar{x}) < 0$.*

Then any point $\tilde{x} \in \Omega$ such that $\tilde{x}_i^\nu \neq \bar{x}_i^\nu$ cannot be a solution of NEP.

More specifically, it will now be proven that, if all prerequisites of Proposition A.1 are fulfilled, Theorem 3.2 is applicable and yields the same result. Lemma A.2 shows that the requirements on $F_i^\nu(\bar{x})$ in Theorem 3.2.(i)/(ii) follow from the assumptions in Proposition A.1.(i)/(ii). Given that, Corollary A.3 proves that (1) and (2) also follow for the respective cases and that Theorem 3.2 yields the same result.

Lemma A.2. Suppose that Ω is defined by box constraints and that θ_ν is strictly convex with respect to x_i^ν , as stated in Proposition A.1. From \bar{x} being a solution of \widehat{NEP} and $l_i^\nu \neq u_i^\nu$ follow the two statements:

(i) If $\bar{x}_i^\nu = l_i^\nu$, then $F_i^\nu(\bar{x}) \geq 0$ must hold.

(ii) If $\bar{x}_i^\nu = u_i^\nu$, then $F_i^\nu(\bar{x}) \leq 0$ must hold.

Proof. Since \bar{x} is a solution of \widehat{NEP} , for each ν the point \bar{x}^ν is a minimal point of the problem $\widehat{Q}^\nu(\bar{x}^{-\nu})$. Since the gradients of active box restrictions are linearly independent, \bar{x}^ν is a KKT point of $\widehat{Q}^\nu(\bar{x}^{-\nu})$ and

$$F^\nu(\bar{x}) + \sum_{j \in I_0(\bar{x}^\nu, \widehat{X}_\nu)} \lambda_j \cdot \nabla_{x^\nu} g_j^\nu(\bar{x}^\nu) = 0$$

holds with $\lambda_j \in \mathbb{R}_{\geq 0}$ for all $j \in I_0(\bar{x}^\nu, \widehat{X}_\nu)$. For the i -th row of this equation, solely the gradients of the i -th variable's upper and lower bound constraints are non-zero, $\nabla_{x^\nu} g_i^\nu = e_i$ and $\nabla_{x^\nu} g_{n_\nu+i}^\nu = -e_i$. By $l_i^\nu \neq u_i^\nu$ only one of them can be in the active index set.

- If $\bar{x}_i^\nu = l_i^\nu$, the constraint $g_{n_\nu+i}^\nu$ is active and statement (i) follows from $F_i^\nu(\bar{x}) + \lambda_{n_\nu+i} \cdot (-1) = 0$.
- If $\bar{x}_i^\nu = u_i^\nu$, the constraint g_i^ν is active and statement (ii) follows from $F_i^\nu(\bar{x}) + \lambda_i \cdot 1 = 0$.

□

Corollary A.3. Theorem 3.2 is a generalization of Proposition A.1.

Proof. Since, as stated in Proposition A.1, the set Ω is defined by box constraints, the strategy sets are defined by convex (linear) functions. Given Lemma A.2, it remains to show that (1) and (2) follow in the two particular cases and that the consequential statements are equivalent.

Firstly, we start with Proposition A.1.(i). For an arbitrary player μ we have:

- Let $J_\mu \subseteq \{1, \dots, n_\mu\}$ be the set of all indices j such that $(\mu, j) \neq (\nu, i)$ and $\nabla_{x_j^\mu} F_i^\nu(\bar{x}) < 0$. Thus, $\bar{x}_j^\mu = u_j^\mu$ for all $j \in J_\mu$ and the box constraint $g_j^\mu(x^\mu) = x_j^\mu - u_j^\mu$ is active ($j \in I_0(\bar{x}^\mu, \widehat{X}_\mu)$). From $\nabla_{x^\mu} g_j^\mu(x^\mu) = e_j$ follows

$$d_j^\mu \leq 0 \text{ for all } d^\mu \in L_{\leq}(\bar{x}^\mu, \widehat{X}_\mu).$$

- Let $K_\mu \subseteq \{1, \dots, n_\mu\}$ be the set of all indices k such that $(\mu, k) \neq (\nu, i)$ and $\nabla_{x_k^\mu} F_i^\nu(\bar{x}) > 0$. Thus, $\bar{x}_k^\mu = l_k^\mu$ for all $k \in K_\mu$ and the box constraint $g_{n_\mu+k}^\mu(x^\mu) = l_k^\mu - x_k^\mu$ is active ($n_\mu + k \in I_0(\bar{x}^\mu, \widehat{X}_\mu)$). From $\nabla_{x^\mu} g_{n_\mu+k}^\mu(x^\mu) = -e_k$ follows

$$d_k^\mu \geq 0 \text{ for all } d^\mu \in L_{\leq}(\bar{x}^\mu, \widehat{X}_\mu).$$

If $\mu = \nu$, this also applies to the index i :

- $\nabla_{x_i^\nu} F_i^\nu(\bar{x}) > 0$ due to strict convexity of θ_ν in this component,
- $g_{n_\nu+i}^\nu(x^\mu)$ is also active and $\nabla_{x^\nu} g_{n_\nu+i}^\nu(x^\mu) = -e_i$.

Together, for every player $\mu \in \{1, \dots, N\}$ the scalar product

$$\langle \nabla_{x^\mu} F_i^\nu(\bar{x}), d^\mu \rangle = \sum_{p=1}^{n_\mu} \nabla_{x_p^\mu} F_i^\nu(\bar{x}) \cdot d_p^\mu$$

is non-negative for all $d^\mu \in L_{\leq}(\bar{x}^\mu, \widehat{X}_\mu)$ and therefore (1) is fulfilled. This is easy to verify with help of the statements above. For each index $p \in \{1, \dots, n_\mu\}$ one of these three possibilities holds:

- $p \in J_\mu$: $\nabla_{x_p^\mu} F_i^\nu(\bar{x}) < 0$ and $d_p^\mu \leq 0$.
- $p \in K_\mu$: $\nabla_{x_p^\mu} F_i^\nu(\bar{x}) > 0$ and $d_p^\mu \geq 0$.
- $\nabla_{x_p^\mu} F_i^\nu(\bar{x}) = 0$.

For any point $\tilde{x} \in \Omega$ with $\tilde{x}_i^\nu \neq \bar{x}_i^\nu$ it follows that $\tilde{x}_i^\nu \geq \bar{x}_i^\nu + 1 = l_i^\nu + 1$ and directly the feasibility of \tilde{x} , which is constructed like in Theorem 3.2. Therefore, the conclusion of both rationales is equivalent.

Secondly, in the case of Proposition A.1.(ii) we obtain that (2) holds with similar argumentation and yields the same conclusion, we present it for the sake of completeness. With the sets J_μ and K_μ defined as above, for an arbitrary player μ and index p one of these three possibilities holds:

- $p \in J_\mu$: Thus $\bar{x}_p^\mu = l_p^\mu$ and the box restriction $g_{n_\mu+p}^\mu(x^\mu) = l_p^\mu - x_p^\mu$ is active. From $\nabla_{x^\mu} g_{n_\mu+p}^\mu(x^\mu) = -e_p$ follows

$$d_p^\mu \geq 0 \text{ for all } d^\mu \in L_{\leq}(\bar{x}^\mu, \widehat{X}_\mu).$$

- $p \in K_\mu$: Thus $\bar{x}_p^\mu = u_p^\mu$ and the box restriction $g_p^\mu(x^\mu) = x_p^\mu - u_p^\mu$ is active. From $\nabla_{x^\mu} g_p^\mu(x^\mu) = e_p$ follows

$$d_p^\mu \leq 0 \text{ for all } d^\mu \in L_{\leq}(\bar{x}^\mu, \widehat{X}_\mu).$$

If $(\mu, p) = (\nu, i)$, this also applies:

- $\nabla_{x_i^\nu} F_i^\nu(\bar{x}) > 0$ due to strict convexity of θ_ν in this component,
- $g_i^\nu(x^\mu)$ is also active and $\nabla_{x^\nu} g_i^\nu(x^\mu) = e_i$.
- $\nabla_{x_p^\mu} F_i^\nu(\bar{x}) = 0$.

Together, for every player the scalar product

$$\langle -\nabla_{x^\mu} F_i^\nu(\bar{x}), d^\mu \rangle = \sum_{p=1}^{n_\mu} -\nabla_{x_p^\mu} F_i^\nu(\bar{x}) d_p^\mu$$

is non-negative for all $d^\mu \in L_{\leq}(\bar{x}^\mu, \widehat{X}_\mu)$ and therefore (2) is fulfilled.

For any point $\tilde{x} \in \Omega$ with $\tilde{x}_i^\nu \neq \bar{x}_i^\nu$ it follows that $\tilde{x}_i^\nu \leq \bar{x}_i^\nu - 1 = u_i^\nu - 1$ and directly the feasibility of \tilde{x} . Again, the conclusion of both rationales is equivalent. \square

Remark A.4. *The assumption in Proposition A.1 that \bar{x} is an integer solution of \widehat{NEP} is not necessary to obtain the statement, but it is sufficient to require that \bar{x} is any solution of \widehat{NEP} . Nevertheless, this does not restrict the applicability of Proposition A.1 very much, because for all variables x_j^μ with $\nabla_{x_j^\mu} F_i^\nu(\bar{x}) \neq 0$ it is required that their value coincides with one of the integer valued bounds anyway.*

B Algorithm 3: Remove strategy from search space

Algorithm 3 is employed after \bar{x} is processed by Algorithm 1. As we have $\bar{x} \in B_1$, this set is divided into at most $2 \cdot n$ sets, excluding \bar{x} and preserving all other integer points from B_1 . Note that at least one component of a point x in any new subset of B_1 must be different from \bar{x} in order to achieve the exclusion.

Algorithm 3: [11, Procedure C] Branching out a single point from the strategy set.

Input: Strategy subset B_1 of a discrete Nash game; Feasible strategy \bar{x}

Output: List of sets \mathcal{B}^+ such that \bar{x} is not contained

- 1 Initialize list $\mathcal{B}^+ := \{\}$
 - 2 initialize sets $B_i^+ := B_1$ for all $i \in \{1, \dots, 2n\}$
 - 3 **for** $j = 1$ **to** n **do**
 - 4 $B_{2j-1}^+ \leftarrow B_{2j-1}^+ \cap \{x \in \mathbb{R}^n : x_j \leq \bar{x}_j - 1\}$
 - 5 **if** $B_{2j-1}^+ \neq \emptyset$ **then**
 - 6 \lfloor put B_{2j-1}^+ in \mathcal{B}^+
 - 7 $B_{2j}^+ \leftarrow B_{2j}^+ \cap \{x \in \mathbb{R}^n : x_j \geq \bar{x}_j + 1\}$
 - 8 **if** $B_{2j}^+ \neq \emptyset$ **then**
 - 9 \lfloor put B_{2j}^+ in \mathcal{B}^+
 - 10 **for** $t = 2j + 1$ **to** $2n$ **do**
 - 11 \lfloor $B_t^+ \leftarrow B_t^+ \cap \{x \in \mathbb{R}^n : x_j = \bar{x}_j\}$
-