Algorithmic Game Theory

Algorithmische Spieltheorie

How Hard Is It to Find a Nash Equilibrium?

Wintersemester 2022/2023

Dozent: Prof. Dr. J. Rothe

Algorithmic Game Theory

- Algorithmic game theory is not satisfied with merely an existence result for an important solution concept such as Nash equilibrium in mixed strategies.
- Rather, one seeks to determine the complexity of finding it.
- Unfortunately, for most games we know only algorithms either with an unknown running time (because it is very difficult to analyze these cases) or with an exponential running time in the worst case.
- Merely for some special cases, it was possible to design efficient algorithms for computing Nash equilibria.
- For example, Nash equilibria in mixed strategies can be computed in polynomial time for *finite, two-player, zero-sum games*.

Constant-Sum and Zero-Sum Games

Definition (constant-sum and zero-sum game)

A noncooperative game in normal form with the set $\mathscr{S} = S_1 \times S_2 \times \cdots \times S_n$ of strategy profiles, where player i's gain function is g_i, is a *constant-sum game* if there is a constant c such that for each strategy profile $\vec{s} \in \mathscr{S}$,

$$\sum_{i=1}^{n} g_i(\vec{s}) = c.$$

For $c = 0$, it is called a *zero-sum game*.

Example

Examples of zero-sum games are:

- penalty game,
- paper-rock-scissors game.

Maxmin Strategies and Values

Definition (maxmin strategy and value)

Consider a noncooperative game in normal form for n players with the set $\Pi = \Pi_1 \times \Pi_2 \times \cdots \times \Pi_n$ of mixed-strategy profiles, where player i's expected gain function is G_i.

1. The *maxmin strategy of player i* is defined as

$$\arg\max_{\pi_i \in \Pi_i} \min_{\vec{\pi}_{-i} \in \Pi_{-i}} G_i(\vec{\pi}_{-i}, \pi_i).$$

2. The *maxmin value for player i* is defined as

$$\max_{\pi_i \in \Pi_i} \min_{\vec{\pi}_{-i} \in \Pi_{-i}} G_i(\vec{\pi}_{-i}, \pi_i).$$

Minmax Strategies and Values: 2 Players

Definition (minmax strategy and value)

Consider a noncooperative game in normal form for 2 players, i and $-i$, with the set $\Pi = \Pi_i \times \Pi_{-i}$ of mixed-strategy profiles and expected gain functions G_i and G_{-i}.

1. The *minmax strategy of player i against player $-i$* is defined as

$$\arg\min_{\pi_i \in \Pi_i} \max_{\pi_{-i} \in \Pi_{-i}} G_{-i}(\pi_i, \pi_{-i}).$$

2. The *minmax value for player $-i$* is defined as

$$\min_{\pi_i \in \Pi_i} \max_{\pi_{-i} \in \Pi_{-i}} G_{-i}(\pi_i, \pi_{-i}).$$

Minmax Strategies and Values: $n > 2$ Players

Definition (minmax strategy and value)

Consider a noncooperative game in normal form for $n > 2$ players with the set $\Pi = \Pi_1 \times \Pi_2 \times \cdots \times \Pi_n$ of mixed-strategy profiles, where player i's expected gain function is G_i.

1. The *minmax strategy of player i against player $j \neq i$* is defined as i's component of the mixed-strategy profile $\vec{\pi}_{-j}$ in the expression
$$\arg\min_{\vec{\pi}_{-j} \in \Pi_{-j}} \max_{\pi_j \in \Pi_j} G_j(\vec{\pi}_{-j}, \pi_j),$$
where $-j$ denotes the set of players other than j.
2. The *minmax value for player j* is defined as
$$\min_{\vec{\pi}_{-j} \in \Pi_{-j}} \max_{\pi_j \in \Pi_j} G_j(\vec{\pi}_{-j}, \pi_j).$$

Maxmin and Minmax Strategy Profiles

Definition (maxmin and minmax strategy profile)

1. A mixed-strategy profile $\vec{\pi} = (\pi_1, \pi_2, \dots, \pi_n) \in \Pi = \Pi_1 \times \Pi_2 \times \cdots \times \Pi_n$ is said to be a *maxmin strategy profile* if for each i, $1 \leq i \leq n$, π_i is a maxmin strategy of player i.
2. A mixed-strategy profile $\vec{\pi} = (\pi_i, \pi_{-i})$ for two players is a *minmax strategy profile* if π_i is a minmax strategy of player i against player $-i$ and π_{-i} is a minmax strategy of player $-i$ against player i.

Remark

- *In two-player, zero-sum games in normal form, a player's maxmin value is always equal to her minmax value.*
- *In normal-form, zero-sum games with more than two players, a player's maxmin value is always at most her minmax value.*

Minimax Theorem by von Neumann (1928)

Theorem (Minimax Theorem)

In every finite, zero-sum game in normal form with two players who have exptected gain functions G_1 and G_2, where

- $\mathscr{A}$ *is the set of mixed strategies for player* 1 *and*
- $\mathscr{B}$ *is the set of mixed strategies for player* 2,

there exists a value v^* such that the following hold:

1. *For player* 1, *there exists a mixed strategy $\alpha^* \in \mathscr{A}$ such that*

$$\max_{\alpha \in \mathscr{A}} \min_{\beta \in \mathscr{B}} G_1(\alpha, \beta) = \min_{\beta \in \mathscr{B}} G_1(\alpha^*, \beta) = v^*.$$

2. *For player* 2, *there exists a mixed strategy $\beta^* \in \mathscr{B}$ such that*

$$\min_{\beta \in \mathscr{B}} \max_{\alpha \in \mathscr{A}} G_1(\alpha, \beta) = \max_{\alpha \in \mathscr{A}} G_1(\alpha, \beta^*) = v^*.$$

Minimax Theorem by von Neumann (1928): Proof

Proof: Let $(\alpha^*, \beta^*) \in \mathscr{A} \times \mathscr{B}$ be a Nash equilibrium in mixed strategies for our game. (We know that such an equilibrium exists.)

- We set $v^* = G_1(\alpha^*, \beta^*)$ and we claim that

$$\max_{\alpha \in \mathscr{A}} \min_{\beta \in \mathscr{B}} G_1(\alpha, \beta) \geq \min_{\beta \in \mathscr{B}} G_1(\alpha^*, \beta) \geq v^*. \tag{1}$$

- The first inequality holds because
  - on the LHS we have the maximum over all possible values of α, and
  - on the RHS we use one particular value, α^*.
- The second inequality holds because (α^*, β^*) is a Nash equilibrium in mixed strategies and, thus, β^* is a best response for α^*.

Minimax Theorem by von Neumann (1928): Proof

- More concretely, by definition of a best response, for each mixed strategy $\beta \in \mathscr{B}$ we have

$$G_2(\alpha^*, \beta) \leq G_2(\alpha^*, \beta^*).$$

- Since our game is a zero-sum game, $G_2(\alpha^*, \beta) \leq G_2(\alpha^*, \beta^*)$ implies

$$G_1(\alpha^*, \beta) \geq G_1(\alpha^*, \beta^*) = v^*.$$

Minimax Theorem by von Neumann (1928): Proof

- However, by applying similar reasoning, we can also show that

$$\max_{\alpha \in \mathscr{A}} \min_{\beta \in \mathscr{B}} G_1(\alpha, \beta) \leq \max_{\alpha \in \mathscr{A}} G_1(\alpha, \beta^*) \leq v^*. \tag{2}$$

- The first inequality holds because
  - on the LHS the maximum operator has to find a mixed strategy α that does well against every possible mixed strategy β, whereas
  - on the RHS it only has to do well against one fixed strategy, β^*.

- The second inequality holds because α^* is a best response to β^* and, thus, for every mixed strategy $\alpha \in \mathscr{A}$ we have that

$$G_1(\alpha, \beta^*) \leq G_1(\alpha^*, \beta^*) = v^*.$$

Minimax Theorem by von Neumann (1928): Proof

- By combining inequalities (1) and (2), we obtain the first part of the theorem:

$$\max_{\alpha \in \mathscr{A}} \min_{\beta \in \mathscr{B}} G_1(\alpha, \beta) = \min_{\beta \in \mathscr{B}} G_1(\alpha^*, \beta) = v^*.$$

- Analogous reasoning for player 2 (exercise!) gives the second part as well:

$$\min_{\beta \in \mathscr{B}} \max_{\alpha \in \mathscr{A}} G_1(\alpha, \beta) = \max_{\alpha \in \mathscr{A}} G_1(\alpha, \beta^*) = v^*. \quad ❑$$

Minimax Theorem: Why Is It Important?

Remark

Because in a finite, two-player, zero-sum game in normal form:

- *Every player's maxmin value equals her minmax value. By convention, player 1's maxmin value is called the value of the game.*
- *For both players, the set of maxmin strategies coincides with the set of minmax strategies.*
- *Every maxmin (equivalently, minmax) strategy profile is a Nash equilibrium in mixed strategies.*
  Conversely, every Nash equilibrium in mixed strategies is a maxmin (equivalently, minmax) strategy profile.
  Thus, all Nash equilibria have the same gain vector (in which player 1 gets the value of the game).

Computing Nash Equilibria in 2-Player, Zero-Sum Games

- We show that Nash equilibria in two-player, zero-sum games in normal form can be expressed as a *linear program (LP)*.
- That linear programs can actually be solved in polynomial time was shown only near the end of the 1970s by Hačijan (1979), whose algorithm is based on the *ellipsoid method*.
- His work was a milestone in linear programming, since it had been unclear before if linear programs are solvable in polynomial time.
- Also later developed procedures such as the *interior point methods*, which too run in polynomial time, have been inspired by his work.
- In practice, the older *simplex methods* are still in use.

Computing Nash Equilibria in 2-Player, Zero-Sum Games

- Consider a zero-sum game in normal form for two players, 1 and 2, with the sets
  - $\mathscr{S} = S_1 \times S_2$ of pure-strategy profiles, with gain functions g_1 and g_2;
  - $\Pi = \Pi_1 \times \Pi_2$ of mixed-strategy profiles, with expected gain functions G_1 and G_2.
- Let G_i^*, $i \in \{1,2\}$, be the expected gain of player i in a mixed Nash equilibrium (i.e., G_1^* is the value of the game).
- Since the game is zero-sum, $G_1^* = -G_2^*$.
- By the minimax theorem,
  - G_1^* is the same in all mixed Nash equilibria and
  - G_1^* equals the expected gain that player 1 can achieve under a minmax strategy of player 2.

Computing Nash Equilibria in 2-Player, Zero-Sum Games

- Thus, we can construct a linear program that computes player 2's mixed equilibrium strategy:

$$
\begin{array}{rlrr}
\text{minimize} & G_1^* & & (3) \\
\text{subject to} & \displaystyle\sum_{s_k \in S_2} g_1(s_j, s_k) \cdot \pi_2(s_k) \leq G_1^* & \forall s_j \in S_1 & (4) \\
& \displaystyle\sum_{s_k \in S_2} \pi_2(s_k) = 1 & & (5) \\
& \pi_2(s_k) \geq 0 & \forall s_k \in S_2 & (6)
\end{array}
$$

Computing Nash Equilibria in 2-Player, Zero-Sum Games

- *Reversing the roles of players 1 and 2 in the constraints*, we can construct the *dual program of player 2's LP* to obtain a linear program that computes player 1's mixed equilibrium strategy. The objective now is to *maximize* G_1^*:

$$\begin{aligned}
\text{maximize} \quad & G_1^* && && (7)\\
\text{subject to} \quad & \sum_{s_j \in S_1} g_1(s_j, s_k) \cdot \pi_1(s_j) \geq G_1^* && \forall s_k \in S_2 && (8)\\
& \sum_{s_j \in S_1} \pi_1(s_j) = 1 && && (9)\\
& \pi_1(s_j) \geq 0 && \forall s_j \in S_1 && (10)
\end{aligned}$$

Maxmin/Minmax Strategies in Arbitrary 2-Player Games

- The problem of finding a Nash equilibrium in a two-player, general-sum game cannot be formulated as a linear program because:
  - the two players' interests are no longer diametrically opposed;
  - thus, the problem cannot be stated as an optimization problem: one player is not seeking to minimize the other player's gains.
- However, using the minimax theorem, we can reduce the problem of computing maxmin and minmax strategies to the problem of finding a Nash equilibrium in an appropriate two-player, zero-sum game.
- Consider a general-sum game $\mathscr{G}$ in normal form for two players, 1 and 2, with the set $\mathscr{S} = S_1 \times S_2$ of pure-strategy profiles and with gain functions g_1 and g_2.
- How can we compute a maxmin strategy for player 1 in $\mathscr{G}$?

Maxmin/Minmax Strategies in Arbitrary 2-Player Games

- Since player 1's maxmin strategy depends solely on 1's gains, changing player 2's gains does not affect it.
- Define a zero-sum game $\mathscr{G}'$ in normal form for two players, 1 and 2, with the same set $\mathscr{S} = S_1 \times S_2$ of pure-strategy profiles, and with gain functions g_1 (remaining unchanged) and $g_2' \equiv -g_1$ (negating g_1).
- By the minimax theorem, since $\mathscr{G}'$ is zero-sum, whenever player 1's mixed strategy is part of a Nash equilibrium, it is a maxmin strategy for 1 in $\mathscr{G}'$.
- Consequently, player 1's maxmin strategy is the same in $\mathscr{G}$ and $\mathscr{G}'$.
- Using the LP for computing 1's Nash equilibrium strategy in $\mathscr{G}'$, we obtain 1's maxmin strategy in $\mathscr{G}$.
- An analogous approach works for finding minmax strategies.

LCP Formulation for 2-Player, General-Sum Games

- Although the problem of finding a Nash equilibrium in a two-player, general-sum game cannot be formulated as a linear program, it can be formulated as a *linear complementary problem (LCP)*.
- First, by introducing slack variables r_1^j for $1 \leq j \leq \|S_1\|$, we express the LP (3)–(6) (for player 2 in the zero-sum case) as follows:

$$
\begin{array}{rlrr}
\text{minimize} & G_1^* & & (11)\\
\text{subject to} & \displaystyle\sum_{s_k \in S_2} g_1(s_j, s_k) \cdot \pi_2(s_k) + r_1^j = G_1^* & \forall s_j \in S_1 & (12)\\
& \displaystyle\sum_{s_k \in S_2} \pi_2(s_k) = 1 & & (13)\\
& \pi_2(s_k) \geq 0 & \forall s_k \in S_2 & (14)\\
& r_1^j \geq 0 & \text{for } 1 \leq j \leq \|S_1\| & (15)
\end{array}
$$

LCP Formulation for 2-Player, General-Sum Games

- We do the same for player 1.
- Note that in the general-sum case we can no longer find one player's equilibrium strategy by considering only the other player's gains.
  We need to discuss both players simultaneously.
- Note further that we have no longer an optimization problem, but rather a *constraint satisfaction problem*.
  We describe only the constraints, not an objective function.

LCP Formulation for 2-Player, General-Sum Games

- The LCP for computing a Nash equilibrium in two-player, general-sum games then has the form (for player 1 and player 2):

$$
\begin{aligned}
&\sum_{s_k \in S_2} g_1(s_j, s_k) \cdot \pi_2(s_k) + r_1^j = G_1^* && \forall s_j \in S_1 && (16)\\
&\sum_{s_j \in S_1} g_2(s_j, s_k) \cdot \pi_1(s_j) + r_2^k = G_2^* && \forall s_k \in S_2 && (17)\\
&\sum_{s_j \in S_1} \pi_1(s_j) = 1, \quad \sum_{s_k \in S_2} \pi_2(s_k) = 1 && && (18)\\
&\pi_1(s_j) \geq 0, \quad \pi_2(s_k) \geq 0 && \forall s_j \in S_1, \quad \forall s_k \in S_2 && (19)\\
&r_1^j \geq 0, \quad r_2^k \geq 0 && \text{for } 1 \leq j \leq \|S_1\|,\ 1 \leq k \leq \|S_2\| && (20)\\
&r_1^j \cdot \pi_1(s_j) = 0, \quad r_2^k \cdot \pi_2(s_k) = 0 && \forall s_j \in S_1, \quad \forall s_k \in S_2 && (21)
\end{aligned}
$$

LCP Formulation for 2-Player, General-Sum Games

- Constraint (21) is called the *complementary condition* and ensures that G_1^* and G_2^* cannot take unboundedly large values.
- The best-known algorithm to solve this LCP is the LEMKE–HOWSON *Algorithm* (which will not be presented here).
  - Advantages:
    - LEMKE–HOWSON is guaranteed to find a Nash equilibrium.
    - Its constructive nature provides an alternative proof of Nash's theorem.
    - It can be used to find more than one Nash equilibrium.
  - Disadvantages:
    - LEMKE–HOWSON is not guaranteed to find *all* Nash equilibria.
    - It is not even possible to decide efficiently whether all Nash equilibria have been found.
    - It is provably *exponential-time in the worst case*.
    - Since there is no objective function, we don't know how close we are to a solution before we actually find one.

Related Problems for 2-Player, General-Sum Games

The following problems refer to general-sum games $\mathscr{G}$ in normal form:

- **Uniqueness**: Given $\mathscr{G}$, does there exist a unique Nash equilibrium in $\mathscr{G}$?
- **Pareto Optimality**: Given $\mathscr{G}$, does there exist a Pareto-optimal Nash equilibrium in $\mathscr{G}$?
- **Guaranteed Payoff**: Given $\mathscr{G}$ and a value k, does there exist a Nash equilibrium in $\mathscr{G}$ such that some player i has expected gain of at least k?
- **Guaranteed Social Welfare**: Given $\mathscr{G}$ and a value k, does there exist a Nash equilibrium in $\mathscr{G}$ such that the sum of all players expected gains is at least k?

Related Problems for 2-Player, General-Sum Games

- **Action Inclusion**: Given $\mathscr{G}$ and a pure strategy $s_j \in S_i$ for some player i, does there exist a Nash equilibrium in $\mathscr{G}$ such that i plays s_j with positive probability?
- **Action Exclusion**: Given $\mathscr{G}$ and a pure strategy $s_j \in S_i$ for some player i, does there exist a Nash equilibrium in $\mathscr{G}$ such that i plays s_j with zero probability?

Related Problems for 2-Player, General-Sum Games

Theorem

Each of the six problems just defined is NP*-hard, even when restricted to two-player games.* **without proof**

Theorem

Computing all the Nash equilibria in a given two-player, general-sum game in normal form requires worst-case time that is exponential in the number of pure strategies of each player. **without proof**

Nash Equilibria in General-Sum Normal Form Games

- In general-sum normal form games with more than two players, the problem of finding Nash equilibria can no longer be represented even as an LCP, but only as a *nonlinear complementary problem*.
- These, however, are hopelessly impractical to solve exactly.
- **How can we formulate the problem as an input to an algorithm?**

1. Approximate the solution via a *sequence of linear complementary problems (SLCP)*:
    - Each LCP provides an approximation, and its solution is used to get the next LCP in the sequence.
    - This generalizes *Newton's method* of approximating local maxima of quadratic equations.
    - Although it is not globally convergent, in practice it is often useful.

Nash Equilibria in General-Sum Normal Form Games

2. Formulate the problem as a minimum of a function:
   - Starting from a mixed-strategy profile $\vec{\pi}$, define:

$$\begin{aligned} c_i^j(\vec{\pi}) &= G_i(\vec{\pi}_{-i}, s_j) - G_i(\vec{\pi}) \quad \text{for each } s_j \in S_i \\ d_i^j(\vec{\pi}) &= \max(c_i^j(\vec{\pi}), 0) \end{aligned}$$

   - Define the following optimization problem with objective function f:

$$\begin{array}{llll} \text{minimize} & f(\vec{\pi}) = \sum_{1 \leq i \leq n} \sum_{s_j \in S_i} \left(d_i^j(\vec{\pi}) \right)^2 & & (22) \\ \text{subject to} & \sum_{s_j \in S_i} \pi_i(s_j) = 1 & 1 \leq i \leq n & (23) \\ & \pi_i(s_j) \geq 0 & 1 \leq i \leq n, \quad \forall s_j \in S_i & (24) \end{array}$$

   - **Advantage: Flexibility!**

Nash Equilibria in General-Sum Normal Form Games

3. Enroll the constraints into the objective function, denoted by $g(\vec{\pi})$, and apply an unconstrained optimization method:

$$\text{minimize} \qquad g(\vec{\pi}) = \sum_{1 \leq i \leq n} \sum_{s_j \in S_i} \left(d_i^j(\vec{\pi}) \right)^2 \tag{25}$$

$$+ \sum_{1 \leq i \leq n} \left(1 - \sum_{s_j \in S_i} \pi_i(s_j) \right)^2 \tag{26}$$

$$+ \sum_{1 \leq i \leq n} \sum_{s_j \in S_i} (\min(\pi_i(s_j), 0))^2 \tag{27}$$

- Disadvantage of minimizing f and g: Both optimization problems have local minima that do not correspond to Nash equilibria.

Global convergence is thus a problem.

Hardness of Computing Nash Equilibria: Issues

- In general-sum normal form games that are not restricted to two players, Nash equilibria appear to be not efficiently computable.
- This conjecture has recently been shown by Daskalakis, Goldberg, and Papadimitriou (2006).
- Unlike the decision problems for discrete structures, we are here concerned with
  - a functional search problem
  - defined over continuous structures.

  For example, the gain functions of the players may have irrational values, which cannot exactly be handled by a computer but can only be approximated.

Hardness of Computing Nash Equilibria: Issues

- This problem can be sidestepped by searching not for an exact Nash equilibrium $\vec{\pi}$, but for merely an *ε-approximation $\vec{\pi}_\varepsilon$ of $\vec{\pi}$*: The expected gain of each player in the strategy profile $\vec{\pi}_\varepsilon$ is by at most the factor ε better than her expected gain in the strategy profile $\vec{\pi}$.
- Such an ε-approximation of a Nash equilibrium $\vec{\pi}$ thus allows the players to deviate from their mixed equilibrium strategies so as to improve their gains, but by no more than the factor $\varepsilon > 0$, which can be chosen arbitrarily small.
- This approach is justified: Computing an ε-approximation of a Nash equilibrium is no harder than computing this Nash equilibrium itself (since even an ε-error is allowed): Hardness of ε-approximations of Nash equilibria transfers to hardness of computing them exactly.

Hardness of Computing Nash Equilibria: Issues

- The next problem: Classical complexity theory is actually not applicable here, since there *always* exists a solution of the problem; the corresponding decision problem thus is trivial to solve.
- **How can one show the computational hardness of such search problems that have a solution for all instances?**
- To illustrate the approach to circumventing this difficulty, consider:

EQUAL-SUBSETS

Given: A sequence $a_1, \ldots, a_n$ of positive integers such that $\sum_{i=1}^{n} a_i \leq 2^n - 2$.

Find: Nonempty, disjoint subsets I, J of $\{1, \ldots, n\}$, such that

$$\sum_{i \in I} a_i = \sum_{j \in J} a_j.$$

Hardness of Computing Nash Equilibria: Issues

- Note that there always exists a solution for EQUAL-SUBSETS because
  1. there are exactly $2^n - 1$ nonempty subsets I of $\{1\ldots,n\}$, and
  2. for each nonempty subset I of $\{1,\ldots,n\}$, the value $\sum_{i\in I} a_i$ has one out of $2^n - 2$ values (these values come from the set $\{1,2,\ldots,2^n-2\}$).
- Thus, by the pigeonhole principle, there must be two distinct subsets, $I', J' \subset \{1,\ldots n\}$, such that

$$\sum_{i\in I'} a_i = \sum_{j\in J'} a_j.$$

- If we let $I = I' \smallsetminus (I' \cap J')$ and $J = J' \smallsetminus (I' \cap J')$, then we get our two disjoint sets that have the same sums.
  (It must be the case that I and J are nonempty because I' and J' are distinct and the numbers $a_1,\ldots,a_n$ are positive.)

Hardness of Computing Nash Equilibria: Issues

- If we would have defined EQUAL-SUBSETS as a decision problem where we ask about the existence of the sets I and J then, of course, each problem instance would have trivially the answer "yes."
- Still, computing the two sets that witness this "yes" answer intutively seems to be difficult, as the proof that there always is a "yes" answer relies on the inherently *nonconstructive* pigeonhole principle.
- The **pigeonhole principle** says that if we have some m objects (for EQUAL-SUBSETS: the $2^n - 1$ subsets) and each of them has one of $m-1$ "features" (for EQUAL-SUBSETS: the $2^n - 2$ sums), then there must be at least two objects that have the same "feature."
- However, the pigeonhole principle gives no hint whatsoever as to how to find these two objects.

Four Types of Nonconstructive Proof Steps

- Papadimitriou (1994) introduced a new complexity class, PPP, which stands for "**P***olynomial* **P***igeonhole* **P***rinciple*."
- Computing a Nash equilibrium in mixed strategies is, in fact, a problem in PPP.
- However, the class PPAD ("**P***olynomial* **P***arity* **A***rgument for* **D***irected graphs*") captures the complexity of this problem more precisely.
- Intuitively, for such total search problems in PPP or PPAD that are solvable for each instance:
  - there must exist a mathematical proof of this fact and
  - if the problem is not solvable in polynomial time, then there must be a *nonconstructive* step in that proof.

Four Types of Nonconstructive Proof Steps: PPP

For all known total search problems that appear to be not polynomial-time solvable, Papadimitriou (1994) identifies one of the following four simple reasons for such a nonconstructive proof step:

1. PPP, for "**P***olynomial* **P***igeonhole* **P***rinciple*."

   For each problem in PPP, the nonconstructive proof step can be described by the following "pigeonhole" argument:

   Every function mapping n elements to $n-1$ elements has a collision, i.e., $f(i) = f(j)$ for $i \neq j$.

Four Types of Nonconstructive Proof Steps: PLS

2. PLS, for "**P***olynomial* **L***ocal* **S***earch*."

   For each problem in PLS, the nonconstructive proof step can be described by the following argument:

   Every directed acyclic graph has a sink, i.e., a vertex without any outgoing edges.

Four Types of Nonconstructive Proof Steps: PPA

3. PPA, for "**P***olynomial* **P***arity* **A***rgument for graphs*."

   For each problem in PPA, the nonconstructive proof step can be described by the following parity argument:

   If an undirected graph has a vertex of odd degree,
   then it has at least one other such vertex.

   Here, the *degree of a vertex* in an undirected graph is the number of edges incident with this vertex.

Four Types of Nonconstructive Proof Steps: PPAD

4. PPAD, for "**P***olynomial* **P***arity* **A***rgument for* **D***irected graphs*."

   For each problem in PPAD, the nonconstructive proof step can be described by the following parity argument:

   If a directed graph has an unbalanced vertex
   (i.e., a vertex with distinct indegree and outdegree),
   then it has at least one other such vertex.

   Here, the *indegree of a vertex* in a directed graph is the number of incoming edges and its *outdegree* is the number of outgoing edges.

The Polynomial Pigeonhole Principle

PIGEONHOLE-FUNCTION

Given: A function f, $f\colon \{0,1\}^n \to \{0,1\}^n$, expressed as an algorithm computing f in time linear in the length of the encoding of the algorithm.

Find: An input $x \in \{0,1\}^n$ such that $f(x) = 0^n$, or two distinct inputs $x, x' \in \{0,1\}^n$ such that $f(x) = f(x')$.

Observations:

1. The function f can essentially be any polynomial-time computable function because every reasonable way of encoding it allows to extend the algorithm for computing f by a polynomial factor.

2. There always is a solution for PIGEONHOLE-FUNCTION: If there is no input x such that $f(x) = 0^n$ then, by the pigeonhole principle, there must be two distinct inputs, x and x', such that $f(x) = f(x')$.

PPP: Polynomial Pigeonhole Principle

Define PPP as the class of all problems that can be reduced to PIGEONHOLE-FUNCTION using the following notion of reducibility:
Let $F, G : \Sigma^* \to \Sigma^*$ be two total functions. We say *F functionally (many-one-)reduces in polynomial time to G* if there exist polynomial-time computable functions r and s such that for all $x \in \Sigma^*$,

$$F(x) = s(G(r(x))).$$

1. The efficiently computable function r transforms a given instance x of search problem F into an equivalent instance of search problem G.
2. The efficiently computable function s transforms a solution of this instance $r(x)$ of search problem G back into an equivalent solution of the given instance x of search problem F.

PLS: Polynomial Local Search

PLS is the class of problems reducible to

LOCAL-OPTIMUM	
Given:	Two functions, $f\colon \{0,1\}^n \to \{0,1\}^n$ and $p\colon \{0,1\}^n \to \{0,\dots,2^n-1\}$, expressed as algorithms computing these functions in time linear in the lengths of their encodings.
Find:	An input x such that $p(f(x)) \geq p(x)$.

Observations:

1. Since for each argument, function p takes one of finitely many values, there always exists a solution x such that $p(f(x)) \geq p(x)$.
2. Intuitively, LOCAL-OPTIMUM captures the idea of a heuristic local search: We compute $x, f(x), f(f(x)), \dots$, until we find an $x' = f(\cdots f(x) \cdots)$ with $p(f(x')) \geq p(x')$, so x' is a local minimum of p wrt. local search function f.

PPA: Polynomial Parity Argument for Graphs

- Let f be some function that on input $x \in \{0,1\}^n$ outputs a set of zero, one, or two elements from $\{0,1\}^n$. f defines an undirected graph $U(f)$: Each $x \in \{0,1\}^n$ is a vertex and there is an edge connecting the vertices x and x' exactly if $x' \in f(x)$ and $x \in f(x')$.
- Note that each vertex in $U(f)$ has degree at most two.

PPA is the class of problems reducible to

Odd-Degree-Vertex

Given:	A function $f : \{0,1\}^n \rightarrow \{\emptyset\} \cup \{0,1\}^n \cup \{0,1\}^{2n}$, represented by an algorithm computable in time linear in the length of its encoding.
Find:	If vertex 0^n has odd degree in $U(f)$, then find another vertex of odd degree in $U(f)$.

PPAD: Polynomial Parity Argument for Directed Graphs

We represent directed graphs with indegrees and outdegrees at most one:

- Let s and p be two functions from $\{0,1\}^n$ to $\{0,1\}^n$.
- We define the graph $G(s,p)$ to have vertex set $\{0,1\}^n$ and the following edges: For each $x, x' \in \{0,1\}^n$, there is a directed edge from x to x' if and only if $s(x) = x'$ and $p(x') = x$.
- Intuitively,
  - s is the *successor* function, i.e., $s(x)$ is the vertex we can move to from vertex x (provided that $p(s(x)) = x$),
  - and p is the *predecessor* function, i.e., $p(x)$ is the vertex from which we could reach x (provided that $s(p(x)) = x$).

PPAD: Polynomial Parity Argument for Directed Graphs

PPAD is the class of problems reducible to

END-OF-THE-LINE

Given: Two functions, $s\colon \{0,1\}^n \rightarrow \{0,1\}^n$ and $p\colon \{0,1\}^n \rightarrow \{0,1\}^n$, represented by algorithms with linear running times with respect to the lengths of their encodings.

Find: If in graph $G(s,p)$ vertex 0^n has indegree zero, then find a vertex x with

- either the outdegree equal to zero
- or the indegree equal to zero (in the latter case, this x must be different from 0^n).

Relations Among These Complexity Classes

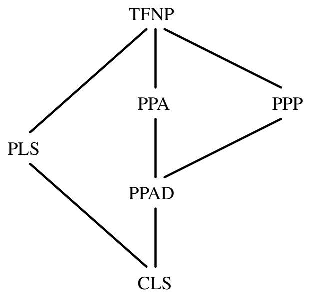

Figure: The structure of the subclasses of TFNP

- CLS: "*continuous local search*"
- TFNP: "*total* NP *functions*"

Nash-Equilibrium is PPAD-Complete

Nash-Equilibrium	
Given:	A cooperative game in normal form, discretely represented in terms of an ε-approximation for an arbitrarily small positive constant ε.
Find:	An ε-approximation of a Nash equilibrium in mixed strategies.

- A total search problem is said to be PPAD-*hard* if End-of-the-Line (and thus any problem of the class PPAD) can be reduced to it, and
- it is PPAD-*complete* if it belongs to PPAD and is PPAD-hard.

Theorem (Daskalakis, Goldberg, and Papadimitriou (2006))

Nash-Equilibrium *is* PPAD-*complete.*

NASH-EQUILIBRIUM is PPAD-Complete: Proof Sketch

1. That NASH EQUILIBRIUM belongs to PPAD is shown by reducing NASH-EQUILIBRIUM to END-OF-THE-LINE.
2. That NASH-EQUILIBRIUM is PPAD-hard is shown by reducing END-OF-THE-LINE to NASH-EQUILIBRIUM.

We will only sketch the proof of the former reduction, which uses:

Lemma (Sperner's Lemma)

Let $T_n = \vec{x}_0 \cdots \vec{x}_n$ be a simplicially subdivided n-simplex and let $\mathscr{L}$ be a proper labeling of the subdivision of T_n. There are an odd number of subsimplexes that are completely labeled by $\mathscr{L}$ in this subdivision of T_n.

Nash-Equilibrium is PPAD-Complete: Proof Sketch

Recall the idea behind the proof of Sperner's lemma:

- Construct walks over the n-subsimplexes, using the fact that on one of the faces of the simplex, there must be an n-subsimplex T'_n such that
  1. the face that T'_n shares with the main simplex is labeled with $0, 1, \dots, n-1$, and
  2. the remaining vertex v is labeled either with n (in which case we have found a completely labeled n-subsimplex), or with a label from set $\{0, \dots, n-1\}$.
- In the latter case, there is a unique face that includes v and is labeled with $0, \dots, n-1$.
- We cross over this face to the next n-subsimplex and repeat the process.

NASH-EQUILIBRIUM is PPAD-Complete: Proof Sketch

- Sperner's lemma guarantees that there is a starting *n*-subsimplex that, through this process, leads to finding a completely labeled *n*-subsimplex.
- However, note that the proof of Sperner's lemma is not constructive!
- It says that there is a good starting *n*-subsimplex that will lead to the completely labeled *n*-subsimplex, but some of the promising starting subsimplexes can, just as well, lead us "through the simplex" and then "out of it."

Nash-Equilibrium is PPAD-Complete: Proof Sketch

- These walks correspond to the “indegree/outdegree bounded by one” graphs necessary for End-of-the-Line.
- The subsimplexes correspond to the vertices of the graph.
- The only missing piece of the puzzle is finding a good starting point.
- **Idea:** Make the “in and out” walks work for us, instead of against us! That is, extend the space available to our walks by attaching a number of additional *n*-simplexes at the face through which we used to enter.

Nash-Equilibrium is PPAD-Complete: Proof Sketch

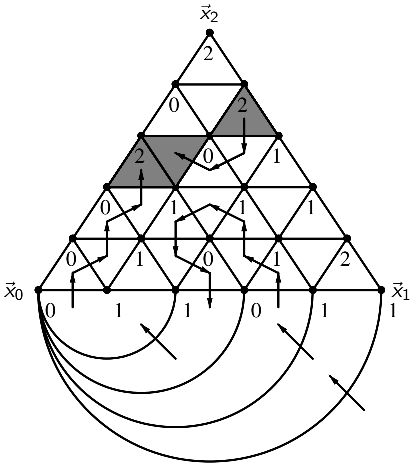

Figure: Reducing Nash-Equilibrium to End-of-the-Line

NASH-EQUILIBRIUM is PPAD-Complete: Proof Sketch

- Now, the starting point is simply the outermost added subsimplex.
- Following the walk from this point (using exactly the same rules as before, in the proof of Sperner's lemma), we will eventually reach the completely labeled n-subsimplex, possibly winding in and out of the main simplex several times, but eventually reaching our goal.
- This gives the outline of how to use Sperner's lemma to reduce NASH-EQUILIBRIUM to END-OF-THE-LINE.
- We omit the technical details of actually encoding the simplex and the underlying graph in the format required by END-OF-THE-LINE.

Further Algorithmic Results on Nash Equilibria

- The result of Daskalakis, Goldberg, and Papadimitriou (2006) is a milestone in algorithmic game theory.
- Chen and Deng (2006) showed that NASH-EQUILIBRIUM remains PPAD-complete even if restricted to games with *two* players.
- Many more results of this kind have since been shown, such as:
  - Conitzer and Sandholm (2006): A technique for reducing normal-form games to compute a Nash equilibrium.
  - Elkind, Goldberg, Goldberg, and Wooldridge (2007): Computing good Nash equilibria in graphical games.
  - Brandt, Fischer, and Holzer (2011): Equilibria of graphical games with symmetries.