Algorithmic game theory

Martin Balko

5th lecture

November 1st 2024

Nash equilibria in bimatrix games

What have we learned so far

What have we learned so far

- We have seen three algorithms to find NE in bimatrix games:

What have we learned so far

- We have seen three algorithms to find NE in bimatrix games:
  - the brute-force algorithm with support enumeration,

What have we learned so far

- We have seen three algorithms to find NE in bimatrix games:
  - the brute-force algorithm with support enumeration,
  - the algorithm with vertex enumeration,

What have we learned so far

- We have seen three algorithms to find NE in bimatrix games:
  - the brute-force algorithm with support enumeration,
  - the algorithm with vertex enumeration,
  - the Lemke–Howson algorithm.

What have we learned so far

- We have seen three algorithms to find NE in bimatrix games:
  - the brute-force algorithm with support enumeration,
  - the algorithm with vertex enumeration,
  - the Lemke–Howson algorithm.
- All these algorithms have exponential running time in the worst case.

What have we learned so far

- We have seen three algorithms to find NE in bimatrix games:
  - the brute-force algorithm with support enumeration,
  - the algorithm with vertex enumeration,
  - the Lemke–Howson algorithm.
- All these algorithms have exponential running time in the worst case.

Source: https://www.shutterstock.com/

What have we learned so far

- We have seen three algorithms to find NE in bimatrix games:
  - the brute-force algorithm with support enumeration,
  - the algorithm with vertex enumeration,
  - the Lemke–Howson algorithm.
- All these algorithms have exponential running time in the worst case.

Source: https://www.shutterstock.com/

- Is there a chance to get an efficient algorithm?

What have we learned so far

- We have seen three algorithms to find NE in bimatrix games:
  - the brute-force algorithm with support enumeration,
  - the algorithm with vertex enumeration,
  - the Lemke–Howson algorithm.
- All these algorithms have exponential running time in the worst case.

Source: https://www.shutterstock.com/

- Is there a chance to get an efficient algorithm?
- NASH = the problem of finding NE in bimatrix games.

What have we learned so far

- We have seen three algorithms to find NE in bimatrix games:
  - the brute-force algorithm with support enumeration,
  - the algorithm with vertex enumeration,
  - the Lemke–Howson algorithm.
- All these algorithms have exponential running time in the worst case.

Source: https://www.shutterstock.com/

- Is there a chance to get an efficient algorithm?
- NASH = the problem of finding NE in bimatrix games.
- Today, we discuss the computational complexity of NASH.

Where does NASH belong to?

Where does NASH belong to?

- Is NASH NP-complete?

Where does NASH belong to?

- Is NASH NP-complete?
  - No.

Where does NASH belong to?

- Is NASH NP-complete?
  - No. NP is a class of decision problems (yes/no answers)

Where does NASH belong to?

- Is NASH NP-complete?
  - No. NP is a class of decision problems (yes/no answers) while NE always exist (so the answer is always yes).

Where does NASH belong to?

- Is NASH NP-complete?
  - No. NP is a class of decision problems (yes/no answers) while NE always exist (so the answer is always yes).
- Another candidate is the complexity class FNP

Where does NASH belong to?

- Is NASH NP-complete?
  - No. NP is a class of decision problems (yes/no answers) while NE always exist (so the answer is always yes).
- Another candidate is the complexity class FNP ("functional NP").

Where does NASH belong to?

- Is NASH NP-complete?
  - No. NP is a class of decision problems (yes/no answers) while NE always exist (so the answer is always yes).
- Another candidate is the complexity class FNP ("functional NP").
  - The input of FNP problem is an instance of a problem from NP.

Where does NASH belong to?

- Is NASH NP-complete?
  - No. NP is a class of decision problems (yes/no answers) while NE always exist (so the answer is always yes).
- Another candidate is the complexity class FNP ("functional NP").
  - The input of FNP problem is an instance of a problem from NP. The algorithm outputs a solution if one exists.

Where does NASH belong to?

- Is NASH NP-complete?
  - No. NP is a class of decision problems (yes/no answers) while NE always exist (so the answer is always yes).
- Another candidate is the complexity class FNP ("functional NP").
  - The input of FNP problem is an instance of a problem from NP. The algorithm outputs a solution if one exists. If there is no solution, the algorithm outputs 'no'.

Where does NASH belong to?

- Is NASH NP-complete?
  - No. NP is a class of decision problems (yes/no answers) while NE always exist (so the answer is always yes).
- Another candidate is the complexity class FNP ("functional NP").
  - The input of FNP problem is an instance of a problem from NP. The algorithm outputs a solution if one exists. If there is no solution, the algorithm outputs 'no'.
  - That is, we demand a solution for 'yes' instances.

Where does NASH belong to?

- Is NASH NP-complete?
  - No. NP is a class of decision problems (yes/no answers) while NE always exist (so the answer is always yes).
- Another candidate is the complexity class FNP ("functional NP").
  - The input of FNP problem is an instance of a problem from NP. The algorithm outputs a solution if one exists. If there is no solution, the algorithm outputs 'no'.
  - That is, we demand a solution for 'yes' instances.
  - NASH belongs to FNP,

Where does NASH belong to?

- Is NASH NP-complete?
  - No. NP is a class of decision problems (yes/no answers) while NE always exist (so the answer is always yes).
- Another candidate is the complexity class FNP ("functional NP").
  - The input of FNP problem is an instance of a problem from NP. The algorithm outputs a solution if one exists. If there is no solution, the algorithm outputs 'no'.
  - That is, we demand a solution for 'yes' instances.
  - NASH belongs to FNP, as checking whether a strategy profile is NE can be done using the Best Response Condition.

Where does NASH belong to?

- Is NASH NP-complete?
  - No. NP is a class of decision problems (yes/no answers) while NE always exist (so the answer is always yes).
- Another candidate is the complexity class FNP ("functional NP").
  - The input of FNP problem is an instance of a problem from NP. The algorithm outputs a solution if one exists. If there is no solution, the algorithm outputs 'no'.
  - That is, we demand a solution for 'yes' instances.
  - NASH belongs to FNP, as checking whether a strategy profile is NE can be done using the Best Response Condition.
  - Is NASH FNP-complete?

Where does NASH belong to?

- Is NASH NP-complete?
    - No. NP is a class of decision problems (yes/no answers) while NE always exist (so the answer is always yes).
- Another candidate is the complexity class FNP ("functional NP").
    - The input of FNP problem is an instance of a problem from NP. The algorithm outputs a solution if one exists. If there is no solution, the algorithm outputs 'no'.
    - That is, we demand a solution for 'yes' instances.
    - NASH belongs to FNP, as checking whether a strategy profile is NE can be done using the Best Response Condition.
    - Is NASH FNP-complete? Unlikely, because of the following result.

Where does NASH belong to?

- Is NASH NP-complete?
  - No. NP is a class of decision problems (yes/no answers) while NE always exist (so the answer is always yes).
- Another candidate is the complexity class FNP ("functional NP").
  - The input of FNP problem is an instance of a problem from NP. The algorithm outputs a solution if one exists. If there is no solution, the algorithm outputs 'no'.
  - That is, we demand a solution for 'yes' instances.
  - NASH belongs to FNP, as checking whether a strategy profile is NE can be done using the Best Response Condition.
  - Is NASH FNP-complete? Unlikely, because of the following result.

Theorem 2.34 (Megiddo and Papadimitriou, 1991)

If the problem NASH is FNP-complete, then NP $=$ coNP.

Where does NASH belong to?

- Is NASH NP-complete?
  - No. NP is a class of decision problems (yes/no answers) while NE always exist (so the answer is always yes).
- Another candidate is the complexity class FNP ("functional NP").
  - The input of FNP problem is an instance of a problem from NP. The algorithm outputs a solution if one exists. If there is no solution, the algorithm outputs 'no'.
  - That is, we demand a solution for 'yes' instances.
  - NASH belongs to FNP, as checking whether a strategy profile is NE can be done using the Best Response Condition.
  - Is NASH FNP-complete? Unlikely, because of the following result.

Theorem 2.34 (Megiddo and Papadimitriou, 1991)

If the problem NASH is FNP-complete, then NP = coNP.

- Without proof (but you can find it in the lecture notes).

New complexity class

New complexity class

- The proof of the correctness of the Lemke–Howson algorithm reveals the structure of NASH

New complexity class

- The proof of the correctness of the Lemke–Howson algorithm reveals the structure of NASH (finding another endpoint of a path in graph of maximum degree 2).

New complexity class

- The proof of the correctness of the Lemke–Howson algorithm reveals the structure of NASH (finding another endpoint of a path in graph of maximum degree 2).
- Let us capture this abstract structure.

New complexity class

- The proof of the correctness of the Lemke–Howson algorithm reveals the structure of NASH (finding another endpoint of a path in graph of maximum degree 2).
- Let us capture this abstract structure.

- The END-OF-THE-LINE problem:

New complexity class

- The proof of the correctness of the Lemke–Howson algorithm reveals the structure of NASH (finding another endpoint of a path in graph of maximum degree 2).
- Let us capture this abstract structure.

- The END-OF-THE-LINE problem: for a directed graph G with every vertex having at most one predecessor and one successor, given a vertex s of G with no predecessor, find a vertex $t \neq s$ with no predecessor or no successor.

New complexity class

- The proof of the correctness of the Lemke–Howson algorithm reveals the structure of NASH (finding another endpoint of a path in graph of maximum degree 2).
- Let us capture this abstract structure.

- The END-OF-THE-LINE problem: for a directed graph G with every vertex having at most one predecessor and one successor, given a vertex s of G with no predecessor, find a vertex $t \neq s$ with no predecessor or no successor. The graph G is not given on the input, but it is specified by some polynomial-time computable function $f(v)$ that returns the predecessor and successor (if they exist) of v.

New complexity class

- The proof of the correctness of the Lemke–Howson algorithm reveals the structure of NASH (finding another endpoint of a path in graph of maximum degree 2).
- Let us capture this abstract structure.

- The END-OF-THE-LINE problem: for a directed graph G with every vertex having at most one predecessor and one successor, given a vertex s of G with no predecessor, find a vertex $t \neq s$ with no predecessor or no successor. The graph G is not given on the input, but it is specified by some polynomial-time computable function $f(v)$ that returns the predecessor and successor (if they exist) of v.
    - Thus, G can be exponentially large with respect to the input.

New complexity class

- The proof of the correctness of the Lemke–Howson algorithm reveals the structure of NASH (finding another endpoint of a path in graph of maximum degree 2).
- Let us capture this abstract structure.

- The END-OF-THE-LINE problem: for a directed graph G with every vertex having at most one predecessor and one successor, given a vertex s of G with no predecessor, find a vertex $t \neq s$ with no predecessor or no successor. The graph G is not given on the input, but it is specified by some polynomial-time computable function $f(v)$ that returns the predecessor and successor (if they exist) of v.
  - Thus, G can be exponentially large with respect to the input.

- Let PPAD be a complexity class consisting of problems that admit a polynomial-time reduction to END-OF-THE-LINE.

The class PPAD

The class PPAD

- The class PPAD was introduced in 1994 by Papadimitrou.

The class PPAD

- The class PPAD was introduced in 1994 by Papadimitrou.

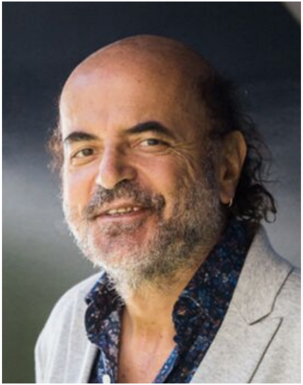


Figure: Christos Papadimitriou (born 1949).

Source: https://cs.columbia.edu

The class PPAD

- The class PPAD was introduced in 1994 by Papadimitrou.

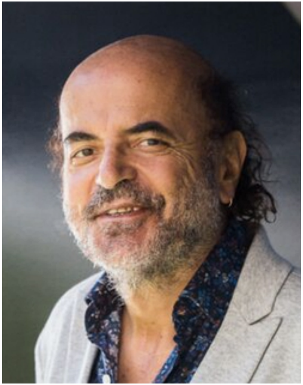


Figure: Christos Papadimitriou (born 1949).

Source: https://cs.columbia.edu

- Abbreviation for "Polynomial Parity Arguments on Directed graphs".

The class PPAD

- The class PPAD was introduced in 1994 by Papadimitrou.

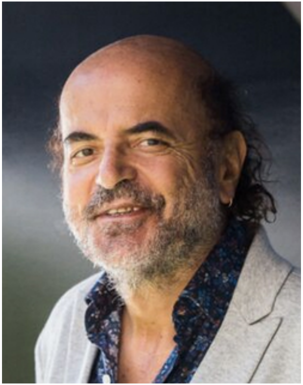


Figure: Christos Papadimitriou (born 1949).

Source: https://cs.columbia.edu

- Abbreviation for "Polynomial Parity Arguments on Directed graphs".
- This complexity class contains a lot of well-known problems.

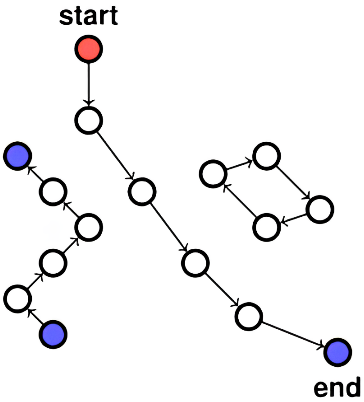
Problems from PPAD: End-of-the-line

Problems from PPAD: End-of-the-line

- For an oriented graph G with max. indegree and outdegree 1 and a source in G, find a target in G. The graph is given by a polynomial-time computable function $f(v)$ that returns predecessor and successor of v.

Problems from PPAD: End-of-the-line

- For an oriented graph G with max. indegree and outdegree 1 and a source in G, find a target in G. The graph is given by a polynomial-time computable function $f(v)$ that returns predecessor and successor of v.



Source: R. Savani "Polymatrix Games" Tutorial at WINE 2015

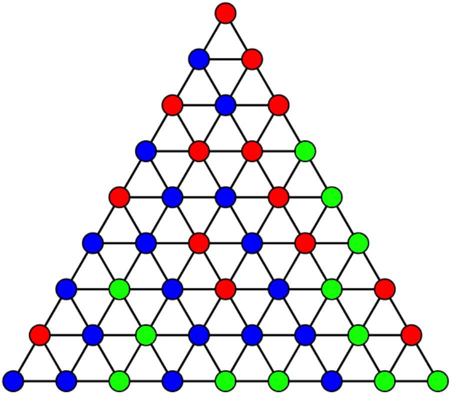
Problems from PPAD: Sperner's lemma

Problems from PPAD: Sperner's lemma

- Given a legal 3-coloring of a triangulated triangle, find a triangle with vertices colored by all 3 colors.

Problems from PPAD: Sperner's lemma

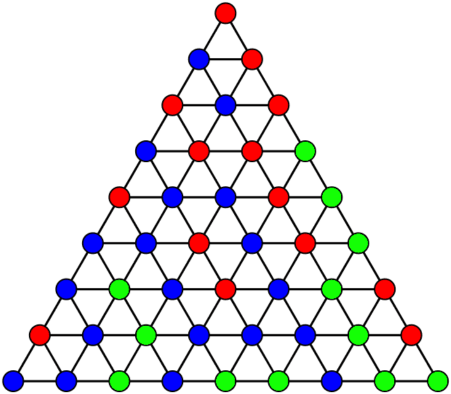
- Given a legal 3-coloring of a triangulated triangle, find a triangle with vertices colored by all 3 colors.



Source: https://lesswrong.com

Problems from PPAD: Sperner's lemma

- Given a legal 3-coloring of a triangulated triangle, find a triangle with vertices colored by all 3 colors.



Source: https://lesswrong.com

- Discrete version of the Brouwer's fixed point theorem.

Problems from PPAD: Brouwer's fixed point theorem

Problems from PPAD: Brouwer's fixed point theorem

- An approximate version of the following theorem is in PPAD: For each $d \in \mathbb{N}$, a non-empty compact convex set K in $\mathbb{R}^d$, and a continuous mapping $f : K \to K$, there exists $x_0 \in K$ such that $f(x_0) = x_0$.

Figure: L. E. J. Brouwer (1881–1966).

Source: https://arxiv.org/pdf/1612.06820.pdf

Problems from PPAD: Ham sandwich theorem

Source: https://www.seekpng.com/

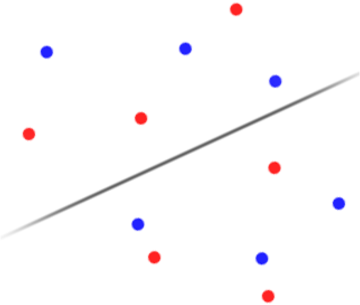
Problems from PPAD: Ham sandwich theorem

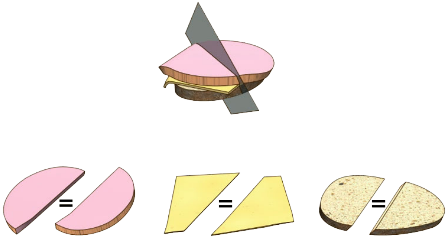
Problems from PPAD: Ham sandwich theorem

- Given n sets of $2n$ points in $\mathbb{R}^n$, find a hyperplane H that contains exactly n points from each of the sets in each open halfspace determined by H.

Problems from PPAD: Ham sandwich theorem

- Given n sets of $2n$ points in $\mathbb{R}^n$, find a hyperplane H that contains exactly n points from each of the sets in each open halfspace determined by H.





Sources: https://ejarzo.github.io and https://curiosamathematica.tumblr.com

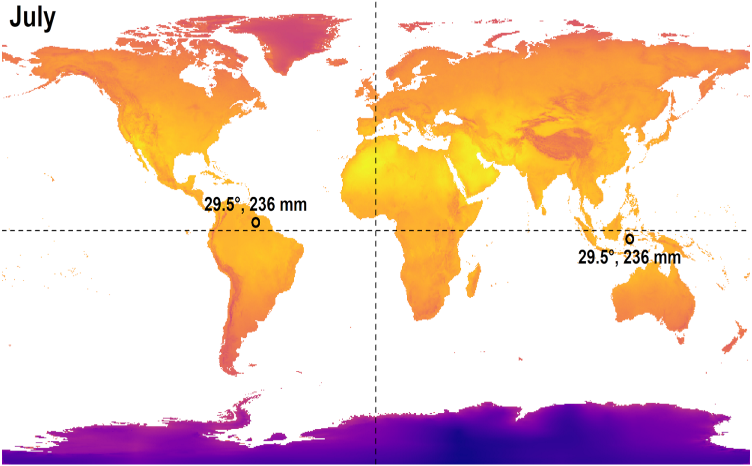
Problems from PPAD: The Borsuk–Ulam theorem

Problems from PPAD: The Borsuk–Ulam theorem

- An approximate version of the following theorem is in PPAD: For every continuous $f : S^n \to \mathbb{R}^n$ there is $x \in S^n$ with $f(x) = f(-x)$.

Problems from PPAD: The Borsuk–Ulam theorem

- An approximate version of the following theorem is in PPAD: For every continuous $f : S^n \to \mathbb{R}^n$ there is $x \in S^n$ with $f(x) = f(-x)$.



Source: https://scientificgems.wordpress.com/

NASH and PPAD

NASH and PPAD

- The proof of the correctness of the Lemke–Howson algorithm shows that NASH belongs to PPAD (for nondegenerate games).

NASH and PPAD

- The proof of the correctness of the Lemke–Howson algorithm shows that NASH belongs to PPAD (for nondegenerate games).
- Is NASH PPAD-complete?

NASH and PPAD

- The proof of the correctness of the Lemke–Howson algorithm shows that NASH belongs to PPAD (for nondegenerate games).
- Is NASH PPAD-complete?
  - That is, is it among the most difficult problems in this class?

NASH and PPAD

- The proof of the correctness of the Lemke–Howson algorithm shows that NASH belongs to PPAD (for nondegenerate games).
- Is NASH PPAD-complete?
  - That is, is it among the most difficult problems in this class?
  - PPAD-completeness gives some evidence of computational intractability, although somehow weaker than NP-completeness.

NASH and PPAD

- The proof of the correctness of the Lemke–Howson algorithm shows that NASH belongs to PPAD (for nondegenerate games).
- Is NASH PPAD-complete?
  - That is, is it among the most difficult problems in this class?
  - PPAD-completeness gives some evidence of computational intractability, although somehow weaker than NP-completeness.
  - Open for a long time.

NASH and PPAD

- The proof of the correctness of the Lemke–Howson algorithm shows that NASH belongs to PPAD (for nondegenerate games).
- Is NASH PPAD-complete?
  - That is, is it among the most difficult problems in this class?
  - PPAD-completeness gives some evidence of computational intractability, although somehow weaker than NP-completeness.
  - Open for a long time.

Theorem 2.35 (Chen, Deng, and Teng and Daskalakis, Goldberg, and Papadimitriou, 2009)

The problem NASH is PPAD-complete.

NASH and PPAD

- The proof of the correctness of the Lemke–Howson algorithm shows that NASH belongs to PPAD (for nondegenerate games).
- Is NASH PPAD-complete?
  - That is, is it among the most difficult problems in this class?
  - PPAD-completeness gives some evidence of computational intractability, although somehow weaker than NP-completeness.
  - Open for a long time.

Theorem 2.35 (Chen, Deng, and Teng and Daskalakis, Goldberg, and Papadimitriou, 2009)

The problem NASH is PPAD-complete.

- One of the main breakthroughs in algorithmic game theory.

NASH and PPAD

- The proof of the correctness of the Lemke–Howson algorithm shows that NASH belongs to PPAD (for nondegenerate games).
- Is NASH PPAD-complete?
  - That is, is it among the most difficult problems in this class?
  - PPAD-completeness gives some evidence of computational intractability, although somehow weaker than NP-completeness.
  - Open for a long time.

Theorem 2.35 (Chen, Deng, and Teng and Daskalakis, Goldberg, and Papadimitriou, 2009)

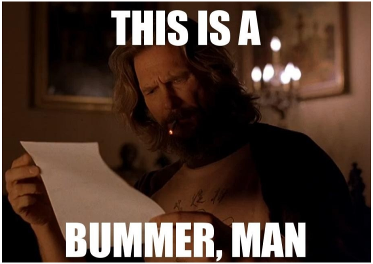
The problem NASH is PPAD-complete.

- One of the main breakthroughs in algorithmic game theory.
- We omit the proof, as it is complicated (the papers have over 50 and 70 pages).

What now?

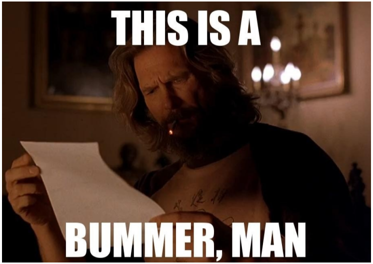
What now?

- So it is likely that there is no polynomial-time algorithm for NASH.



What now?

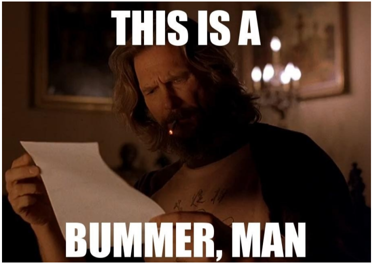
- So it is likely that there is no polynomial-time algorithm for NASH.



- Finding approximate NE in games with at least three players lies in PPAD, but the problem appears to be strictly harder than PPAD.

What now?

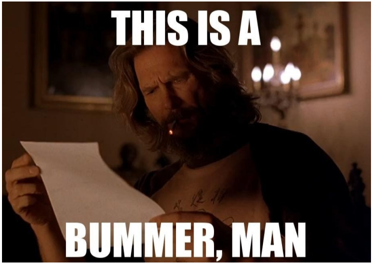
- So it is likely that there is no polynomial-time algorithm for NASH.



- Finding approximate NE in games with at least three players lies in PPAD, but the problem appears to be strictly harder than PPAD.
- If we modify NASH so that the existence is not always guaranteed, then the resulting problem often becomes NP-complete.

What now?

- So it is likely that there is no polynomial-time algorithm for NASH.

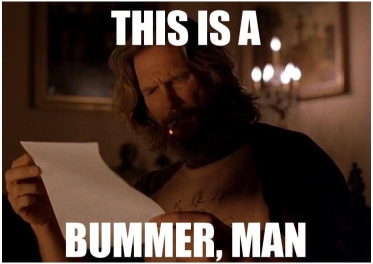


- Finding approximate NE in games with at least three players lies in PPAD, but the problem appears to be strictly harder than PPAD.
- If we modify NASH so that the existence is not always guaranteed, then the resulting problem often becomes NP-complete.

- This seems to be a problem with the concept of NE.

What now?

- So it is likely that there is no polynomial-time algorithm for NASH.

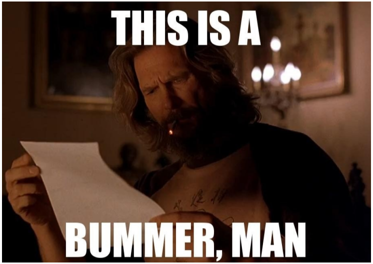


- Finding approximate NE in games with at least three players lies in PPAD, but the problem appears to be strictly harder than PPAD.
- If we modify NASH so that the existence is not always guaranteed, then the resulting problem often becomes NP-complete.

- This seems to be a problem with the concept of NE. “How can we expect the players to find a Nash equilibrium, if our computers cannot?”

What now?

- So it is likely that there is no polynomial-time algorithm for NASH.



- Finding approximate NE in games with at least three players lies in PPAD, but the problem appears to be strictly harder than PPAD.
- If we modify NASH so that the existence is not always guaranteed, then the resulting problem often becomes NP-complete.

- This seems to be a problem with the concept of NE. "How can we expect the players to find a Nash equilibrium, if our computers cannot?"
- We introduce other solution concepts that possess some qualities of NE and yet are easier to compute.

Other notions of equilibria

Two new solution concepts

Two new solution concepts

- Since finding NE is computationally difficult unless $PPAD \subseteq FP$, we look for different solution concepts that are computationally tractable.

Two new solution concepts

- Since finding NE is computationally difficult unless $PPAD \subseteq FP$, we look for different solution concepts that are computationally tractable.
- We introduce two such solution concepts:

Two new solution concepts

- Since finding NE is computationally difficult unless $PPAD \subseteq FP$, we look for different solution concepts that are computationally tractable.
- We introduce two such solution concepts: ε-Nash equilibria

Two new solution concepts

- Since finding NE is computationally difficult unless $PPAD \subseteq FP$, we look for different solution concepts that are computationally tractable.
- We introduce two such solution concepts: ε-Nash equilibria and correlated equilibria.

Two new solution concepts

- Since finding NE is computationally difficult unless $PPAD \subseteq FP$, we look for different solution concepts that are computationally tractable.
- We introduce two such solution concepts: ε-Nash equilibria and correlated equilibria.
  - The first one will seem natural with an easy-to-understand definition,

Two new solution concepts

- Since finding NE is computationally difficult unless $PPAD \subseteq FP$, we look for different solution concepts that are computationally tractable.
- We introduce two such solution concepts: ε-Nash equilibria and correlated equilibria.
    - The first one will seem natural with an easy-to-understand definition, but we will later notice some of its drawbacks.

Two new solution concepts

- Since finding NE is computationally difficult unless $PPAD \subseteq FP$, we look for different solution concepts that are computationally tractable.
- We introduce two such solution concepts: ε-Nash equilibria and correlated equilibria.
  - The first one will seem natural with an easy-to-understand definition, but we will later notice some of its drawbacks.
  - The second one will have a rather complicated definition at first sight,

Two new solution concepts

- Since finding NE is computationally difficult unless $PPAD \subseteq FP$, we look for different solution concepts that are computationally tractable.

- We introduce two such solution concepts: ε-Nash equilibria and correlated equilibria.
  - The first one will seem natural with an easy-to-understand definition, but we will later notice some of its drawbacks.
  - The second one will have a rather complicated definition at first sight, but we will later laern to appreciate it and see that it might be even more natural than NE!

ε-Nash equilibria

ε-Nash equilibria

- For $\varepsilon > 0$, a strategy profile $s = (s_1, \ldots, s_n)$ in a normal-form game $G = (P, A, u)$ is an ε-Nash equilibrium (ε-NE) if, for every player $i \in P$ and every $s_i' \in S_i$, we have $u_i(s_i; s_{-i}) \geq u_i(s_i'; s_{-i}) - \varepsilon$.

ε-Nash equilibria

- For $\varepsilon > 0$, a strategy profile $s = (s_1, \ldots, s_n)$ in a normal-form game $G = (P, A, u)$ is an ε-Nash equilibrium (ε-NE) if, for every player $i \in P$ and every $s_i' \in S_i$, we have $u_i(s_i; s_{-i}) \geq u_i(s_i'; s_{-i}) - \varepsilon$.
  - That is, no other strategy can improve the payoff by more than ε.

ε-Nash equilibria

- For $\varepsilon > 0$, a strategy profile $s = (s_1, \ldots, s_n)$ in a normal-form game $G = (P, A, u)$ is an ε-Nash equilibrium (ε-NE) if, for every player $i \in P$ and every $s_i' \in S_i$, we have $u_i(s_i; s_{-i}) \geq u_i(s_i'; s_{-i}) - \varepsilon$.
  - That is, no other strategy can improve the payoff by more than ε.
  - If we allowed $\varepsilon = 0$, we would get the standard NE.

ε-Nash equilibria

- For $\varepsilon > 0$, a strategy profile $s = (s_1, \ldots, s_n)$ in a normal-form game $G = (P, A, u)$ is an ε-Nash equilibrium (ε-NE) if, for every player $i \in P$ and every $s_i' \in S_i$, we have $u_i(s_i; s_{-i}) \geq u_i(s_i'; s_{-i}) - \varepsilon$.
  - That is, no other strategy can improve the payoff by more than ε.
  - If we allowed $\varepsilon = 0$, we would get the standard NE.
- Advantages:

ε-Nash equilibria

- For $\varepsilon > 0$, a strategy profile $s = (s_1, \ldots, s_n)$ in a normal-form game $G = (P, A, u)$ is an ε-Nash equilibrium (ε-NE) if, for every player $i \in P$ and every $s_i' \in S_i$, we have $u_i(s_i; s_{-i}) \geq u_i(s_i'; s_{-i}) - \varepsilon$.
    - That is, no other strategy can improve the payoff by more than ε.
    - If we allowed $\varepsilon = 0$, we would get the standard NE.
- Advantages:
    - Easy-to-understand definition

ε-Nash equilibria

- For $\varepsilon > 0$, a strategy profile $s = (s_1, \ldots, s_n)$ in a normal-form game $G = (P, A, u)$ is an ε-Nash equilibrium (ε-NE) if, for every player $i \in P$ and every $s_i' \in S_i$, we have $u_i(s_i; s_{-i}) \geq u_i(s_i'; s_{-i}) - \varepsilon$.
  - That is, no other strategy can improve the payoff by more than ε.
  - If we allowed $\varepsilon = 0$, we would get the standard NE.
- Advantages:
  - Easy-to-understand definition
  - ε-NE always exist by Nash's theorem

ε-Nash equilibria

- For $\varepsilon > 0$, a strategy profile $s = (s_1, \ldots, s_n)$ in a normal-form game $G = (P, A, u)$ is an ε-Nash equilibrium (ε-NE) if, for every player $i \in P$ and every $s_i' \in S_i$, we have $u_i(s_i; s_{-i}) \geq u_i(s_i'; s_{-i}) - \varepsilon$.
  - That is, no other strategy can improve the payoff by more than ε.
  - If we allowed $\varepsilon = 0$, we would get the standard NE.
- Advantages:
  - Easy-to-understand definition
  - ε-NE always exist by Nash's theorem (every NE is ε-NE).

ε-Nash equilibria

- For $\varepsilon > 0$, a strategy profile $s = (s_1, \dots, s_n)$ in a normal-form game $G = (P, A, u)$ is an ε-Nash equilibrium (ε-NE) if, for every player $i \in P$ and every $s_i' \in S_i$, we have $u_i(s_i; s_{-i}) \geq u_i(s_i'; s_{-i}) - \varepsilon$.
  - That is, no other strategy can improve the payoff by more than ε.
  - If we allowed $\varepsilon = 0$, we would get the standard NE.
- Advantages:
  - Easy-to-understand definition
  - ε-NE always exist by Nash's theorem (every NE is ε-NE).
  - Using ε as the "machine precision" we do not have to work with irrational numbers.

ε-Nash equilibria

- For $\varepsilon > 0$, a strategy profile $s = (s_1, \ldots, s_n)$ in a normal-form game $G = (P, A, u)$ is an ε-Nash equilibrium (ε-NE) if, for every player $i \in P$ and every $s_i' \in S_i$, we have $u_i(s_i; s_{-i}) \geq u_i(s_i'; s_{-i}) - \varepsilon$.
  - That is, no other strategy can improve the payoff by more than ε.
  - If we allowed $\varepsilon = 0$, we would get the standard NE.
- Advantages:
  - Easy-to-understand definition
  - ε-NE always exist by Nash's theorem (every NE is ε-NE).
  - Using ε as the "machine precision" we do not have to work with irrational numbers.
- Disadvantages:

ε-Nash equilibria

- For $\varepsilon > 0$, a strategy profile $s = (s_1, \ldots, s_n)$ in a normal-form game $G = (P, A, u)$ is an ε-Nash equilibrium (ε-NE) if, for every player $i \in P$ and every $s_i' \in S_i$, we have $u_i(s_i; s_{-i}) \geq u_i(s_i'; s_{-i}) - \varepsilon$.
  - That is, no other strategy can improve the payoff by more than ε.
  - If we allowed $\varepsilon = 0$, we would get the standard NE.
- Advantages:
  - Easy-to-understand definition
  - ε-NE always exist by Nash's theorem (every NE is ε-NE).
  - Using ε as the "machine precision" we do not have to work with irrational numbers.
- Disadvantages:
  - There are ε-NE that are not close to any NE

ε-Nash equilibria

- For $\varepsilon > 0$, a strategy profile $s = (s_1, \ldots, s_n)$ in a normal-form game $G = (P, A, u)$ is an ε-Nash equilibrium (ε-NE) if, for every player $i \in P$ and every $s_i' \in S_i$, we have $u_i(s_i; s_{-i}) \geq u_i(s_i'; s_{-i}) - \varepsilon$.
  - That is, no other strategy can improve the payoff by more than ε.
  - If we allowed $\varepsilon = 0$, we would get the standard NE.
- Advantages:
  - Easy-to-understand definition
  - ε-NE always exist by Nash's theorem (every NE is ε-NE).
  - Using ε as the "machine precision" we do not have to work with irrational numbers.
- Disadvantages:
  - There are ε-NE that are not close to any NE (so ε-NE are not exactly approximations of NE).

ε-Nash equilibria

- For $\varepsilon > 0$, a strategy profile $s = (s_1, \dots, s_n)$ in a normal-form game $G = (P, A, u)$ is an ε-Nash equilibrium (ε-NE) if, for every player $i \in P$ and every $s_i' \in S_i$, we have $u_i(s_i; s_{-i}) \geq u_i(s_i'; s_{-i}) - \varepsilon$.
  - That is, no other strategy can improve the payoff by more than ε.
  - If we allowed $\varepsilon = 0$, we would get the standard NE.
- Advantages:
  - Easy-to-understand definition
  - ε-NE always exist by Nash's theorem (every NE is ε-NE).
  - Using ε as the "machine precision" we do not have to work with irrational numbers.
- Disadvantages:
  - There are ε-NE that are not close to any NE (so ε-NE are not exactly approximations of NE).
  - We will see that his concept is also somehow computationally difficult.

Algorithmic aspects of ε-Nash equilibria

Algorithmic aspects of ε-Nash equilibria

- An optimization problem P with input of size n and a parameter $\varepsilon > 0$ has a PTAS if there is an algorithm that computes an ε-approximate solution of P in time $O(n^{f(1/\varepsilon)})$ for some function f.

Algorithmic aspects of ε-Nash equilibria

- An optimization problem P with input of size n and a parameter $\varepsilon > 0$ has a PTAS if there is an algorithm that computes an ε-approximate solution of P in time $O(n^{f(1/\varepsilon)})$ for some function f.
- The problem P has FPTAS if there is such an algorithm that runs in time $O((1/\varepsilon)^c n^d)$ for some constants c and d.

Algorithmic aspects of ε-Nash equilibria

- An optimization problem P with input of size n and a parameter $\varepsilon > 0$ has a PTAS if there is an algorithm that computes an ε-approximate solution of P in time $O(n^{f(1/\varepsilon)})$ for some function f.
- The problem P has FPTAS if there is such an algorithm that runs in time $O((1/\varepsilon)^c n^d)$ for some constants c and d.
- Do we have FPTAS for ε-NE?

Algorithmic aspects of ε-Nash equilibria

- An optimization problem P with input of size n and a parameter $\varepsilon > 0$ has a PTAS if there is an algorithm that computes an ε-approximate solution of P in time $O(n^{f(1/\varepsilon)})$ for some function f.
- The problem P has FPTAS if there is such an algorithm that runs in time $O((1/\varepsilon)^c n^d)$ for some constants c and d.
- Do we have FPTAS for ε-NE?
    - No, unless $PPAD \subseteq FP$ (Chen, Deng, and Teng, 2006).

Algorithmic aspects of ε-Nash equilibria

- An optimization problem P with input of size n and a parameter $\varepsilon > 0$ has a PTAS if there is an algorithm that computes an ε-approximate solution of P in time $O(n^{f(1/\varepsilon)})$ for some function f.
- The problem P has FPTAS if there is such an algorithm that runs in time $O((1/\varepsilon)^c n^d)$ for some constants c and d.
- Do we have FPTAS for ε-NE?
  - No, unless $PPAD \subseteq FP$ (Chen, Deng, and Teng, 2006).
- Do we have PTAS for ε-NE?

Algorithmic aspects of ε-Nash equilibria

- An optimization problem P with input of size n and a parameter $\varepsilon > 0$ has a PTAS if there is an algorithm that computes an ε-approximate solution of P in time $O(n^{f(1/\varepsilon)})$ for some function f.
- The problem P has FPTAS if there is such an algorithm that runs in time $O((1/\varepsilon)^c n^d)$ for some constants c and d.
- Do we have FPTAS for ε-NE?
  - No, unless $PPAD \subseteq FP$ (Chen, Deng, and Teng, 2006).
- Do we have PTAS for ε-NE?
  - Open problem!

Algorithmic aspects of ε-Nash equilibria

- An optimization problem P with input of size n and a parameter $\varepsilon > 0$ has a PTAS if there is an algorithm that computes an ε-approximate solution of P in time $O(n^{f(1/\varepsilon)})$ for some function f.
- The problem P has FPTAS if there is such an algorithm that runs in time $O((1/\varepsilon)^c n^d)$ for some constants c and d.
- Do we have FPTAS for ε-NE?
  - No, unless $PPAD \subseteq FP$ (Chen, Deng, and Teng, 2006).
- Do we have PTAS for ε-NE?
  - Open problem!
- So what do we have?

Algorithmic aspects of ε-Nash equilibria

- An optimization problem P with input of size n and a parameter $\varepsilon > 0$ has a PTAS if there is an algorithm that computes an ε-approximate solution of P in time $O(n^{f(1/\varepsilon)})$ for some function f.
- The problem P has FPTAS if there is such an algorithm that runs in time $O((1/\varepsilon)^c n^d)$ for some constants c and d.
- Do we have FPTAS for ε-NE?
  - No, unless $PPAD \subseteq FP$ (Chen, Deng, and Teng, 2006).
- Do we have PTAS for ε-NE?
  - Open problem!
- So what do we have? A quasi-polynomial-time algorithm.

Algorithmic aspects of ε-Nash equilibria

- An optimization problem P with input of size n and a parameter $\varepsilon > 0$ has a PTAS if there is an algorithm that computes an ε-approximate solution of P in time $O(n^{f(1/\varepsilon)})$ for some function f.
- The problem P has FPTAS if there is such an algorithm that runs in time $O((1/\varepsilon)^c n^d)$ for some constants c and d.
- Do we have FPTAS for ε-NE?
  - No, unless $PPAD \subseteq FP$ (Chen, Deng, and Teng, 2006).
- Do we have PTAS for ε-NE?
  - Open problem!
- So what do we have? A quasi-polynomial-time algorithm.

Theorem 2.37 (Lipton, Markakis, and Mehta, 2003)

Let $G = (P, A, u)$ be a normal-form game of two players, each having m actions, such that the payoff matrices have entries in $[0, 1]$. For every $\varepsilon > 0$, there is an algorithm for computing ε-NE of G in time $m^{O(\log m/\varepsilon^2)}$.

Algorithmic aspects of ε-Nash equilibria

- An optimization problem P with input of size n and a parameter $\varepsilon > 0$ has a PTAS if there is an algorithm that computes an ε-approximate solution of P in time $O(n^{f(1/\varepsilon)})$ for some function f.
- The problem P has FPTAS if there is such an algorithm that runs in time $O((1/\varepsilon)^c n^d)$ for some constants c and d.
- Do we have FPTAS for ε-NE?
  - No, unless $PPAD \subseteq FP$ (Chen, Deng, and Teng, 2006).
- Do we have PTAS for ε-NE?
  - Open problem!
- So what do we have? A quasi-polynomial-time algorithm.

Theorem 2.37 (Lipton, Markakis, and Mehta, 2003)

Let $G = (P, A, u)$ be a normal-form game of two players, each having m actions, such that the payoff matrices have entries in $[0, 1]$. For every $\varepsilon > 0$, there is an algorithm for computing ε-NE of G in time $m^{O(\log m/\varepsilon^2)}$.

- I no longer present the proof (see the lecture notes).

Correlated equilibria

Correlated equilibria

- The most fundamental solution concept according to several people.

Correlated equilibria

- The most fundamental solution concept according to several people.
- *"If there is intelligent life on other planets, in majority of them, they would have discovered correlated equilibrium before NE."* (Myerson)

Correlated equilibria

- The most fundamental solution concept according to several people.
- *"If there is intelligent life on other planets, in majority of them, they would have discovered correlated equilibrium before NE."* (Myerson)
- In $G = (P, A, u)$, let p be a probability distribution on A,

Correlated equilibria

- The most fundamental solution concept according to several people.
- *"If there is intelligent life on other planets, in majority of them, they would have discovered correlated equilibrium before NE."* (Myerson)
- In $G = (P, A, u)$, let p be a probability distribution on A, that is, $p(a) \geq 0$ for every $a \in A$ and $\sum_{a \in A} p(a) = 1$.

Correlated equilibria

- The most fundamental solution concept according to several people.
- *"If there is intelligent life on other planets, in majority of them, they would have discovered correlated equilibrium before NE."* (Myerson)
- In $G = (P, A, u)$, let p be a probability distribution on A, that is, $p(a) \geq 0$ for every $a \in A$ and $\sum_{a \in A} p(a) = 1$. The distribution p is a correlated equilibrium (CE) in G if

$$\sum_{a_{-i} \in A_{-i}} u_i(a_i; a_{-i}) p(a_i; a_{-i}) \geq \sum_{a_{-i} \in A_{-i}} u_i(a'_i; a_{-i}) p(a_i; a_{-i})$$

for every player $i \in P$ and all pure strategies $a_i, a'_i \in A_i$.

Correlated equilibria

- The most fundamental solution concept according to several people.
- *"If there is intelligent life on other planets, in majority of them, they would have discovered correlated equilibrium before NE."* (Myerson)
- In $G = (P, A, u)$, let p be a probability distribution on A, that is, $p(a) \geq 0$ for every $a \in A$ and $\sum_{a \in A} p(a) = 1$. The distribution p is a correlated equilibrium (CE) in G if

$$\sum_{a_{-i} \in A_{-i}} u_i(a_i; a_{-i}) p(a_i; a_{-i}) \geq \sum_{a_{-i} \in A_{-i}} u_i(a'_i; a_{-i}) p(a_i; a_{-i})$$

  for every player $i \in P$ and all pure strategies $a_i, a'_i \in A_i$.
- Imagine a trusted third party with the distribution p being publicly known.

Correlated equilibria

- The most fundamental solution concept according to several people.
- *"If there is intelligent life on other planets, in majority of them, they would have discovered correlated equilibrium before NE."* (Myerson)
- In $G = (P, A, u)$, let p be a probability distribution on A, that is, $p(a) \geq 0$ for every $a \in A$ and $\sum_{a \in A} p(a) = 1$. The distribution p is a correlated equilibrium (CE) in G if

$$\sum_{a_{-i} \in A_{-i}} u_i(a_i; a_{-i}) p(a_i; a_{-i}) \geq \sum_{a_{-i} \in A_{-i}} u_i(a'_i; a_{-i}) p(a_i; a_{-i})$$

  for every player $i \in P$ and all pure strategies $a_i, a'_i \in A_i$.
- Imagine a trusted third party with the distribution p being publicly known. The trusted third party samples $a \in A$ according to p and privately suggests the strategy a_i to i, but does not reveal a_{-i} to i.

Correlated equilibria

- The most fundamental solution concept according to several people.
- *"If there is intelligent life on other planets, in majority of them, they would have discovered correlated equilibrium before NE."* (Myerson)
- In $G = (P, A, u)$, let p be a probability distribution on A, that is, $p(a) \geq 0$ for every $a \in A$ and $\sum_{a \in A} p(a) = 1$. The distribution p is a correlated equilibrium (CE) in G if

$$\sum_{a_{-i} \in A_{-i}} u_i(a_i; a_{-i}) p(a_i; a_{-i}) \geq \sum_{a_{-i} \in A_{-i}} u_i(a'_i; a_{-i}) p(a_i; a_{-i})$$

  for every player $i \in P$ and all pure strategies $a_i, a'_i \in A_i$.
- Imagine a trusted third party with the distribution p being publicly known. The trusted third party samples $a \in A$ according to p and privately suggests the strategy a_i to i, but does not reveal a_{-i} to i. The player i can follow this suggestion, or not.

Correlated equilibria

- The most fundamental solution concept according to several people.
- *"If there is intelligent life on other planets, in majority of them, they would have discovered correlated equilibrium before NE."* (Myerson)
- In $G = (P, A, u)$, let p be a probability distribution on A, that is, $p(a) \geq 0$ for every $a \in A$ and $\sum_{a \in A} p(a) = 1$. The distribution p is a correlated equilibrium (CE) in G if

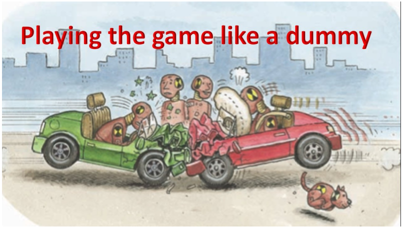
$$\sum_{a_{-i} \in A_{-i}} u_i(a_i; a_{-i}) p(a_i; a_{-i}) \geq \sum_{a_{-i} \in A_{-i}} u_i(a'_i; a_{-i}) p(a_i; a_{-i})$$

  for every player $i \in P$ and all pure strategies $a_i, a'_i \in A_i$.
- Imagine a trusted third party with the distribution p being publicly known. The trusted third party samples $a \in A$ according to p and privately suggests the strategy a_i to i, but does not reveal a_{-i} to i. The player i can follow this suggestion, or not. Then, p is CE if every player maximizes his expected utility by playing the suggested strategy a_i.

Example of correlated equilibria: Game of Chicken

Example of correlated equilibria: Game of Chicken

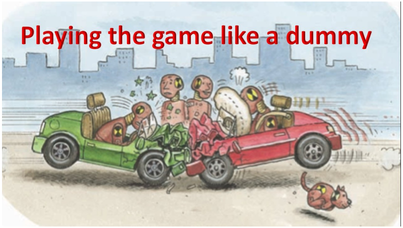
	Stop	Go
Stop	(0,0)	(-1,1)
Go	(1,-1)	(-10,-10)



Sources: https://peakd.com/

Example of correlated equilibria: Game of Chicken

	Stop	Go
Stop	(0,0)	(-1,1)
Go	(1,-1)	(-10,-10)

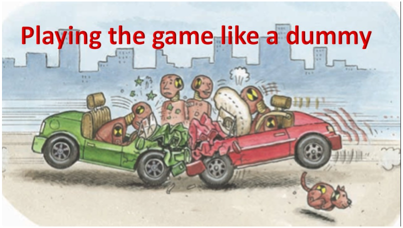


Sources: https://peakd.com/

- There are two pure NE with $(s_1(S), s_2(S)) = (1, 0)$ and $(s_1(S), s_2(S)) = (0, 1)$, and one mixed NE with $(s_1(S), s_2(S)) = (9/10, 9/10)$.

Example of correlated equilibria: Game of Chicken

	Stop	Go
Stop	(0,0)	(-1,1)
Go	(1,-1)	(-10,-10)

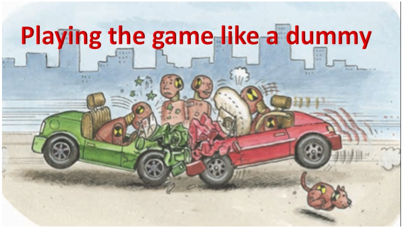


Sources: https://peakd.com/

- There are two pure NE with $(s_1(S), s_2(S)) = (1, 0)$ and $(s_1(S), s_2(S)) = (0, 1)$, and one mixed NE with $(s_1(S), s_2(S)) = (9/10, 9/10)$.
- Consider a trusted third party, a traffic light.

Example of correlated equilibria: Game of Chicken

	Stop	Go
Stop	(0,0)	(-1,1)
Go	(1,-1)	(-10,-10)

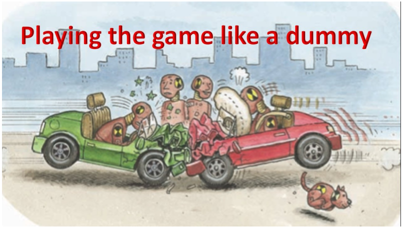


Sources: https://peakd.com/

- There are two pure NE with $(s_1(S), s_2(S)) = (1, 0)$ and $(s_1(S), s_2(S)) = (0, 1)$, and one mixed NE with $(s_1(S), s_2(S)) = (9/10, 9/10)$.
- Consider a trusted third party, a traffic light. The traffic light chooses (S, S), (S, G), and (G, S) independently at random with probability $1/3$.

Example of correlated equilibria: Game of Chicken

	Stop	Go
Stop	(0,0)	(-1,1)
Go	(1,-1)	(-10,-10)

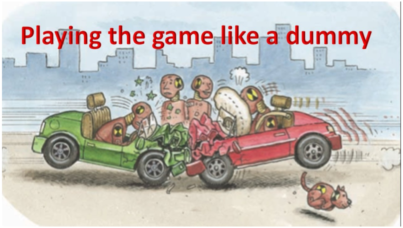


Sources: https://peakd.com/

- There are two pure NE with $(s_1(S), s_2(S)) = (1, 0)$ and $(s_1(S), s_2(S)) = (0, 1)$, and one mixed NE with $(s_1(S), s_2(S)) = (9/10, 9/10)$.
- Consider a trusted third party, a traffic light. The traffic light chooses (S, S), (S, G), and (G, S) independently at random with probability $1/3$. The traffic light gives CE.

Example of correlated equilibria: Game of Chicken

	Stop	Go
Stop	(0,0)	(-1,1)
Go	(1,-1)	(-10,-10)

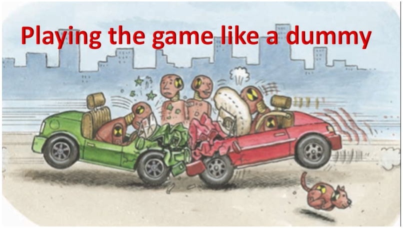


Sources: https://peakd.com/

- There are two pure NE with $(s_1(S), s_2(S)) = (1, 0)$ and $(s_1(S), s_2(S)) = (0, 1)$, and one mixed NE with $(s_1(S), s_2(S)) = (9/10, 9/10)$.
- Consider a trusted third party, a traffic light. The traffic light chooses (S, S), (S, G), and (G, S) independently at random with probability $1/3$. The traffic light gives CE.
  - If 1 follows the suggestion "go", then he gets 1 while deviating gives him 0.

Example of correlated equilibria: Game of Chicken

	Stop	Go
Stop	(0,0)	(-1,1)
Go	(1,-1)	(-10,-10)

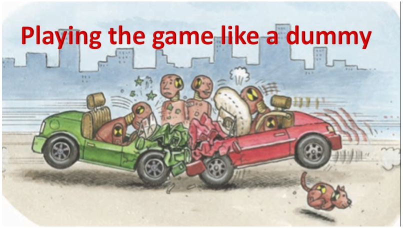


Sources: https://peakd.com/

- There are two pure NE with $(s_1(S), s_2(S)) = (1,0)$ and $(s_1(S), s_2(S)) = (0,1)$, and one mixed NE with $(s_1(S), s_2(S)) = (9/10, 9/10)$.
- Consider a trusted third party, a traffic light. The traffic light chooses (S,S), (S,G), and (G,S) independently at random with probability $1/3$. The traffic light gives CE.
  - If 1 follows the suggestion "go", then he gets 1 while deviating gives him 0.
  - If 1 follows the suggestion "stop", then he gets $-1/2$ while deviating gives him $-9/2$.

Example of correlated equilibria: Game of Chicken

	Stop	Go
Stop	(0,0)	(-1,1)
Go	(1,-1)	(-10,-10)



Sources: https://peakd.com/

- There are two pure NE with $(s_1(S), s_2(S)) = (1, 0)$ and $(s_1(S), s_2(S)) = (0, 1)$, and one mixed NE with $(s_1(S), s_2(S)) = (9/10, 9/10)$.
- Consider a trusted third party, a traffic light. The traffic light chooses (S, S), (S, G), and (G, S) independently at random with probability $1/3$. The traffic light gives CE.
    - If 1 follows the suggestion "go", then he gets 1 while deviating gives him 0.
    - If 1 follows the suggestion "stop", then he gets $-1/2$ while deviating gives him $-9/2$.
    - By symmetry, driver 2 does not deviate as well.

Source: Students of MFF UK

Example of correlated equilibria: Battle of sexes

Example of correlated equilibria: Battle of sexes

	Football	Opera
Football	(2,1)	(0,0)
Opera	(0,0)	(1,2)

Sources: https://media.istockphoto.com/

Example of correlated equilibria: Battle of sexes

	Football	Opera
Football	(2,1)	(0,0)
Opera	(0,0)	(1,2)

Sources: https://media.istockphoto.com/

- There are two pure NE with $(s_1(F), s_2(F)) = (1, 1)$ and $(s_1(F), s_2(F)) = (0, 0)$, and one mixed NE with $(s_1(F), s_2(O)) = (2/3, 2/3)$.

Example of correlated equilibria: Battle of sexes

	Football	Opera
Football	(2,1)	(0,0)
Opera	(0,0)	(1,2)

Sources: https://media.istockphoto.com/

- There are two pure NE with $(s_1(F), s_2(F)) = (1, 1)$ and $(s_1(F), s_2(F)) = (0, 0)$, and one mixed NE with $(s_1(F), s_2(O)) = (2/3, 2/3)$.
- Consider a trusted third party, a mother-in-law.

Example of correlated equilibria: Battle of sexes

	Football	Opera
Football	(2,1)	(0,0)
Opera	(0,0)	(1,2)

Sources: https://media.istockphoto.com/

- There are two pure NE with $(s_1(F), s_2(F)) = (1, 1)$ and $(s_1(F), s_2(F)) = (0, 0)$, and one mixed NE with $(s_1(F), s_2(O)) = (2/3, 2/3)$.
- Consider a trusted third party, a mother-in-law. The mother-in-law flips a coin and chooses (F, F) or (O, O) independently at random with probability $1/2$.

Example of correlated equilibria: Battle of sexes

	Football	Opera
Football	(2,1)	(0,0)
Opera	(0,0)	(1,2)

Sources: https://media.istockphoto.com/

- There are two pure NE with $(s_1(F), s_2(F)) = (1, 1)$ and $(s_1(F), s_2(F)) = (0, 0)$, and one mixed NE with $(s_1(F), s_2(O)) = (2/3, 2/3)$.
- Consider a trusted third party, a mother-in-law. The mother-in-law flips a coin and chooses (F, F) or (O, O) independently at random with probability $1/2$. The mother-in-law gives CE.

Example of correlated equilibria: Battle of sexes

	Football	Opera
Football	(2,1)	(0,0)
Opera	(0,0)	(1,2)

Sources: https://media.istockphoto.com/

- There are two pure NE with $(s_1(F), s_2(F)) = (1,1)$ and $(s_1(F), s_2(F)) = (0,0)$, and one mixed NE with $(s_1(F), s_2(O)) = (2/3, 2/3)$.
- Consider a trusted third party, a mother-in-law. The mother-in-law flips a coin and chooses (F, F) or (O, O) independently at random with probability $1/2$. The mother-in-law gives CE.
  - If the husband follows the suggestion "football", then he gets 2 while deviating gives him 0.

Example of correlated equilibria: Battle of sexes

	Football	Opera
Football	(2,1)	(0,0)
Opera	(0,0)	(1,2)

Sources: https://media.istockphoto.com/

- There are two pure NE with $(s_1(F), s_2(F)) = (1, 1)$ and $(s_1(F), s_2(F)) = (0, 0)$, and one mixed NE with $(s_1(F), s_2(O)) = (2/3, 2/3)$.
- Consider a trusted third party, a mother-in-law. The mother-in-law flips a coin and chooses (F, F) or (O, O) independently at random with probability $1/2$. The mother-in-law gives CE.
  - If the husband follows the suggestion "football", then he gets 2 while deviating gives him 0.
  - If the husband follows the suggestion "opera", then he gets 1 while deviating gives him 0.

Example of correlated equilibria: Battle of sexes

	Football	Opera
Football	(2,1)	(0,0)
Opera	(0,0)	(1,2)

Sources: https://media.istockphoto.com/

- There are two pure NE with $(s_1(F), s_2(F)) = (1, 1)$ and $(s_1(F), s_2(F)) = (0, 0)$, and one mixed NE with $(s_1(F), s_2(O)) = (2/3, 2/3)$.
- Consider a trusted third party, a mother-in-law. The mother-in-law flips a coin and chooses (F, F) or (O, O) independently at random with probability $1/2$. The mother-in-law gives CE.
  - If the husband follows the suggestion "football", then he gets 2 while deviating gives him 0.
  - If the husband follows the suggestion "opera", then he gets 1 while deviating gives him 0.
  - By symmetry, the wife does not deviate as well.

Advantages and disadvantages of correlated equilibria

Advantages and disadvantages of correlated equilibria

- Disadvantages:

Advantages and disadvantages of correlated equilibria

- Disadvantages:
  - The definition of CE takes some getting used to.

Advantages and disadvantages of correlated equilibria

- Disadvantages:
  - The definition of CE takes some getting used to.
- Advantages:

Advantages and disadvantages of correlated equilibria

- Disadvantages:
  - The definition of CE takes some getting used to.
- Advantages:
  - Every NE is CE (Exercise).

Advantages and disadvantages of correlated equilibria

- Disadvantages:
  - The definition of CE takes some getting used to.
- Advantages:
  - Every NE is CE (Exercise). So CE always exist by Nash's theorem.

Advantages and disadvantages of correlated equilibria

- Disadvantages:
  - The definition of CE takes some getting used to.
- Advantages:
  - Every NE is CE (Exercise). So CE always exist by Nash's theorem.
  - Each NE s is CE with the product distribution $p = \prod_{i=1}^{n} s_i$.

Advantages and disadvantages of correlated equilibria

- Disadvantages:
  - The definition of CE takes some getting used to.
- Advantages:
  - Every NE is CE (Exercise). So CE always exist by Nash's theorem.
  - Each NE s is CE with the product distribution $p = \prod_{i=1}^{n} s_i$.
    So CE can give better payoffs than NE.

Advantages and disadvantages of correlated equilibria

- Disadvantages:
  - The definition of CE takes some getting used to.
- Advantages:
  - Every NE is CE (Exercise). So CE always exist by Nash's theorem.
  - Each NE s is CE with the product distribution $p = \prod_{i=1}^{n} s_i$.
    So CE can give better payoffs than NE.
  - Can be computed in polynomial time using LP!

Advantages and disadvantages of correlated equilibria

- Disadvantages:
  - The definition of CE takes some getting used to.
- Advantages:
  - Every NE is CE (Exercise). So CE always exist by Nash's theorem.
  - Each NE s is CE with the product distribution $p = \prod_{i=1}^{n} s_i$. So CE can give better payoffs than NE.
  - Can be computed in polynomial time using LP! Consider the following LP with variables $(p(a))_{a \in A}$:

Advantages and disadvantages of correlated equilibria

- Disadvantages:
  - The definition of CE takes some getting used to.
- Advantages:
  - Every NE is CE (Exercise). So CE always exist by Nash's theorem.
  - Each NE s is CE with the product distribution $p = \prod_{i=1}^{n} s_i$. So CE can give better payoffs than NE.
  - Can be computed in polynomial time using LP! Consider the following LP with variables $(p(a))_{a \in A}$:

$$\max \left\{ \sum_{i \in P} \left(\sum_{a \in A} u_i(a) p(a) \right) \right\} \text{ subject to, for all } i \in P, a_i, a_i' \in A_i,$$

$$\sum_{a_{-i} \in A_{-i}} u_i(a_i; a_{-i}) p(a_i; a_{-i}) \geq \sum_{a_{-i} \in A_{-i}} u_i(a_i'; a_{-i}) p(a_i; a_{-i})$$

$$\sum_{a \in A} p(a) = 1, p(a) \geq 0 \text{ for every } a \in A.$$

- The concept of correlated equilibria was introduced by Robert Aumann, who received a Nobel prize in economics for his work in game theory.

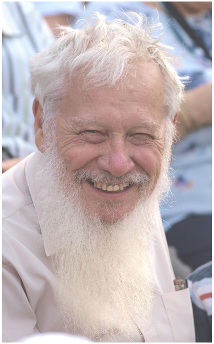


Figure: Robert Aumann (born 1930).

Sources: https://en.wikipedia.org and https://slideslive.com/38910863/strategic-information-theory

- The concept of correlated equilibria was introduced by Robert Aumann, who received a Nobel prize in economics for his work in game theory.

Figure: Robert Aumann (born 1930).

Sources: https://en.wikipedia.org and https://slideslive.com/38910863/strategic-information-theory

- In 2018, Robert Aumann visited Prague and gave a lecture at Prague mathematical colloquium. You can see the lecture here: https://slideslive.com/38910863/strategic-information-theory.

- The concept of correlated equilibria was introduced by Robert Aumann, who received a Nobel prize in economics for his work in game theory.

Figure: Robert Aumann (born 1930).

Sources: https://en.wikipedia.org and https://slideslive.com/38910863/strategic-information-theory

- In 2018, Robert Aumann visited Prague and gave a lecture at Prague mathematical colloquium. You can see the lecture here: `https://slideslive.com/38910863/strategic-information-theory`.

Thank you for your attention.