

Algoritmická teorie her

Jarek Kusák

October 2024

1 Úvod do teorie her

Teorie her se zabývá studiem matematických modelů strategických interakcí mezi racionálními rozhodovateli. Těmito rozhodovateli mohou být jednotlivci, firmy, národy nebo jiné subjekty, které se snaží dosáhnout určitého cíle. Racionální v tomto kontextu znamená, že každý hráč má určité preference a snaží se maximalizovat svůj prospěch.

V tomto předmětu se zaměříme na *algoritmickou stránku* teorie her, kde budeme hledat řešení herních situací pomocí algoritmů.

Figure 1: Obrázek ilustrující strategickou interakci (Zdroj: <https://quantamagazine.org>)

2 Osnova

2.1 Nalezení Nashových ekvilibrií

- Nashovy ekvilibria a Nashův teorém,
- Zero-sum games (nulové součtové hry),
- Bimaticové hry a Lemke–Howsonův algoritmus,
- Další pojetí ekvilibrií,
- Minimalizace lítosti (regret minimization).

2.2 Návrh mechanismů

- Aukce (Vickreyova aukce),
- Myersonova lemma a její aplikace,
- Maximalizace zisků.

3 Hry v normální formě

3.1 Intuitivní vysvětlení

Normální forma hry představuje nejzákladnější a nejpoužívanější způsob, jakým lze hru matematicky popsat. V tomto modelu jsou hráči (hráči jsou jednotlivci nebo subjekty) přiřazeni k určité množině možných akcí, ze kterých si mohou vybrat. Každý hráč si zvolí svou akci současně s ostatními, aniž by věděl, co si zvolí ostatní hráči. Po výběru akcí jsou tyto akce vyhodnoceny pomocí takzvané užitkové funkce, která určuje, jak moc si hráč v dané situaci polepší (nebo pohorší).

3.2 Formální zápis

Hra v normální formě je definována jako trojice (P, A, u) , kde:

- P je konečná množina n hráčů,
- $A = A_1 \times \dots \times A_n$ je množina profilů akcí, kde A_i je množina akcí, které jsou dostupné hráči i ,
- $u = (u_1, \dots, u_n)$ je n -tice, kde každá funkce $u_i : A \rightarrow \mathbb{R}$ je užitková funkce pro hráče i .

Známe-li užitkovou funkci, všichni hráči i si současně zvolí akci a_i z A_i . Výsledný profil akcí $a = (a_1, \dots, a_n)$ je poté vyhodnocen pomocí užitkové funkce.

Souřadnice i -tého hráče $u_i(a)$ z $u(a)$ představuje zisk hráče i pro akci a .

3.3 Příklad

Představme si jednoduchou hru dvou hráčů. Množina hráčů P je $\{1, 2\}$.

Každý hráč má dvě možné akce, takže akční množiny jsou:

$$A_1 = \{X, Y\}, \quad A_2 = \{A, B\}$$

Profily akcí tedy budou všechny možné kombinace:

$$A = A_1 \times A_2 = \{(X, A), (X, B), (Y, A), (Y, B)\}$$

Užitkové funkce pro hráče u_1 a u_2 mohou být definovány jako:

$$u_1(X, A) = 3, \quad u_1(X, B) = 1, \quad u_1(Y, A) = 0, \quad u_1(Y, B) = 2$$

$$u_2(X, A) = 2, \quad u_2(X, B) = 4, \quad u_2(Y, A) = 1, \quad u_2(Y, B) = 3$$

Takže pro každý profil akcí je výsledek popsán dvojicí (u_1, u_2) , která určuje zisky hráčů pro jednotlivé kombinace akcí.

4 Strategie hráčů

4.1 Intuitivní vysvětlení

Ve hře si každý hráč volí určitou strategii. Strategie je soubor pravidel, která určují, jak hráč vybírá své akce z množiny možných akcí A_i . Strategie může být:

- **Čistá strategie:** Hráč si vybere jednu konkrétní akci a tu vždy provede.
- **Smíšená strategie:** Jedná se o pravděpodobnostní rozdělení, kde hráč přiřadí různým akcím pravděpodobnosti a následně si náhodně vybere jednu z akcí na základě těchto pravděpodobností.

4.2 Formální zápis

Každý hráč i následuje určitou strategii s_i , což je předpis, jak si vybírá akce z množiny A_i .

- **Čistá strategie** s_i hráče i je konkrétní akce z A_i :
 - Hráč si vybere jednu konkrétní akci a hraje ji.
 - Profil čisté strategie je n -tice (s_1, \dots, s_n) , kde $s_i \in A_i$ pro každého hráče i .
- **Smíšená strategie** s_i hráče i je pravděpodobnostní rozdělení nad množinou akcí A_i :
 - s_i přiřazuje každé akci $a_i \in A_i$ hodnotu $s_i(a_i) \in [0, 1]$, přičemž platí, že:
$$\sum_{a_i \in A_i} s_i(a_i) = 1$$
 - $s_i(a_i)$ je pravděpodobnost, že hráč i zvolí akci a_i .
 - Množinu všech smíšených strategií hráče i označme S_i .
 - Profil smíšené strategie je n -tice (s_1, \dots, s_n) , kde $s_i \in S_i$ pro každého hráče i .

4.3 Příklad

Představme si dva hráče, kteří mají na výběr ze dvou možných akcí. Množiny akcí jsou:

$$A_1 = \{X, Y\}, \quad A_2 = \{A, B\}$$

Čisté strategie:

- Hráč 1 si vybere buď X , nebo Y .
- Hráč 2 si vybere buď A , nebo B .

Smíšené strategie:

- Hráč 1 může například zvolit strategii s_1 , kde $s_1(X) = 0.7$ a $s_1(Y) = 0.3$. To znamená, že hráč 1 zvolí akci X s pravděpodobností 0.7 a akci Y s pravděpodobností 0.3.
- Hráč 2 může například zvolit strategii s_2 , kde $s_2(A) = 0.5$ a $s_2(B) = 0.5$. To znamená, že hráč 2 volí akce A a B s rovnou pravděpodobností 0.5.

Množina smíšených strategií S_i každého hráče je množina všech možných pravděpodobnostních rozdělení nad jeho akcemi.

- Pro hráče 1 je S_1 tvořena všemi možnými kombinacemi pravděpodobností pro X a Y , například $s_1(X) = 0.7, s_1(Y) = 0.3$, nebo $s_1(X) = 0.5, s_1(Y) = 0.5$, atd.
- Pro hráče 2 je S_2 tvořena všemi možnými kombinacemi pravděpodobností pro A a B , například $s_2(A) = 0.5, s_2(B) = 0.5$, nebo $s_2(A) = 0.9, s_2(B) = 0.1$, atd.

Každá čistá strategie je zároveň smíšená strategie, kde je pravděpodobnost jedné akce rovna 1 a všech ostatních akcí rovna 0.

5 Očekávaný zisk

5.1 Intuitivní vysvětlení

Cílem každého hráče ve hře je maximalizovat svůj očekávaný zisk. Když hráč používá smíšenou strategii, jeho zisk není pevně určen, protože jeho strategie obsahuje pravděpodobnosti různých akcí. Namísto toho se zisk počítá jako očekávaná hodnota — to je vážený průměr zisků, které hráč může získat, přičemž váhy jsou pravděpodobnosti různých akcí.

Jinými slovy, pokud hráč zvolí nějakou akci s určitou pravděpodobností, očekávaný zisk mu říká, kolik v průměru může získat, pokud tuto akci zahraje mnohokrát.

5.2 Formální zápis

Ve hře $G = (P, A, u)$ je očekávaný zisk hráče i pro smíšený profil strategií $s = (s_1, \dots, s_n)$ definován jako:

$$u_i(s) = \sum_{a=(a_1, \dots, a_n) \in A} u_i(a) \cdot \prod_{j=1}^n s_j(a_j)$$

To znamená, že $u_i(s)$ je očekávaná hodnota funkce u_i pod distribučním produktem $\prod_{j=1}^n s_j$.

Dále platí linearita očekávaného zisku:

$$u_i(s) = \sum_{a_i \in A_i} s_i(a_i) \cdot u_i(a_i; s_{-i}),$$

kde $s_{-i} = (s_1, \dots, s_{i-1}, s_{i+1}, \dots, s_n)$ jsou strategie všech ostatních hráčů kromě hráče i .

5.3 Příklad

Představme si dva hráče, kteří mají na výběr ze dvou možných akcí, jak už bylo uvedeno dříve. Akce hráče 1 jsou X a Y , zatímco akce hráče 2 jsou A a B . Množiny akcí jsou tedy:

$$A_1 = \{X, Y\}, \quad A_2 = \{A, B\}$$

Předpokládejme, že smíšené strategie jsou následující:

- Hráč 1 volí akci X s pravděpodobností $s_1(X) = 0.6$ a akci Y s pravděpodobností $s_1(Y) = 0.4$.
- Hráč 2 volí akci A s pravděpodobností $s_2(A) = 0.5$ a akci B s pravděpodobností $s_2(B) = 0.5$.

Nyní si představme, že ziskové funkce jsou definovány takto:

$$u_1(X, A) = 3, \quad u_1(X, B) = 1, \quad u_1(Y, A) = 2, \quad u_1(Y, B) = 4$$

$$u_2(X, A) = 2, \quad u_2(X, B) = 4, \quad u_2(Y, A) = 3, \quad u_2(Y, B) = 1$$

Očekávaný zisk pro hráče 1 lze vypočítat jako:

$$u_1(s) = (0.6 \cdot 0.5) \cdot 3 + (0.6 \cdot 0.5) \cdot 1 + (0.4 \cdot 0.5) \cdot 2 + (0.4 \cdot 0.5) \cdot 4$$

$$u_1(s) = 0.9 + 0.3 + 0.4 + 0.8 = 2.4$$

Takže očekávaný zisk hráče 1 při těchto smíšených strategiích je 2.4.

Podobně můžeme spočítat očekávaný zisk pro hráče 2:

$$u_2(s) = (0.6 \cdot 0.5) \cdot 2 + (0.6 \cdot 0.5) \cdot 4 + (0.4 \cdot 0.5) \cdot 3 + (0.4 \cdot 0.5) \cdot 1$$

$$u_2(s) = 0.6 + 1.2 + 0.6 + 0.2 = 2.6$$

Očekávaný zisk hráče 2 je tedy 2.6.

6 Nashovo ekvilibrium

6.1 Intuitivní vysvětlení

V teorii her studujeme různé koncepty řešení her, které nám pomáhají předpovědět, jak se hra bude hrát. Jedním z nejdůležitějších konceptů je Nashovo ekvilibrium, které představuje stav stability.

Nashovo ekvilibrium nastává tehdy, když každý hráč zvolí takovou strategii, která je pro něj nejlepší odpověď na strategie ostatních hráčů. Jinými slovy, nikdo by neměl důvod měnit svou strategii, pokud ví, co ostatní dělají. Taková situace je stabilní, protože žádný hráč nemá motivaci jednat jinak.

6.2 Formální zápis

Nejlepší odpověď (best response) hráče i na strategický profil s_{-i} je smíšená strategie s_i^* , pro kterou platí:

$$u_i(s_i^*; s_{-i}) \geq u_i(s'_i; s_{-i}) \quad \text{pro každou } s'_i \in S_i.$$

To znamená, že pokud hráč i ví, jaké strategie hrají ostatní hráči, zvolí si takovou strategii s_i^* , která maximalizuje jeho očekávaný zisk.

Pro hru v normální formě $G = (P, A, u)$ s n hráči je Nashovo ekvilibrium (NE) strategický profil (s_1, \dots, s_n) , kde s_i je nejlepší odpověď hráče i na s_{-i} pro každého hráče $i \in P$.

Toto je stabilní koncept řešení: žádný hráč by neměnil svou strategii, pokud by věděl, jaké strategie hrají ostatní hráči. Tento koncept byl představen Johnem Nashem a poprvé se objevil ve výzkumech von Neumanna a Morgensterna.

6.3 Příklad

Představme si jednoduchou hru dvou hráčů. Každý hráč má na výběr dvě možné akce:

$$A_1 = \{X, Y\}, \quad A_2 = \{A, B\}$$

Ziskové funkce pro každého hráče jsou následující:

$$u_1(X, A) = 3, \quad u_1(X, B) = 1, \quad u_1(Y, A) = 0, \quad u_1(Y, B) = 2$$

$$u_2(X, A) = 2, \quad u_2(X, B) = 0, \quad u_2(Y, A) = 3, \quad u_2(Y, B) = 1$$

Teď zjistíme, zda existuje Nashovo ekvilibrium. Pro každého hráče určíme jeho nejlepší odpověď na strategie druhého hráče:

- Pokud hráč 2 zvolí akci A , hráč 1 maximalizuje svůj zisk volbou X , protože $u_1(X, A) = 3$.
- Pokud hráč 2 zvolí akci B , hráč 1 maximalizuje svůj zisk volbou Y , protože $u_1(Y, B) = 2$.

- Pokud hráč 1 zvolí akci X , hráč 2 maximalizuje svůj zisk volbou A , protože $u_2(X, A) = 2$.
- Pokud hráč 1 zvolí akci Y , hráč 2 maximalizuje svůj zisk volbou A , protože $u_2(Y, A) = 3$.

Z tohoto výpočtu plyně, že Nashovo ekvilibrium nastává při strategickém profilu (X, A) , protože žádný hráč nemá motivaci změnit svou strategii, pokud ten druhý zůstane u své volby.

7 Nashovo ekvilibrium v hrách v normální formě

7.1 Intuitivní vysvětlení

Hru ”Kámen, nůžky, papír” lze popsat jako hru v normální formě, kde každý hráč volí jednu ze tří akcí: kámen, nůžky, nebo papír. Výplaty (zisky) hráčů závisí na kombinaci jejich akcí, a jsou znázorněny v tabulce níže. Každý hráč se snaží maximalizovat svůj zisk na základě toho, jakou akci zvolí soupeř.

Ve hře je Nashovo ekvilibrium situace, kdy žádný hráč nemá motivaci změnit svou strategii, pokud ví, co hraje druhý hráč. Tato rovnováha existuje i ve hře ”Kámen, nůžky, papír”, kde smíšené strategie hráčů vedou k tomu, že každý zvolí každou akci s pravděpodobností 1/3.

7.2 Tabulka zisků pro hru ”Kámen, nůžky, papír”

	Rock	Paper	Scissors
Rock	(0, 0)	(-1, 1)	(1, -1)
Paper	(1, -1)	(0, 0)	(-1, 1)
Scissors	(-1, 1)	(1, -1)	(0, 0)

7.3 Formální zápis Nashova ekvilibria

Nejlepší odpověď (best response) hráče i na strategický profil s_{-i} je smíšená strategie s_i^* , pro kterou platí:

$$u_i(s_i^*; s_{-i}) \geq u_i(s'_i; s_{-i}) \quad \text{pro každou } s'_i \in S_i.$$

To znamená, že pokud hráč i ví, jaké strategie hrají ostatní hráči, zvolí si takovou strategii s_i^* , která maximalizuje jeho očekávaný zisk.

Pro hru v normální formě $G = (P, A, u)$ s n hráči je Nashovo ekvilibrium (NE) strategický profil (s_1, \dots, s_n) , kde s_i je nejlepší odpověď hráče i na s_{-i} pro každého hráče $i \in P$.

Každá hra v normální formě má Nashovo ekvilibrium, což bylo dokázáno Nashovým teorémem.

8 Nashův teorém

8.1 Nashův teorém (Věta 2.16)

Každá hra v normální formě má Nashovo ekvilibrium.

8.2 Intuitivní vysvětlení

Nashův teorém je jedním z nejdůležitějších výsledků v teorii her. Tvrdí, že bez ohledu na to, jaká je konkrétní hra v normální formě, existuje vždy alespoň jedno Nashovo ekvilibrium. To znamená, že hráči mohou vždy dosáhnout rovnováhy, ve které nikdo nemá důvod měnit svou strategii, pokud ostatní zůstanou u svých voleb.

Tento teorém zaručuje existenci rovnovážného stavu, ale neříká nám, kolik ekvilibrium existuje nebo jak je nalézt.

9 Přípravy na důkaz Nashova teorému

Důkaz Nashova teorému je v podstatě topologický, protože jeho hlavní složkou je věta o pevném bodě. Použijeme větu, která pochází od matematiky L. E. J. Brouwera.

9.1 Pojmy potřebné k důkazu

- Pro $d \in \mathbb{N}$, podmnožina $X \subset \mathbb{R}^d$ je kompaktní, pokud je uzavřená a omezená.
- Řekneme, že podmnožina $Y \subset \mathbb{R}^d$ je konvexní, pokud každý úsečka spojující dva body z Y je zcela obsažena v Y . Formálně: pro všechna $x, y \in Y$, platí, že $tx + (1 - t)y \in Y$ pro každé $t \in [0, 1]$.
- Pro n affině nezávislých bodů $x_1, \dots, x_n \in \mathbb{R}^d$, je $(n - 1)$ -jednoduchých Δ_n nad x_1, \dots, x_n množinou konvexních kombinací těchto bodů. Každý simplex je kompaktní konvexní množina v \mathbb{R}^d .

9.2 Lemma (Lemma 2.18)

Pro $n, d_1, \dots, d_n \in \mathbb{N}$, nechť K_1, \dots, K_n jsou kompaktní množiny, přičemž každá $K_i \subseteq \mathbb{R}^{d_i}$. Pak $K_1 \times \dots \times K_n$ je kompaktní množina v $\mathbb{R}^{d_1 + \dots + d_n}$.

10 Brouwerova věta o pevném bodě

10.1 Brouwerova věta o pevném bodě (Věta 2.17)

Pro každé $d \in \mathbb{N}$, nechť $K \subset \mathbb{R}^d$ je neprázdná kompaktní konvexní množina a $f : K \rightarrow K$ je spojité zobrazení. Pak existuje pevný bod $x_0 \in K$ pro f , tj. $f(x_0) = x_0$.

11 Důkaz Nashova teorému I

11.1 Intuitivní vysvětlení

V této části důkazu chceme aplikovat Brouwerovu větu o pevném bodě. Brouwerova věta říká, že pro každou kompaktní a konvexní množinu a spojité zobrazení na této množině existuje bod, který se mapuje sám na sebe (pevný bod). V rámci Nashova teorému se snažíme ukázat, že tyto pevné body odpovídají Nashovým ekvilibriím.

Proto nejprve musíme najít vhodnou kompaktní a konvexní množinu K , která reprezentuje všechny možné smíšené strategie hráčů, a spojité zobrazení f , jehož pevné body jsou Nashova ekvilibria.

11.2 Formální zápis

Nechť $G = (P, A, u)$ je hra v normální formě s n hráči. Připomeňme, že S_i je množina smíšených strategií hráče i .

Chceme aplikovat Brouwerovu větu, a proto musíme najít vhodné kompaktní a konvexní těleso K a spojité zobrazení $f : K \rightarrow K$, jehož pevné body jsou Nashova ekvilibria ve hře G .

Začínáme výběrem K . Nechť $K = S_1 \times \dots \times S_n$ je množina všech smíšených strategií.

- Ověříme, že K je kompaktní a konvexní.
- Podle definice je každý S_i simplex, který je kompaktní a konvexní.
- Podle Lemmy 2.18 je množina $K = S_1 \times \dots \times S_n$ kompaktní.
- Pro jakýkoli strategický profil $s = (s_1, \dots, s_n)$ a $s' = (s'_1, \dots, s'_n) \in K$ a číslo $t \in [0, 1]$, bod

$$ts + (1 - t)s' = (ts_1 + (1 - t)s'_1, \dots, ts_n + (1 - t)s'_n)$$

je také smíšený strategický profil v K . Tím pádem je K konvexní.

Takže K je kompaktní a konvexní množina.

12 Důkaz Nashova teorému II

12.1 Intuitivní vysvětlení

V této části důkazu najdeme spojité zobrazení $f : K \rightarrow K$, které nám umožní aplikovat Brouwerovu větu o pevném bodě. Zobrazení f bude definováno tak, že vylepší“ strategii každého hráče tím, že zvýší pravděpodobnost těch akcí, které jsou lepší odpověď na strategie ostatních hráčů.

Cílem je vytvořit zobrazení, které pro každý strategický profil určí nový „vylepšený“ profil, který maximalizuje očekávaný zisk hráče, pokud všichni ostatní hráči nezmění své strategie. Tento nový profil bude pak Nashovo ekvilibrium, pokud je stabilní – tedy když žádný hráč nemá důvod měnit svou strategii.

12.2 Formální zápis

Nyní najdeme spojité zobrazení $f : K \rightarrow K$.

Pro každého hráče $i \in P$ a akci $a_i \in A_i$ definujeme zobrazení $\varphi_{i,a_i} : K \rightarrow \mathbb{R}$ následujícím způsobem:

$$\varphi_{i,a_i}(s) = \max\{0, u_i(a_i; s_{-i}) - u_i(s)\}$$

- $\varphi_{i,a_i}(s) > 0$ právě tehdy, pokud si hráč i může zlepšit svůj zisk použitím akce a_i místo současné strategie s_i .
- Podle definice u_i je toto zobrazení spojité.

Pro dané $s \in K$ definujeme nový vylepšený“ strategický profil $s' \in K$ jako:

$$s'_i(a_i) = \frac{s_i(a_i) + \varphi_{i,a_i}(s)}{\sum_{b_i \in A_i} (s_i(b_i) + \varphi_{i,b_i}(s))} = \frac{s_i(a_i) + \varphi_{i,a_i}(s)}{1 + \sum_{b_i \in A_i} \varphi_{i,b_i}(s)}$$

- Zvyšujeme pravděpodobnost u akcí, které jsou lepší odpovědí na s_{-i} .
- $s' \in K$, protože každé $s'_i(a_i)$ leží v intervalu $[0, 1]$ a $\sum_{a_i \in A_i} s'_i(a_i) = 1$.

Nyní definujeme zobrazení f tak, že pro každý s nastavíme $f(s) = s'$.

13 Důkaz Nashova teorému III

13.1 Intuitivní vysvětlení

V této poslední části důkazu ukážeme, že pevné body zobrazení f jsou právě Nashova ekvilibria ve hře G . Brouwerova věta nám zaručuje existenci pevného bodu, takže naším úkolem je nyní dokázat, že tento pevný bod odpovídá Nashovu ekvilibriu.

Nejprve ověříme, že pokud s je Nashovo ekvilibrium, pak je to také pevný bod zobrazení f . Dále ukážeme, že pokud je s pevný bod zobrazení f , pak je to také Nashovo ekvilibrium.

13.2 Formální zápis

Protože zobrazení φ_{i,a_i} jsou spojité, z toho vyplývá, že zobrazení f je také spojité.

Zbývá ukázat, že pevné body zobrazení f jsou právě Nashova ekvilibria ve hře G . Potom Brouwerova věta nám dává pevný bod f , což je Nashovo ekvilibrium ve hře G .

- Nejprve, pokud s je Nashovo ekvilibrium, pak jsou všechny funkce φ_{i,a_i} nulové, a tedy $f(s) = s$. Tudíž s je pevný bod zobrazení f .
- Dále předpokládejme, že $s = (s_1, \dots, s_n) \in K$ je pevný bod zobrazení f .
 - Pro každého hráče i existuje $a'_i \in A_i$, pro které $s_i(a'_i) > 0$, takže platí:

$$u_i(a'_i; s_{-i}) \leq u_i(s).$$

Jinak by $u_i(s)$ bylo menší než $\sum_{a_i \in A_i} s_i(a_i) u_i(a_i; s_{-i})$, což by bylo v rozporu s linearitou očekávaného zisku.

- Tím pádem $\varphi_{i,a'_i}(s) = 0$, a dostaneme:

$$s'_i(a'_i) = \frac{s_i(a'_i)}{1 + \sum_{b_i \in A_i} \varphi_{i,b_i}(s)}.$$

- Jelikož s je pevný bod, platí $s'_i(a'_i) = s_i(a'_i)$, a protože $s_i(a'_i) > 0$, znamená to, že jmenovatel v tomto výrazu je 1. To znamená, že $\varphi_{i,b_i}(s) = 0$ pro všechna $b_i \in A_i$.
- Z toho vyplývá, že:

$$u_i(s'_i; s_{-i}) = \sum_{b_i \in A_i} s''_i(b_i) u_i(b_i; s_{-i}) \leq \sum_{b_i \in A_i} s'_i(b_i) u_i(b_i; s_{-i}) = u_i(s).$$

Tudíž s je Nashovo ekvilibrium.

14 Pareto optimálnost

14.1 Intuitivní vysvětlení

Pareto optimálnost je další zajímavý koncept řešení her, který je odlišný od Nashova ekvilibria. Zatímco Nashovo ekvilibrium se zaměřuje na stabilitu, Pareto optimálnost se snaží identifikovat takový stav, který je "nejlepší" pro všechny hráče společně. Konkrétně jde o situaci, kdy žádný hráč nemůže zlepšit svůj zisk, aniž by zhorskil zisk někoho jiného.

Pokud strategie s Pareto dominuje strategii s' , znamená to, že všichni hráči jsou na tom minimálně stejně dobře ve strategii s jako v s' , a přinejmenším jeden hráč je na tom v s lépe.

14.2 Formální zápis

Chceme zachytit ”nejlepší” stav hry. To může být obtížné, zvažme například hru **Battle of the sexes**.

Strategický profil s ve hře G Pareto dominuje strategii s' , pokud pro každého hráče i platí, že $u_i(s) \geq u_i(s')$, a navíc existuje hráč j , pro kterého platí, že $u_j(s) > u_j(s')$.

- Vztah \prec je částečné uspořádání množiny všech strategických profilů G .
- Výstupy hry G , které jsou považovány za nejlepší, jsou maximální prvky množiny S ve vztahu \prec .

Strategický profil $s \in S$ je Pareto optimální, pokud neexistuje jiný strategický profil $s' \in S$, který by Pareto dominoval s .

- V **nulových hrách** jsou všechny strategické profily Pareto optimální.
- Ne všechna Nashova ekvilibria jsou Pareto optimální (například Nashovo ekvilibrium ve vězeňské dilematu není Pareto optimální).

15 Hledání Nashova ekvilibria

15.1 Intuitivní vysvětlení

Víme, že Nashovo ekvilibrium (NE) existuje v každé hře v normální formě, což dokazuje Nashův teorém. Problém je však v tom, že zatím nemáme obecný algoritmus, jak Nashovo ekvilibrium v každé hře efektivně najít. To znamená, že v praxi nevíme, jak ekvilibrium najít pro všechny typy her.

Začínáme však se zvláštním typem her – s tzv. nulovými hrami (zero-sum games), které mají jen dva hráče. Ukažeme, že v tomto případě můžeme Nashovo ekvilibrium najít efektivně. Nulové hry jsou speciální v tom, že zisk jednoho hráče je přesně vyvážen ztrátou druhého hráče – co jeden hráč získá, druhý ztratí.

Historicky byly nulové hry prvními typy her, které byly zkoumány v teorii her, a to především díky práci Morgensterna a Von Neumanna ve 40. letech 20. století.

—

15.2 Formální zápis

- Víme, že Nashova ekvilibria (NE) existují v každé hře v normální formě (Nashův teorém).
- Nicméně zatím nemáme žádný algoritmus, jak je efektivně najít.
- Začínáme jednoduchou třídou her se dvěma hráči, tzv. **nulovými hrami**.

- Ukážeme, že v tomto případě můžeme NE najít efektivně. Ve skutečnosti ukážeme, že NE ”řeší” nulové hry úplně.
- Historicky byly nulové hry zkoumány jako první v teorii her (Morgenstern a Von Neumann ve 40. letech).

16 Nulové hry (Zero-sum games)

16.1 Intuitivní vysvětlení

Nulové hry jsou speciální typ dvouhráčových her, kde zisk jednoho hráče je přesně vyvážen ztrátou druhého hráče. Jinými slovy, pokud jeden hráč získá určitou hodnotu, druhý hráč tuto hodnotu ztratí. Takovéto hry lze snadno reprezentovat pomocí výplatní matic a smíšených strategií hráčů.

Nulové hry byly historicky prvními typy her, které byly v teorii her zkoumány, a mají velmi specifickou vlastnost – Nashovo ekvilibrium v těchto hrách lze efektivně nalézt. Hra je považována za vyřešenou, když známe její Nashovo ekvilibrium.

16.2 Formální zápis

Definice nulové hry:

- Dvouhráčové hry (P, A, u) , kde $u_1(a) = -u_2(a)$ pro každou $a \in A$.
-

17 Reprezentace nulových her

- Pro nulové hry se naše notace zjednoduší.
- Nechť $G = (P, A = A_1 \times A_2, u)$ je nulová hra. To znamená, že $u_1(a) + u_2(a) = 0$ pro každou $a \in A$.
- Pokud $A_1 = \{1, \dots, m\}$ a $A_2 = \{1, \dots, n\}$, potom můžeme hru G reprezentovat pomocí výplatní matici M , kde $M_{i,j} = u_1(i, j) = -u_2(i, j)$.

Strategické profily a očekávané zisky:

- Pro strategický profil (s_1, s_2) píšeme $x_i = s_1(i)$ a $y_j = s_2(j)$, kde s_1 a s_2 jsou smíšené strategie.
- Smíšené strategie jsou reprezentovány jako vektory $x = (x_1, \dots, x_m)$ a $y = (y_1, \dots, y_n)$, které splňují $\sum_{i=1}^m x_i = 1$ a $\sum_{j=1}^n y_j = 1$.

- Očekávaný zisk hráče 1 je dán vztahem:

$$u_1(s) = \sum_{a=(i,j) \in A} u_1(a) s_1(i) s_2(j) = \sum_{i=1}^m \sum_{j=1}^n M_{i,j} x_i y_j = x^\top M y = -u_2(s).$$

18 Strategie optimální pro nejhorší případ

- Nejlepší odpověď hráče 2 na strategii x hráče 1 je vektor $y \in S_2$, který minimalizuje $x^\top M y$. Nejlepší odpověď hráče 1 na strategii y hráče 2 je $x \in S_1$, který maximalizuje $x^\top M y$.
- Definujme $\beta(x) = \min_{y \in S_2} x^\top M y$ jako nejlepší očekávaný zisk hráče 2 proti x a $\alpha(y) = \max_{x \in S_1} x^\top M y$ jako nejlepší očekávaný zisk hráče 1 proti y .
- Strategický profil (x, y) je Nashovo ekvilibrium, pokud a jen pokud splňuje $\beta(x) = x^\top M y = \alpha(y)$.

Nejhorší možná optimální strategie:

- Hráč 1 očekává, že hráč 2 vybere nejlepší odpověď na každou strategii x , kterou může použít. Hráč 1 tedy vybere smíšenou strategii \bar{x} z S_1 , která maximalizuje jeho očekávaný zisk pod tímto, spíše pesimistickým předpokladem.
- Tato strategie optimální pro nejhorší případ pro hráče 1 splňuje $\beta(\bar{x}) = \max_{x \in S_1} \beta(x)$.
- Strategie optimální pro nejhorší případ pro hráče 2 je smíšená strategie $\bar{y} \in S_2$, která splňuje $\alpha(\bar{y}) = \min_{y \in S_2} \alpha(y)$.

19 Strategie optimální pro nejhorší případ a Nashovo ekvilibrium

19.1 Intuitivní vysvětlení

Abychom dosáhli Nashova ekvilibria v nulových hrách, oba hráči musí zvolit své optimální strategie pro nejhorší možný případ. To znamená, že každý hráč si zvolí strategii, která maximalizuje jeho očekávaný zisk, i kdyby soupeř hrál proti němu co nejlépe (tedy aby minimalizoval jeho zisk).

19.2 Formální zápis

- Abychom dosáhli Nashova ekvilibria v nulové hře, oba hráči musí zvolit své optimální strategie pro nejhorší možný případ.

19.3 Lemma 2.20

- (a) Pro všechna $x \in S_1$ a $y \in S_2$ platí $\beta(x) \leq x^\top M y \leq \alpha(y)$.
 - (b) Pokud je strategický profil (x^*, y^*) Nashovým ekvilibriem, pak obě strategie x^* a y^* jsou optimální pro nejhorší případ.
 - (c) Jakékoliv strategie $x^* \in S_1$ a $y^* \in S_2$, které splňují $\beta(x^*) = \alpha(y^*)$, tvoří Nashovo ekvilibrium (x^*, y^*) .
- (a) Toto plyne přímo z definic β a α .
 - (b) Část (a) naznačuje, že $\beta(x) \leq \alpha(y^*)$ pro každé $x \in S_1$. Protože (x^*, y^*) je Nashovo ekvilibrium, platí $\beta(x^*) = \alpha(y^*)$, a tudíž $\beta(x) \leq \beta(x^*)$ pro každé $x \in S_1$. To znamená, že x^* je optimální pro nejhorší případ pro hráče 1. Analogicky pro hráče 2.
 - (c) Pokud $\beta(x^*) = \alpha(y^*)$, pak (a) implikuje, že $\beta(x^*) = (x^*)^\top M y^* = \alpha(y^*)$.
-

20 Minimax teorém

20.1 Teorém 2.21

- Pro každou nulovou hru existují optimální strategie pro nejhorší případ pro oba hráče, které lze efektivně spočítat. Existuje číslo v , takové, že pro jakékoliv optimální strategie pro nejhorší případ x^* a y^* , strategický profil (x^*, y^*) tvoří Nashovo ekvilibrium a platí $\beta(x^*) = (x^*)^\top M y^* = \alpha(y^*) = v$.

21 Minimax teorém: poznámky

21.1 Intuitivní vysvětlení

Minimax teorém byl klíčovým bodem pro rozvoj teorie her. Von Neumann v roce 1928 dokázal, že pro každou nulovou hru existuje Nashovo ekvilibrium, které lze efektivně nalézt pomocí optimalizace pro nejhorší možný případ. Tento teorém ukázal, že ve nulových hrách není žádné tajemství – hráči si mohou zvolit své optimální strategie a minimalizovat ztráty, i kdyby jejich soupeř hrál co nejlépe proti nim.

21.2 Formální poznámky

- Minimax teorém byl klíčovým začátkem teorie her.
- Dokázal jej Von Neumann v roce 1928 (před Nashovým teorémem).

- Von Neumann řekl: Jak to vidím, nemohla by existovat teorie her bez tohoto teorému... Myslel jsem si, že nestojí za to publikovat cokoliv, dokud nebyl Minimax teorém dokázán.“
- Minimax teorém nám říká vše o nulových hrách: existuje Nashovo ekvilibrium (NE) a lze jej efektivně najít. Navíc existuje jedinečná hodnota hry $v = (x^*)^\top My^*$, což je výplata získaná v jakémkoli NE (x^*, y^*) .
- V nulových hrách neexistují žádná tajemství: i když známe strategie obou hráčů, nemění to nic. Každý hráč může zvolit svou optimální strategii pro nejhorší případ a získat výplatu $\geq v$. Pokud soupeř zvolí svou optimální strategii pro nejhorší případ, jeho výplata bude vždy $\leq v$.
- Název teorému vychází z rovnosti:

$$\beta(x^*) = v = \alpha(y^*) \quad \text{se stane} \quad \max_{x \in S_1} \min_{y \in S_2} x^\top My = v = \min_{y \in S_2} \max_{x \in S_1} x^\top My.$$

- Původní důkaz používá Brouwerovu větu o pevném bodě. My budeme používat lineární programování.

22 Geometrické pojmy

22.1 Intuitivní vysvětlení

Pro důkaz Minimaxové věty budeme potřebovat několik základních geometrických pojmu. Hyperplochy a poloprostory jsou základní stavební kameny lineárního programování, protože nám umožňují definovat lineární omezení a optimalizaci.

- **Hyperplocha**^{**} je obecně plochý geometrický objekt, který dělí prostor na dvě části. V \mathbb{R}^2 se jedná o přímku, v \mathbb{R}^3 je to rovina, a ve vyšších dimenzích se jedná o hyperplochu. - **Poloprostor**^{**} je oblast, která vznikne tím, že vezmeme všechny body na jedné straně hyperplochy (včetně bodů na ní samotné).

22.2 Formální zápis

- **Hyperplocha**^{**} v \mathbb{R}^d je množina bodů

$$\{x \in \mathbb{R}^d : v^\top x = w\}$$

pro nějaké $v \in \mathbb{R}^d$ a $w \in \mathbb{R}$.

- **Poloprostor**^{**} v \mathbb{R}^d je množina bodů

$$\{x \in \mathbb{R}^d : v^\top x \leq w\}.$$

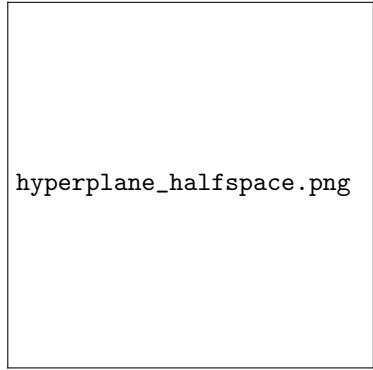


Figure 2: Geometrické znázornění hyperplochy a poloprostoru.

23 Pojmy z geometrie

23.1 Intuitivní vysvětlení

Ve více dimenzích hrají důležitou roli polyedry, což jsou objekty, které vznikají jako průnik konečného počtu poloprostorů. Polyedry nám umožňují modelovat omezení pro optimalizační úlohy, jako je lineární programování. Speciální typ polyedru, nazývaný polytop, je omezený polyedr.

- **Polyedr** je průnik poloprostorů. Pokud je omezený, nazýváme ho polytop. - **Jednoduchý polytop** je takový, kde je každý vrchol připojen přesně k d hranám (v d -dimenzionálním prostoru).

23.2 Formální zápis

- Konvexní **polyedr** P v \mathbb{R}^d je průnik konečného počtu poloprostorů v \mathbb{R}^d :

$$P = \{x \in \mathbb{R}^d : Vx \leq u\}$$

pro nějakou matici $V \in \mathbb{R}^{n \times d}$ a $u \in \mathbb{R}^n$, kde n je počet poloprostorů určujících P .

- **Polytop** je omezený polyedr. Polytop je jednoduchý, pokud je každý jeho vrchol připojen přesně k d hranám.
-

24 Lineární programování

24.1 Intuitivní vysvětlení

Lineární programování (LP) je metoda, která slouží k nalezení optimálního řešení pro úlohu s lineární objektivní funkcí a lineárními omezeními. Tato

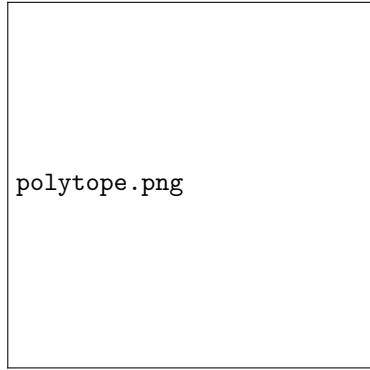


Figure 3: Polyedr a omezený polytop P .

metoda je efektivní, protože umožňuje řešit optimalizační problémy v polynomálním čase.

- **Lineární program** je optimalizační úloha, kde je cílem maximalizovat nebo minimalizovat lineární funkci, přičemž všechny podmínky jsou také lineární.
- LP se typicky zapisuje v kanonické formě, kde hledáme maximum lineární funkce za podmínek lineárních nerovnic.
- Existuje několik metod řešení LP, z nichž nejznámější je **Simplexová metoda** a **Elipsoidní metoda**, která je schopna řešit LP v polynomálním čase.

24.2 Formální zápis

Každý lineární program P lze vyjádřit v kanonické formě:

$$\max c^\top x \quad \text{takové, že} \quad Ax \leq b \quad \text{a} \quad x \geq 0$$

kde $c \in \mathbb{R}^m$, $b \in \mathbb{R}^n$, a $A \in \mathbb{R}^{n \times m}$ jsou dané parametry.

Lineární programování lze také vizualizovat graficky, jak je vidět na následujícím obrázku:

25 Dvojnost v lineárním programování

Lineární program P , ve kterém chceme

$$\text{maximalizovat } c^T x \text{ za podmínek } Ax \leq b \text{ a } x \geq 0$$

se nazývá primální lineární program.

Související dualitní lineární program D je formulován takto:

$$\text{minimalizovat } b^T y \text{ za podmínek } A^T y \geq c \text{ a } y \geq 0.$$

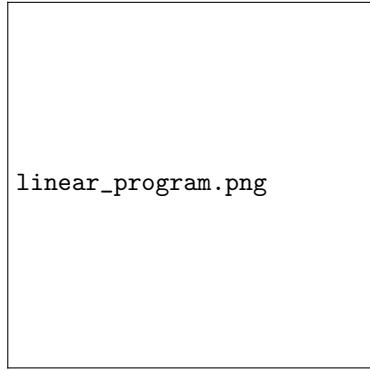


Figure 4: Grafické řešení lineárního programu.

- Primální program se zaměřuje na maximalizaci cílové funkce s lineárními omezeními.
- Dualitní program vyjadřuje stejný problém z jiné perspektivy, tentokrát z pohledu omezení.

25.1 Dvojnostní věta (Theorem 2.22)

Pokud oba programy P a D mají řešení, pak mají obě řešení optimální hodnotu. Navíc, pokud x^* a y^* jsou optimálními řešeními programů P a D , platí, že

$$c^T x^* = b^T y^*.$$

To znamená, že maximum primálního programu se rovná minimu dualitního programu.

	Primální lineární program	Dualitní lineární program
Proměnné	x_1, \dots, x_m	y_1, \dots, y_n
Matice	$A \in \mathbb{R}^{n \times m}$	$A^T \in \mathbb{R}^{m \times n}$
Pravá strana	$b \in \mathbb{R}^n$	$c \in \mathbb{R}^m$
Cílová funkce	$\max c^T x$	$\min b^T y$
Omezení	$Ax \leq b$	$A^T y \geq c$

Table 1: Recept na vytváření dualitních programů.

26 Důkaz Minimaxové věty I

26.1 Intuitivní vysvětlení

Cílem je nalézt Nashovo ekvilibrium v zero-sum hře, kde oba hráči volí své optimální strategie, které maximalizují jejich nejhorší možnou výplatu. Formálně

však zjistíme, že tento přístup není lineárním programováním, protože cílová funkce $\beta(x)$ není lineární v proměnných. Tento krok tedy představuje, jak se dá problém špatně formulovat a co nás to učí o potřebě jiného přístupu.

26.2 Formální zápis

Minimaxová věta říká, že pro každou zero-sum hru existují optimální strategie pro oba hráče, které lze nalézt pomocí lineárního programování.

Naším cílem je maximalizovat nejhorší očekávanou výplatu hráče 1. Tedy najít $\beta(x)$, což je minimální možná výplata, kterou může hráč 1 obdržet.

Chceme maximalizovat:

$$\beta(x) = \max_{x \in S_1} \min_{y \in S_2} x^\top M y$$

Pokusíme se problém naivně formulovat jako lineární program (LP):

$$\max_{x_1, \dots, x_m} \beta(x) \quad \text{za podmínek: } \sum_{i=1}^m x_i = 1, \quad x_i \geq 0 \forall i$$

Avšak tento problém nelze formulovat jako LP, protože funkce $\beta(x) = \min_{y \in S_2} x^\top M y$ není lineární v x . Potřebujeme jiný přístup.

27 Důkaz Minimaxové věty II

27.1 Intuitivní vysvětlení

V tomto kroku přecházíme k lepší formulaci problému, kde zafixujeme x a najdeme nejlepší možnou odpověď hráče 2. To se dá formulovat jako lineární program P , kde hráč 2 minimalizuje očekávanou výplatu hráče 1. Následně formulujeme duální verzi tohoto problému, která je lineárním programem s jedinou proměnnou x_0 . Díky duální větě víme, že oba programy P a D mají stejně optimální řešení.

27.2 Formální zápis

Pro fixní $x \in S_1$ můžeme nalézt nejlepší odpověď hráče 2, tedy strategii y , která minimalizuje výplatu hráče 1. To formulujeme jako následující lineární program P :

$$(P) \min_{y_1, \dots, y_n} x^\top M y \quad \text{za podmínek: } \sum_{j=1}^n y_j = 1, \quad y_j \geq 0 \forall j$$

Jeho duální verze je následující:

$$(D) \max_{x_0} x_0 \quad \text{za podmínek: } 1x_0 \leq M^\top x$$

Podle Duální věty mají P a D stejnou optimální hodnotu $\beta(x)$. Proto pokud řešíme program D s proměnnými x_1, \dots, x_m , dostaneme následující lineární program D' :

$$(D') \max_{x_0} x_0 \quad \text{za podmínek: } 1x_0 - M^\top x \leq 0, \quad \sum_{i=1}^m x_i = 1, \quad x_i \geq 0$$

Optimální řešení x^* tohoto programu je optimální strategie hráče 1.

28 Důkaz Minimaxové věty III

28.1 Intuitivní vysvětlení

Analogicky nyní pro hráče 2 formulujeme lineární program P' , kde minimalizujeme výplatu hráče 1. Tento krok dokazuje, že obě strany (hráč 1 i hráč 2) mohou optimalizovat svou strategii prostřednictvím LP. Duální verze obou programů ukazují, že obě strany mají stejnou optimální hodnotu. Tímto dokončujeme důkaz Minimaxové věty.

28.2 Formální zápis

Analogicky můžeme nalézt optimální odpověď hráče 1 na strategii hráče 2, formulovanou jako lineární program P' :

$$(P') \min_{y_0} y_0 \quad \text{za podmínek: } 1y_0 - My \geq 0, \quad \sum_{j=1}^n y_j = 1, \quad y_j \geq 0$$

Dokázali jsme první část Minimaxové věty. Zbývá ukázat, že (x^*, y^*) je Nashovo ekvilibrium a platí:

$$\beta(x^*) = x^{*\top} My^* = \alpha(y^*) = v$$

Podle Duální věty mají P' a D' stejnou optimální hodnotu. Hodnota v je dosažena v každé optimální strategii.

Z části (c) Lemmy 2.20 víme, že (x^*, y^*) je Nashovo ekvilibrium, tedy platí:

$$\beta(x^*) = x^{*\top} My^* = \alpha(y^*)$$

Další intuitivní poznámky

- **Role proměnné x_0 :** Proměnná x_0 reprezentuje dolní hranici, kterou si hráč 1 může zaručit. Optimalizace x_0 zajišťuje, že hráč 1 získá co nejvíce nezávisle na strategii hráče 2.

- **Duální věta:** Duální věta zaručuje, že optimální hodnoty lineárních programů P' a D' jsou rovné, což odpovídá Nashovu ekvilibriu ve hře s nulovým součtem.
- **Fixní a variabilní strategie:** Zatímco programy P a D uvažují fixní strategii pro jednoho hráče, P' a D' hledají optimální strategie pro oba hráče současně.
- **Význam minimaxové věty:** Tato věta dokazuje, že každý hráč může zaručit své minimální zisky, pokud hraje optimálně, a že výsledek hry je určen Nashovým ekvilibriem.

29 Nashovy rovnováhy v bimatrix hrách pomocí hrubé síly

V tomto kroku se zaměřujeme na hledání Nashových rovnováh ve hrách typu *bimatrix games*, což jsou dvouhráčové hry, které nemusí být nutně zero-sum. Prvním krokem je navrhnout algoritmus, který hledá Nashovu rovnováhu pomocí hrubé síly (*brute-force*). Tento přístup znamená, že procházíme všechny možné kombinace strategií obou hráčů a ověřujeme, zda jsou splněny podmínky Nashovy rovnováhy.

29.1 Podmínka nejlepší odpovědi (Best Response Condition)

Klíčovým pozorováním je koncept podpory smíšené strategie. Podpora smíšené strategie hráče i je množina akcí $Supp(s_i) = \{a_i \in A_i : s_i(a_i) > 0\}$, které mají nenulovou pravděpodobnost ve smíšené strategii s_i . Platí následující podmínka:

[Podmínka nejlepší odpovědi] V normální formě hry $G = (P, A, u)$, pro každého hráče $i \in P$, je smíšená strategie s_i nejlepší odpovědí na s_{-i} tehdy a jen tehdy, pokud všechny čisté strategie v podpoře s_i jsou nejlepší odpovědi na s_{-i} .

Smíšená strategie v podstatě „míchá“ čisté strategie jen tehdy, když jsou všechny čisté strategie v podpoře ekvivalentně dobré.

Tato podmínka nám pomáhá zúžit prostor možných strategií. Namísto prohledávání celého prostoru strategií se můžeme zaměřit pouze na ty kombinace strategií, které splňují tuto podmínu.

29.2 Kombinatorický problém

Z pozorování vyplývá, že problém nalezení Nashovy rovnováhy je kombinatorický, nikoliv spojitý. Hlavní výzvou je nalezení správných podpor pro smíšené strategie obou hráčů. Jakmile jsou tyto podpory správně identifikovány, můžeme přesné smíšené strategie spočítat řešením systému algebraických rovnic. V případě dvou hráčů jde o lineární rovnice.

29.3 Důkaz podmínky nejlepší odpovědi

Intuice:

Důkaz této podmínky nejlepší odpovědi je rozdělen na dvě části:

1. **První část** říká, že pokud je strategie hráče s_i nejlepší odpovědí na strategie ostatních hráčů s_{-i} , pak každá akce v podpoře této strategie musí být nejlepší odpovědí na tyto strategie ostatních hráčů. To znamená, že žádná akce, která má nenulovou pravděpodobnost ve smíšené strategii, nedává hráči lepší výsledek než jiné akce v jeho podpoře. Tento krok ukazuje, že hráč už nemá motivaci měnit svou strategii, protože všechny akce, které hraje, jsou optimální.

2. **Druhá část** zavádí myšlenku kontradikce: Předpokládáme, že některá akce v podpoře $Supp(s_i)$ není nejlepší odpověď. Tím bychom mohli zkonstruovat novou smíšenou strategii, která tuto akci buď vyřadí, nebo ji sníží na nižší pravděpodobnost. Tato nová strategie by vedla k lepšímu výsledku pro hráče, což by vedlo k rozporu s tím, že původní strategie byla nejlepší odpověď. Tímto postupem ukazujeme, že každá akce v podpoře musí být skutečně nejlepší odpověď.

Formální důkaz:

První krok: Předpokládejme, že pro každou akci $a_i \in Supp(s_i)$ platí, že

$$u_i(a_i; s_{-i}) \geq u_i(s'_i; s_{-i}) \quad \text{pro každou } s'_i \in S_i.$$

Potom, pro každou $s'_i \in S_i$, linearita funkce u_i implikuje

$$u_i(s) = \sum_{a_i \in Supp(s_i)} s_i(a_i) u_i(a_i; s_{-i}) \geq \sum_{a_i \in Supp(s_i)} s_i(a_i) u_i(s'_i; s_{-i}) = u_i(s'_i; s_{-i}).$$

Tím je potvrzeno, že s_i je nejlepší odpověď.

Druhý krok: Nyní předpokládejme, že s_i je nejlepší odpověď hráče i na strategii s_{-i} . Předpokládejme, že existuje akce $\bar{a}_i \in Supp(s_i)$, která není nejlepší odpověď hráče i na s_{-i} . Potom existuje $s'_i \in S_i$, kde platí:

$$u_i(\bar{a}_i; s_{-i}) < u_i(s'_i; s_{-i}).$$

Jelikož s_i je nejlepší odpověď na s_{-i} , musí platit $s_i(\bar{a}_i) < 1$. Díky linearitě funkce u_i existuje $\hat{a}_i \in Supp(s_i)$ taková, že

$$u_i(\hat{a}_i; s_{-i}) > u_i(\bar{a}_i; s_{-i}).$$

Definujeme novou smíšenou strategii $s_i^* \in S_i$ následovně: nastavíme $s_i^*(\bar{a}_i) = 0$, $s_i^*(\hat{a}_i) = s_i(\hat{a}_i) + s_i(\bar{a}_i)$ a zachováme $s_i^*(a_i) = s_i(a_i)$ jinak. Potom, díky linearitě u_i ,

$$u_i(s_i^*; s_{-i}) = \sum_{a_i \in A_i} s_i^*(a_i) u_i(a_i; s_{-i}) > \sum_{a_i \in A_i} s_i(a_i) u_i(a_i; s_{-i}) = u_i(s),$$

což je spor s předpokladem, že s_i je nejlepší odpověď. Tím je důkaz hotov.

29.4 Best response condition in bimatrix games

Intuice:

Podmínka nejlepší odpovědi v bimaticových hrách se používá k tomu, aby chom navrhli hrubou silou algoritmus pro nalezení Nashovy rovnováhy (NE) v těchto hrách. Pozorování zde spočívá v tom, že pokud chceme najít NE, potřebujeme nalézt nejlepší odpovědi dvou hráčů na smíšené strategie. Každý hráč má sadu strategií, které jsou reprezentovány maticemi výplat, a tyto smíšené strategie mohou být popsány vektorově. Pro hráče 1 zjištujeme, zda je jeho strategie nejlepší odpověď na strategii hráče 2 tím, že sledujeme podmínu nejlepší odpovědi, což vede k výběru optimálních strategií pro oba hráče.

Formální popis:

Uvažujme hru $G = (\{1, 2\}, A = A_1 \times A_2, u)$, což je bimaticová hra.

- Nechť $A_1 = \{1, \dots, m\}$ a $A_2 = \{1, \dots, n\}$, kde m a n jsou počty akcí pro hráče 1 a hráče 2.
- Výplaty u_1 a u_2 lze reprezentovat maticemi $M, N \in \mathbb{R}^{m \times n}$, kde platí:

$$M_{i,j} = u_1(i, j) \quad \text{a} \quad N_{i,j} = u_2(i, j) \quad \text{pro každé } (i, j) \in A_1 \times A_2.$$

- Očekávané výplaty pro smíšené strategie s s vektory x a y (představující pravděpodobnosti výběru akcí) jsou:

$$u_1(s) = x^\top M y \quad \text{a} \quad u_2(s) = x^\top N y.$$

- Podle podmínky nejlepší odpovědi je x nejlepší odpověď na y , pokud a pouze pokud:

$$\forall i \in A_1 : x_i > 0 \implies M_{i,*} y = \max\{M_{k,*} y : k \in A_1\}.$$

- Analogicky, y je nejlepší odpověď na x , pokud a pouze pokud:

$$\forall j \in A_2 : y_j > 0 \implies N_{*,j}^\top x = \max\{N_{k,*}^\top x : k \in A_2\}.$$

Tento formální popis ukazuje, jak hrubou silou hledáme Nashovu rovnováhu tím, že postupně kontrolujeme, zda jsou strategie nejlepšími odpověďmi hráčů navzájem. Reprezentace výplat pomocí matic umožňuje tento proces matematicky formalizovat.

29.5 Nashovy ekvilibria pomocí enumerace podpor I

Intuice: Tento snímek se zaměřuje na specifickou třídu bimaticových her, nazývaných nondegenerované hry. Pointa je, že ve většině bimaticových her existuje pouze omezený počet čistých strategií, které jsou nejlepšími odpověďmi na určité smíšené strategie, což usnadňuje hledání Nashových ekvilibrií. Pro řešení těchto her se používá metoda enumerace podpor, která umožňuje pracovat pouze s omezenou množinou strategií, což snižuje výpočetní složitost. Tento

postup navrhne rovnice, které zajistí, že očekávané výnosy jsou na podporách rovny a maximální.

Budeme se zabývat speciálními bimaticovými hrami (důvod bude jasný později).

- Bimaticová hra je *nondegenerovaná*, pokud existuje nanejvýš k čistých nejlepších odpovědí na každou smíšenou strategii s podporou velikosti k .
- “Většina bimaticových her je nondegenerovaná” a existují metody pro práci s degenerovanými hrami.
- Označme $I \subseteq A_1$ a $J \subseteq A_2$ jako podporu v nondegenerované hře G .
- Definujeme $|I| + |J|$ proměnných x_i pro $i \in I$ a y_j pro $j \in J$, které budou reprezentovat nenulové hodnoty ve vektorech smíšených strategií x a y .
- Definujeme rovnice $\sum_{i \in I} x_i = 1$ a $\sum_{j \in J} y_j = 1$, a $|I| + |J|$ rovnic, které zajistí, že očekávané výnosy jsou na podpoře rovné a maximální:

$$\sum_{i \in I} N_{j,i}^T x_i = v \quad \text{a} \quad \sum_{j \in J} M_{i,j} y_j = u,$$

kde $u = \max\{M_{i,*} y : i \in I\}$ a $v = \max\{N_{j,*} x : j \in J\}$ jsou dvě nové proměnné.

Jinak řečeno: Pro každou akci j hráče 2 ze supportu J musí být očekávaný výnos v konstantní, pokud hráč 1 hraje podle své smíšené strategie.

29.6 Nashovy ekvilibrium pomocí enumerace podpor II

Intuice: Tento snímek ukazuje, jak použít enumeraci podpor pro nalezení Nashových ekvilibrií v bimaticových hrách. Princip spočívá v procházení různých kombinací podpor strategií a následném řešení lineárního systému rovnic, který vznikne pro dané podpory. Pokud jsou řešení nezáporná, máme Nashovo ekvilibrium. Tento postup je založen na tom, že Nashova ekvilibria mají stejnou velikost podpor pro obě strany, což zjednoduší výpočet. Ačkoli je metoda kvadratická, nabízí efektivní algoritmus pro nalezení NE.

- Máme systém $S(I, J)$ s $|I| + |J| + 2$ proměnnými $x_1, \dots, x_{|I|}, y_1, \dots, y_{|J|}, u, v$ a $|I| + |J| + 2$ lineárními rovnicemi.
- Pokud jsou čísla v řešení všechna nezáporná, máme NE dle *podmínky nejlepší odpovědi*. Pokud je G nondegenerovaná, je takové řešení jedinečné (pokud existuje).
- Vyplývá z toho, že Nashova ekvilibria mají stejně velikosti podpor pro obě strany.
- Tento postup nabízí jednoduchý algoritmus pro nalezení NE ve hře G : projdeme všechny možné podpory $I \subseteq A_1$ a $J \subseteq A_2$ velikosti $k \in \{1, \dots, \min(m, n)\}$ a ověříme, zda podpory I a J vedou k NE vyřešením systému $S(I, J)$ lineárních rovnic.

- Výpočetní složitost je zhruba 4^n pro $m = n$.

Opakování: Brute-force algoritmus

Nashovo ekvilibrium lze nalézt pomocí *best response condition*, která pro hráče i definuje strategii x jako nejlepší odpověď na strategii y ostatních hráčů, pokud platí:

$$x_i > 0 \implies M_{i,*}y = \max(M_{k,*}y) \quad \forall k \in A_1,$$

kde M je matice výplat pro hráče 1 a y je smíšená strategie hráče 2. Podobně platí pro y jako nejlepší odpověď na x :

$$y_j > 0 \implies N_{j,*}x = \max(N_{k,*}x) \quad \forall k \in A_2,$$

kde N je matice výplat hráče 2 a x je smíšená strategie hráče 1.

Best Response Polyhedra

Abychom mohli najít Nashova ekvilibria efektivněji, definujeme *polyedry nejlepší odpovědi* pro oba hráče v bimatrixových hrách.

Polyedr pro hráče 1: Polyedr hráče 1, označme jej P , je množina bodů (x, v) , kde $x \in \mathbb{R}^m$ představuje smíšené strategie hráče 1 a $v \in \mathbb{R}$ je horní obálka očekávaných výnosů. Definujeme jej následovně:

$$P = \{(x, v) \in \mathbb{R}^m \times \mathbb{R} : x \geq 0, 1^\top x = 1, N^\top x \leq 1v\},$$

kde N je matice výplat pro hráče 2. Zde platí, že $1^\top x = 1$ zajišťuje, že x představuje platnou smíšenou strategii (součet všech pravděpodobností je 1) a $N^\top x \leq 1v$ omezuje očekávané výnosy.

Polyedr pro hráče 2: Analogicky definujeme polyedr pro hráče 2, označme jej Q , jako množinu bodů (y, u) , kde $y \in \mathbb{R}^n$ představuje smíšené strategie hráče 2 a $u \in \mathbb{R}$ je horní obálka výnosů pro hráče 2:

$$Q = \{(y, u) \in \mathbb{R}^n \times \mathbb{R} : y \geq 0, 1^\top y = 1, My \leq 1u\},$$

kde M je matice výplat pro hráče 1.

Význam podmínek: - Podmínka $My \leq 1u$ zajišťuje, že očekávané výnosy hráče 1 jsou omezeny hodnotou u . - Podobně podmínka $N^\top x \leq 1v$ omezuje výnosy hráče 2 na v .

Interpretace: Body P a Q tedy představují smíšené strategie, které tvoří horní obálku očekávaných výnosů obou hráčů. Nashovo ekvilibrium hledáme jako bod, kde se tyto polyedry protínají, což znamená, že žádný hráč nemůže dosáhnout lepšího výnosu změnou své strategie.

Labeling of Best Response Polyhedra - Intuice

V tomto kroku přidáváme do polyedrů hráčů označení (labely), které nám pomáhají lépe chápat, které body reprezentují určité strategie v rámci Nashova ekvilibria. Každý bod v polyedru může mít přidělený *label*, což je index strategie, která je důležitá pro danou pozici. Tímto způsobem identifikujeme, zda určitá strategie patří do Nashova ekvilibria nebo jestli odpovídá určité nejlepší odpovědi.

Labely jsou přidělovány v závislosti na tom, zda jsou určité strategie aktivní (tedy jejich pravděpodobnosti nejsou nulové), nebo zda určité výplaty odpovídají nejlepší odpovědi na strategii soupeře.

Labeling of Best Response Polyhedra - Formální popis

Definujeme polyedry P a Q pro hráče 1 a hráče 2, které jsme již zavedli:

$$P = \{(x, v) \in \mathbb{R}^m \times \mathbb{R} : x \geq 0, 1^\top x = 1, N^\top x \leq 1v\},$$

$$Q = \{(y, u) \in \mathbb{R}^n \times \mathbb{R} : y \geq 0, 1^\top y = 1, M y \leq 1u\}.$$

Označujeme body z polyedrů P a Q tak, že každému bodu přiřazujeme labely odpovídající strategiím, které jsou buď v Nashově ekvilibriu, nebo které tvoří nejlepší odpovědi na strategii soupeře.

Labely pro body v P : Bod $(x, v) \in P$ má label $i \in A_1 \cup A_2$, pokud platí jedna z následujících podmínek:

- $i \in A_1$ a $x_i = 0$, tj. i -tá strategie hráče 1 není v podpoře jeho smíšené strategie.
- $i \in A_2$ a $N_{i,*}^\top x = v$, tj. strategie i je nejlepší odpovědí hráče 2 na strategii x .

Labely pro body v Q : Podobně bod $(y, u) \in Q$ má label $i \in A_1 \cup A_2$, pokud platí jedna z následujících podmínek:

- $i \in A_2$ a $y_i = 0$, tj. i -tá strategie hráče 2 není v podpoře jeho smíšené strategie.
- $i \in A_1$ a $M_{i,*} y = u$, tj. strategie i je nejlepší odpovědí hráče 1 na strategii y .

Každý bod v P nebo Q může mít přiřazeno více labelů, protože může splňovat více z výše uvedených podmínek současně.

Best response polyhedra - Intuice na příkladu

V tomto příkladu máme dvě matice výplat hráčů 1 a 2. Matice M s řádky $[2, 0]$ a $[0, 1]$ a matici N s řádky $[1, 0]$ a $[0, 2]$. Tyto matice určují výplaty pro oba hráče v závislosti na jejich strategiích.

Cílem je vytvořit polyedry P a Q , které reprezentují nejlepší odpovědi každého hráče na strategie soupeře. Každý bod v těchto polyedrech odpovídá smíšené strategii hráče. Labely na těchto bodech pak indikují, které strategie jsou nejlepší odpovědí na strategie soupeře.

Krok 1: Polyedr P pro hráče 1

Pro hráče 1 platí následující podmínky:

$$P = \{(x_1, x_2, v) \in \mathbb{R}^2 \times \mathbb{R} : x_1, x_2 \geq 0, x_1 + x_2 = 1, x_1 \leq v, 2x_2 \leq v\}$$

Tyto podmínky nám říkají, že smíšené strategie x_1 a x_2 musí splňovat, že jejich součet je 1 (tj. hráč musí rozdělit pravděpodobnost mezi své strategie) a dále, že jejich výplaty jsou omezeny hodnotou v , která odpovídá nejlepší možné odpovědi na strategii hráče 2.

Krok 2: Polyedr Q pro hráče 2

Podobně pro hráče 2:

$$Q = \{(y_3, y_4, u) \in \mathbb{R}^2 \times \mathbb{R} : y_3, y_4 \geq 0, y_3 + y_4 = 1, 2y_3 \leq u, y_4 \leq u\}$$

Tento polyedr reprezentuje smíšené strategie hráče 2, kde y_3 a y_4 jsou pravděpodobnosti použití jednotlivých strategií a u odpovídá výplatě pro nejlepší odpověď hráče 1.

NE a Best response polyhedra - Intuice

Nyní se zaměříme na to, jak identifikovat Nashovo ekvilibrium (NE) v dané hře G pomocí tzv. "completely labeled" bodů v polyedrech P a Q . Každý bod v těchto polyedrech má určité labely, které reprezentují strategie hráčů. Pokud je bod v $P \times Q$ kompletne označený, tj. každý label $i \in A_1 \cup A_2$ se objeví v některém z polyedrů, jedná se o Nashovo ekvilibrium.

V podstatě, kompletne označení bodů znamená, že pro každou strategii existuje nejlepší odpověď, což zajišťuje, že žádný hráč nemá důvod odchýlit se od své strategie, což je definice Nashova ekvilibria.

NE a Best response polyhedra - Formálně

[2.27] Strategie s je Nashovo ekvilibrium v G , pokud a jen pokud pár $((x, v), (y, u)) \in P \times Q$ je kompletne označený, tj. každý label $i \in A_1 \cup A_2$ se objeví jako label bodu buď v (x, v) , nebo v (y, u) .

Podle *Best response condition*, pro každého hráče $i \in P$, smíšená strategie s_i je nejlepší odpověď na s_{-i} , pouze pokud všechny čisté strategie v podpoře s_i jsou nejlepší odpovědí na s_{-i} .

Pokud chybí label $i \in A_1$, znamená to, že buď čistá strategie $i \in \text{Supp}(s_1)$ ($x_i > 0$) nebyla nejlepší odpověď ($M_{i,*}y < u$). Analogicky pro $i \in A_2$.

Pokud jsou všechny labely přítomny, pak jsou s_1 a s_2 navzájem nejlepší odpovědi, protože každá čistá strategie je nejlepší odpověď, nebo není v podpoře. Tedy, s je NE.

Best response polytopes

Naše best response polyhedra mají některé zbytečné komplikace, protože jsou neomezené a obsahují u a v ve svých souřadnicích. Těchto komplikací se zbavíme za určitých mírných předpokladů.

- Předpokládejme, že matice M a N^\top jsou nezáporné a nemají žádný nulový sloupec. Toho lze docílit tak, že k výplatám přičteme nějakou velkou konstantu.
- Poté můžeme každou nerovnost $N_{i,*}^\top x \leq v$ vydělit v , kde x_i/v považujeme za novou proměnnou. Totéž uděláme i pro polyhedron Q . Tato úprava normalizuje výplaty na hodnotu 1.

Po této úpravě dostáváme následující **best response polytopes**:

$$P = \{x \in \mathbb{R}^m : x \geq 0, N^\top x \leq 1\}$$

Podobně **best response polytope** pro hráče 2 je definováno jako:

$$Q = \{y \in \mathbb{R}^n : y \geq 0, M y \leq 1\}$$

What did we get?

- Nerovnosti mají stejný význam: pokud $x_i \geq 0$ je omezující, pak $i \in A_1$ není v supportu a pokud $N_{j,*}^\top x \leq 1$ je omezující, pak $j \in A_2$ je nejlepší odpověď na s_1 . Analogicky pro Q .
- Z předpokladu o M a N vyplývá, že polyhedra P a Q jsou omezené (polytopy) a mají rozměry m a n .
- Nevýhoda: souřadnice x a y se nesčítají na 1. To můžeme opravit rescalem $x/(1^\top x)$ a $y/(1^\top y)$.
- Polytope P je v jednoznačné korespondenci s $P \setminus \{0\}$ pomocí projektivní transformace x/v . Podobně pro Q a $Q \setminus \{0\}$.
- Projektivní transformace zachovávají incidence, takže labely zůstávají stejné.

Corollary 2.30

Strategický profil (s_1, s_2) s mixovanými strategiemi x a y je Nashovým ekvilibriem hry G , pokud a pouze pokud bod $\left(\frac{x}{u_2(s)}, \frac{y}{u_1(s)}\right) \in P \times Q \setminus \{(0, 0)\}$ je kompletně označený.

NE in nondegenerate games

- Připomeňme si, že bimaticová hra je nedegradovaná, pokud existuje nejvýše k čistých nejlepších odpovědí na každou smíšenou strategii s supportem velikosti k .
- V těchto hrách Nashova ekvilibria odpovídají páru plně označených vrcholů.
 - Protože G je nedegradovaná, každý bod z P má nejvýše m štítků. To proto, že pokud má x support o velikosti k , pak x má nejvýše $m - k$ štítků v A_1 , a pokud by měl více než m štítků, pak by měl více než k nejlepších odpovědí v A_2 .
 - Analogicky, každý bod z Q má nejvýše n štítků.
- Proto jsou P a Q oba jednoduché polytopy (každý bod z P nebo Q , který je obsažen ve více než m nebo n facetách, má více než m nebo n štítků).
- Jelikož $\dim(P) = m$ a $\dim(Q) = n$, pouze vrcholy P a Q mohou mít m a n štítků.
- Podle Corollary 2.30 mohou být Nashova ekvilibria pouze ve vrcholech P a Q .

\Rightarrow Algoritmus pro nalezení Nashova ekvilibria: zkонтrolujte všechny páry vrcholů a jejich štítky!

Algorithm for finding NE with vertex enumeration

Algorithm 0.6: Vertex enumeration(G)

- **Input:** Nedegradovaná bimaticová hra G .
 - **Output:** Všechna Nashova ekvilibria G .
- každý pár vrcholů (x, y) z $(P \setminus \{0\}) \times (Q \setminus \{0\})$ (x, y) je plně označené vrátí $(x/(1^\top x), y/(1^\top y))$ jako Nashovo ekvilibrium
- Všechny vrcholy jednoduchého polytopu v \mathbb{R}^d s v vrcholy a N omezujícími nerovnicemi lze nalézt v čase $O(dNv)$ (Avis a Fukuda).
 - Pokud však $m = n$, nejlepší odpověď polytopech může mít až c^n vrcholů pro nějakou konstantu c , kde $1 < c < 2.9$.
 - Vyhledávání můžeme zrychlit procházením polytopů řízeným podle štítků, což umožňuje snížit množství procházených vrcholů.

The Lemke-Howson algorithm explained

- Protože je P jednoduchý a $\dim(P) = m$, vrcholy P jsou incidentní přesně na m facetek a m hran. Každý vrchol P tedy má m štítků a každá hrana P má $m - 1$ štítků. Analogicky to platí pro Q a n .
- Ponechání štítku $l \in A_1 \cup A_2$ ve vrcholu x z P znamená přechod po jediné hraně P , která je incidentní k x a která má všechny m štítků, které má x , kromě l . Druhý konec této hrany má stejné štítky jako x , pouze l je nahrazeno novým štítkem, což se nazývá zvednutí“.
- Algoritmus začíná v $(0, 0)$ a střídavě sleduje hrany P a Q .
- V prvním kroku vybere štítek $k \in A_1 \cup A_2$ a upustí ho. Poté se zvedne nový štítek l . Tento štítek l má duplikát v druhém polytopu. Tento duplikát l je upuštěn v dalším kroku, což vede k vyzvednutí nového štítku l' . Iterujeme a zastavíme se, když $l' = k$.
- Duplikovaný štítek je buď nová nejlepší odpověď, která dostane kladnou pravděpodobnost, nebo čistá strategie, jejíž pravděpodobnost se stala nulovou a přejdeme dál od její nejlepší odpovědní facety.

Lemke-Howson algoritmus: pseudokód

Algorithm 1 Lemke-Howson(G)

Vstup: Nedegenerovaná bimaticová hra G

Výstup: Nashovo ekvilibrium G

```

1:  $(x, y) \leftarrow (0, 0) \in \mathbb{R}^m \times \mathbb{R}^n$ 
2:  $k \leftarrow$  libovolný štítek z  $A_1 \cup A_2$ ,  $l \leftarrow k$ 
3: while true do
4:   V polytopu  $P$ : Odstraň  $l$  z  $x$  a předefinuj  $x$  jako nový vrchol
5:   Předefinuj  $l$  jako nově přidaný štítek. Přepni na  $Q$ 
6:   if  $l = k$  then zastav smyčku
7:   end if
8:   V polytopu  $Q$ : Odstraň  $l$  z  $y$  a předefinuj  $y$  jako nový vrchol
9:   Předefinuj  $l$  jako nově přidaný štítek. Přepni zpět na  $P$ 
10:  if  $l = k$  then zastav smyčku
11:  end if
12: end while
13: Výstup:  $(x/(1^T x), y/(1^T y))$  jako Nashovo ekvilibrium

```

Lemke-Howson algoritmus: intuice

Algoritmus Lemke-Howson je postup, který nám umožní najít Nashovo ekvilibrium ve hře dvou hráčů (bimatrix games) pomocí průchodu mezi dvěma polytopy. Nashovo ekvilibrium je stav, kdy žádný hráč nemá motivaci jednostranně

měnit svou strategii. Tento algoritmus začíná v bodě $(0, 0)$ a postupně odebírá a přidává štítky, čímž prochází cestou mezi vrcholy polytopů P a Q , dokud nedojde k řešení.

Formálně postupujeme tak, že v každém kroku algoritmu odstraníme štítek z jednoho vrcholu, přejdeme na sousední vrchol a přidáme nový štítek. Tento proces opakujeme mezi dvěma polytopy, dokud nenajdeme vrchol s úplnými štítky, což odpovídá Nashovu ekvilibriu.

Korektnost Lemke–Howson algoritmu I: intuice

Abychom dokázali, že algoritmus Lemke–Howson konverguje k Nashovu ekvilibriu, potřebujeme se ujistit, že po konečném počtu kroků skončí a nalezne smíšené strategie odpovídající Nashovu ekvilibriu. Algoritmus pracuje na základě procházení tzv. konfigurací grafu \mathcal{G} (graf je složený z cest a cyklů, protože každý vrchol má degree buď 1 nebo 2), který představuje téměř úplně označené vrcholy z polytopů $P \times Q$. Každý krok znamená změnu jednoho štítku, což nás posouvá po hranách tohoto grafu.

Korektnost Lemke–Howson algoritmu I

Lemke–Howson algoritmus zastaví po konečném počtu kroků a vrátí smíšené strategie, které jsou Nashovým ekvilibriem v G .

Důkaz: Nechť k je štítek vybraný v prvním kroku. Definujeme *konfigurační graf* \mathcal{G} , jehož vrcholy jsou tvořeny dvojicemi (x, y) z polytopů $P \times Q$, které jsou k -též úplně označené (tj. každý štítek z $A_1 \cup A_2 \setminus \{k\}$ je štítek x nebo y). Dvojice $\{(x, y), (x', y')\}$ je hranou grafu \mathcal{G} , pokud platí $(x = x' \text{ a } yy' \in E(Q))$ nebo $(xx' \in E(P) \text{ a } y = y')$. Očividně je \mathcal{G} konečný.

- Graf \mathcal{G} má pouze stupně 1 nebo 2 (tj. \mathcal{G} je nespojitá unie cest a cyklů).
 - Pokud má (x, y) všechny štítky z $A_1 \cup A_2$, je spojen pouze s jedním jiným vrcholem, protože přesně jeden z x a y má štítek k a štítek k můžeme odebrat pouze z tohoto jednoho vrcholu.
 - V opačném případě má (x, y) všechny štítky z $A_1 \cup A_2 \setminus \{k\}$ a existuje jediný štítek sdílený x a y . Pak je (x, y) sousedící se dvěma vrcholy, protože můžeme odebrat duplicitní štítek z x v P nebo y v Q .

Korektnost Lemke–Howson algoritmu II: intuice

Algoritmus začíná v bodě $(0, 0)$, což odpovídá "listu" v konfiguraci grafu \mathcal{G} . Postupně prochází po cestě v grafu, přičemž každý vrchol navštíví maximálně jednou. Protože graf má stupně 1 a 2, algoritmus po určitém počtu kroků skončí v jiném listu grafu, což znamená, že jsme nalezli Nashovo ekvilibrium.

Korektnost Lemke–Howson algoritmu II

- Algoritmus Lemke–Howson začíná v listu $(0, 0)$ cesty v \mathcal{G} .
- Poté postupuje po této cestě a nenavštíví žádný vrchol grafu dvakrát.
 - Další dvojice vrcholů na cestě je vždy jedinečná (přesuneme se na druhý koncový bod jedinečné hrany obsahující aktuální vrchol, který odpovídá odstraněnému štítku).
 - Pokud bychom navštívili stejný vrchol dvakrát, měl by tento vrchol stupeň větší než 2 v \mathcal{G} , protože jsme začali v listu.
- Algoritmus tedy skončí po konečném počtu kroků v jiném listu (x^*, y^*) cesty. Protože (x^*, y^*) je list v konfiguraci grafu, je zcela označen.
- Tento bod není tvaru $(x, 0)$ ani $(0, y)$ (*cvičení*).
- Podle *Věty 2.30* (x^*, y^*) odpovídá Nashovu ekvilibriu po reskalaci.

Poznámky k Lemke–Howson algoritmu

Nedegenerovaná bimaticová hra má lichý počet Nashových ekvilibrií.

- Degenerované hry mohou mít nekonečně mnoho Nashových ekvilibrií.
- Algoritmus Lemke–Howson vždy najde pouze jedno Nashovo ekvilibrium.
- Algoritmus může být implementován pomocí tzv. doplňkového pivotování (complementary pivoting) – tím se budeme zabývat na cvičeních.
- Výpočetní složitost může být stále exponenciální ($O(2^n)$) kroků pro $n = m$). V praxi ale funguje dobře, a to s polynomiální složitostí na náhodně generovaných hrách.
- Existuje efektivní algoritmus pro hledání Nashových ekvilibrií?

30 Kam NASH patří?

Je NASH NP-úplný?

- Ne. NP je třída rozhodovacích problémů (ano/ne odpovědi), zatímco NE vždy existuje (takže odpověď je vždy ano).

Další kandidát je komplexitní třída FNP (funkcionální NP“).

- Vstupem problému FNP je instance problému z NP. Algoritmus vrací řešení, pokud existuje. Pokud řešení neexistuje, algoritmus vrátí ne“.
- Jinými slovy, požadujeme řešení pro instance s odpovědí ano“.

- NASH patří do FNP, protože ověřování, zda je profil strategie NE, lze provést pomocí podmínky nejlepší odpovědi (Best Response Condition).
- Je NASH FNP-úplný? Nepravděpodobné kvůli následujícímu tvrzení.

Věta 2.34 (Megiddo a Papadimitriou, 1991) Pokud je problém NASH FNP-úplný, potom NP = coNP.

Nová komplexitní třída

- Důkaz správnosti algoritmu Lemke–Howson odhaluje strukturu problému NASH (hledání dalšího koncového bodu cesty v grafu s maximálním stupněm 2).
- Zachytíme tuto abstraktní strukturu.

Problém END-OF-THE-LINE Pro orientovaný graf G s každým vrcholem majícím nejvýše jednoho předchůdce a jednoho následníka, je pro daný vrchol s grafu G bez předchůdce najít vrchol $t \neq s$ bez předchůdce nebo následníka. Graf G není dán ve vstupu, ale je specifikován nějakou polynomiálně časově spočitatelnou funkcí $f(v)$, která vrací předchůdce a následníka (pokud existují) vrcholu v .

- Graf G tedy může být exponenciálně velký vzhledem k velikosti vstupu.

PPAD Nechť PPAD je komplexitní třída sestávající z problémů, které umožňují polynomiální redukci na END-OF-THE-LINE.

NASH a PPAD

- Důkaz správnosti algoritmu Lemke–Howson ukazuje, že NASH patří do třídy PPAD (pro nedegenerované hry).
- Je NASH PPAD-úplný?
 - Jinými slovy, je mezi nejtěžšími problémy v této třídě?
 - PPAD-úplnost poskytuje určitý důkaz o výpočetní neřešitelnosti, i když slabší než NP-úplnost.
 - Otevřený problém po dlouhou dobu.

Věta 2.35 (Chen, Deng, Teng, Daskalakis, Goldberg a Papadimitriou, 2009) Problém NASH je PPAD-úplný.

- Jeden z hlavních průlomů v algoritmické teorii her.
- Důkaz vynecháme, protože je složitý (články mají přes 50 a 70 stran).

NASH a PPAD

Intuice: Dokazování správnosti algoritmu Lemke–Howson nám odhaluje, že problém NASH patří do třídy PPAD, což zahrnuje neoddegenerované bimaticové hry. Tento objev vedl k otázce, zda je problém NASH kompletní pro PPAD, tedy jestli patří mezi nejtěžší problémy v této třídě. PPAD-kompletnost by naznačovala jistou míru výpočetní složitosti, která je sice slabší než NP-kompletnost, ale stále představuje určitou výzvu.

Formální zápis: [Chen, Deng, Teng, Daskalakis, Goldberg, Papadimitriou, 2009] Problém NASH je PPAD-kompletní.

Důsledek: Tento objev je jedním z hlavních průlomů v algoritmické teorii her. Dokazování se zde vynechává, neboť je značně složité (originální důkaz má přes 50 až 70 stran).

Nové koncepty řešení

Intuice: Hledání Nashova ekvilibria je výpočetně náročné, pokud není PPAD podmnožinou FP. Proto se hledají jiné koncepty řešení, které by byly výpočetně zvládnutelné. Dva takové koncepty jsou ε -Nashovo ekvilibrium a korelované ekvilibrium.

Formální zápis:

- **ε -Nashovo ekvilibrium:** Tento koncept je intuitivní a snadno pochopitelný. Definice je jednoduchá a ε -NE existuje vždy podle Nashovy věty.
- **Korelované ekvilibrium:** Tento koncept má komplikovanější definici, ale postupně oceníme jeho přirozenost, která by mohla být ještě silnější než u běžného NE.

ε -Nashovo ekvilibrium

Intuice: Pokud pro každého hráče platí, že žádná jiná strategie mu nemůže zvýšit užitek o více než ε , mluvíme o ε -Nashově ekvilibriu. V limitním případě, kdy $\varepsilon = 0$, dostáváme standardní Nashovo ekvilibrium. Tento koncept je výhodný díky jednoduché definici a existenci ε -NE v každé hře. Nevýhodou však je, že některé ε -NE nemusí být blízko žádnému skutečnému NE, což snižuje jeho přesnost jako přibližného řešení.

Formální zápis: Pro $\varepsilon > 0$, profil strategií $s = (s_1, \dots, s_n)$ v normální formě hry $G = (P, A, u)$ je ε -Nashovo ekvilibrium (ε -NE), pokud pro každého hráče $i \in P$ a každou jeho strategii $s'_i \in S_i$ platí:

$$u_i(s_i; s_{-i}) \geq u_i(s'_i; s_{-i}) - \varepsilon.$$

- Výhody:
 - Snadno pochopitelná definice.
 - ε -NE existuje vždy díky Nashově větě.
 - Použití ε jako přesnosti nám umožňuje vyhnout se práci s iracionálními čísly.
- Nevýhody:
 - Některá ε -NE nemusí být blízko žádnému skutečnému NE.
 - I tento koncept je výpočetně náročný.

Algoritmické aspekty ε -Nashova ekvilibria

Intuice Výpočet ε -Nashova ekvilibria je náročný problém, proto se zavádějí různé koncepty optimalizace algoritmů jako PTAS a FPTAS, které by mohly umožnit rychlejší výpočty. Bohužel, existence plně polynomiálního algoritmu pro ε -Nash ekvilibrium je otevřený problém.

Formální zápis

- Optimalizační problém P s velikostí vstupu n a parametrem $\varepsilon > 0$ má PTAS, pokud existuje algoritmus, který počítá ε -přesné řešení v čase $O(n^{f(1/\varepsilon)})$ pro nějakou funkci f .
- Problém P má FPTAS, pokud existuje algoritmus, který běží v čase $O((1/\varepsilon)^c n^d)$ pro nějaké konstanty c a d .
- Existuje FPTAS pro ε -NE?
 - Ne, pokud neplatí $PPAD \subseteq FP$ (Chen, Deng, a Teng, 2006).
- Existuje PTAS pro ε -NE?
 - Otevřený problém!
- Máme kvazi-polynomiální algoritmus:

[Lipton, Markakis a Mehta, 2003] Nechť $G = (P, A, u)$ je hra ve standardní formě pro dva hráče, každý s m akcemi, kde matice výplat obsahuje hodnoty v $[0, 1]$. Pro každé $\varepsilon > 0$ existuje algoritmus, který vypočítá ε -NE pro G v čase $m^{O(\log m/\varepsilon^2)}$.

Korelované ekvilibrium

Intuice Korelované ekvilibrium (CE) je další koncept řešení, který může být přirozenější než Nashovo ekvilibrium. Představme si důvěryhodnou třetí stranu, která hráčům doporučuje strategie podle předem známé distribuční pravděpodobnosti. Hráči mohou, ale nemusí doporučení dodržet; CE nastává, pokud každý hráč maximalizuje svůj očekávaný užitek při dodržení tohoto doporučení.

Formální zápis

- Nejzákladnější koncept řešení podle několika odborníků.
- ”Pokud existuje inteligentní život na jiných planetách, většina z nich by pravděpodobně objevila korelované ekvilibrium před Nashovým ekvilibrium.“ (Myerson)
- Ve hře $G = (P, A, u)$ nechť p je distribuční pravděpodobnost na A , tedy $p(a) \geq 0$ pro každé $a \in A$ a $\sum_{a \in A} p(a) = 1$. Distribuce p je korelované ekvilibrium (CE) ve hře G pokud

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

pro každého hráče $i \in P$ a pro všechny čisté strategie $a_i, a'_i \in A_i$.

Scénář: Představte si důvěryhodnou třetí stranu, která veřejně zná distribuci p . Třetí strana náhodně vybere akci $a \in A$ podle p a hráči i soukromě doporučí strategii a_i , aniž by prozradila ostatní akce a_{-i} . Pokud každý hráč maximalizuje svůj očekávaný užitek tím, že dodrží doporučení, je p korelovaným ekvilibrium.

Korelovaná rovnováha na příkladech

Hra na kuře

Hru na kuře lze ilustrovat následující tabulkou výplat:

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

Existují dvě čisté Nashovy rovnováhy, kdy (Stop, Go) nebo (Go, Stop), a jedna smíšená Nashova rovnováha s pravděpodobnostmi $(9/10, 9/10)$.

Uvažujme důvěryhodnou třetí stranu, například semafor, který náhodně vybírá (Stop, Stop), (Stop, Go) a (Go, Stop) s pravděpodobností $1/3$. Tento semafor představuje korelovanou rovnováhu (CE).

- Pokud první hráč dodrží návrh Go“, získá 1, zatímco odchylka by mu dala 0.
- Pokud první hráč dodrží návrh Stop“, získá -1 , zatímco odchylka by mu dala $-\frac{9}{2}$.
- Symetricky, druhý hráč také neodbočuje od doporučené strategie.

Boj pohlaví

Boj pohlaví lze ilustrovat následující tabulkou výplat:

	Fotbal	Opera
Fotbal	(2, 1)	(0, 0)
Opera	(0, 0)	(1, 2)

Existují dvě čisté Nashovy rovnováhy, kdy (Fotbal, Fotbal) nebo (Opera, Opera), a jedna smíšená Nashova rovnováha s pravděpodobnostmi $(2/3, 2/3)$.

Uvažujme důvěryhodnou třetí stranu, například tchyni, která hodí minci a náhodně vybere (Fotbal, Fotbal) nebo (Opera, Opera) s pravděpodobností $1/2$. Tato třetí strana představuje korelovanou rovnováhu (CE).

- Pokud manžel dodrží návrh Fotbal“, získá 2, zatímco odchylka by mu dala 0.
- Pokud manžel dodrží návrh Opera“, získá 1, zatímco odchylka by mu dala 0.
- Symetricky, manželka také neodbočuje od doporučené strategie.

Výhody a nevýhody korelované rovnováhy

Nevýhody:

- Definice korelované rovnováhy (CE) může být složitější na pochopení.

Výhody:

- Každá Nashova rovnováha (NE) je také korelovaná rovnováha (cvičení). Tedy korelovaná rovnováha vždy existuje dle Nashovy věty.
- Každá Nashova rovnováha s je korelovaná rovnováha s rozdelením $p = \prod_{i=1}^n s_i$. Takže korelovaná rovnováha může poskytovat lepší výplaty než Nashova rovnováha.
- Korelovaná rovnováha může být vypočítána v polynomálním čase pomocí lineárního programování (LP). Uvažme následující LP s proměnnými $p(a)$ pro $a \in A$:

$$\max \left\{ \sum_{i \in P} \left(\sum_{a \in A} u_i(a)p(a) \right) \right\}$$

s podmínkami pro všechna $i \in P$, $a_i, a'_i \in A_i$:

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

$$\sum_{a \in A} p(a) = 1, \quad p(a) \geq 0 \quad \text{pro všechna } a \in A.$$

31 Minimalizace regretu

Minimalizace regretu představuje zcela nový model interakcí, kde se zaměřujeme na snižování "regretu" (lítosti), tedy rozdílu mezi skutečně dosaženými výsledky a nejlepšími možnými výsledky. Tento model je založen na tzv. *online učení*, kde se agent učí na základě zpětné vazby z předchozích rozhodnutí.

- V budoucnu aplikujeme tyto nové metody k návrhu rychlých algoritmů pro výpočet *aproximovaných korelovaných ekvilibrií*.
- Dnes si představíme model a některé základní algoritmy, které minimalizují regret.

Intuitivně lze říci, že minimalizace regretu znamená optimalizovat své chování tak, aby se agent v dlouhodobém horizontu co nejméně "litoval" svých rozhodnutí. Když agent udělá chybu, může se z ní poučit, což vede k postupnému zlepšování jeho rozhodnutí.

31.1 Nastavení modelu

Při zavádění nového modelu je důležité stanovit některé základní označení a pojmy.

- Máme agenta A , který operuje v prostředí s protivníkem (adversary).
- Agent A má k dispozici N možných akcí, které jsou obsaženy v množině $X = \{1, \dots, N\}$.
- V každém kroku $t = 1, \dots, T$:
 - Agent A vybere pravděpodobnostní rozdělení $p^t = (p_1^t, \dots, p_N^t)$ na množině X , kde p_i^t je pravděpodobnost, že agent A zvolí akci i v kroku t .
 - Následně protivník zvolí vektor ztrát $\ell^t = (\ell_1^t, \dots, \ell_N^t)$, kde $\ell_i^t \in [-1, 1]$ reprezentuje ztrátu spojenou s akcí i v kroku t .
 - Agent A poté zažije očekávanou ztrátu $\ell_A^t = \sum_{i=1}^N p_i^t \ell_i^t$ v kroku t . Tato hodnota představuje očekávanou ztrátu agenta A v daném kroku.
- Po T krocích je kumulativní ztráta akce i vyjádřena jako $L_i^T = \sum_{t=1}^T \ell_i^t$.
- Kumulativní ztráta agenta A je $L_A^T = \sum_{t=1}^T \ell_A^t$.

Intuitivně lze toto nastavení popsat tak, že agent A se snaží vybírat své akce tak, aby minimalizoval očekávanou ztrátu (loss) na základě zpětné vazby, kterou obdrží v každém kroku od protivníka. Tento model je základním krokem pro pochopení minimalizace regretu, protože agent se učí optimalizovat své chování v závislosti na minulých rozhodnutích a jejich výsledcích.

31.2 Externí regret

Abychom mohli posoudit, jak dobře si náš agent vede, zvolíme takzvaný "externí přístup", kdy srovnáváme jeho kumulativní ztrátu se ztrátou nejlepšího agenta z určité porovnávací třídy \mathcal{A} .

- Budeme převážně uvažovat třídu $\mathcal{A}_X = \{A_i : i \in X\}$, kde agent A_i vždy vybírá akci i .
- Označme $R_A^T = L_A^T - \min\{L_B^T : B \in \mathcal{A}_X\}$ jako externí regret agenta A . Tedy $R_A^T = L_A^T - \min\{L_i^T : i \in X\}$.
- Pokud není uvedeno jinak, uvažujeme pouze ztrátové vektory z množiny $\{0, 1\}^N$. Toto zjednoduší zápis, přičemž všechny prezentované výsledky lze rozšířit na obecnější případ.

Intuitivní vysvětlení Externí regret je metrika, která měří, jak moc ztráty agenta A převyšují ztráty nejlepšího možného rozhodnutí (akce) po celou dobu T kroků. Tento koncept nám umožňuje zhodnotit, zda agentovo chování vedlo k vyšší nebo nižší ztrátě ve srovnání s hypotetickým agentem, který by vždy vybíral tu nejlepší možnou akci (v rámci dané třídy).

Příklad - Učení bez regretru (No Regret Learning)

Tento příklad ilustruje princip učení bez regretru v kontextu volby mezi dvěma akcemi v závislosti na předpovědi počasí.

- Máme agenta, který se snaží předpovědět, zda bude pršet nebo bude slunečno, a podle toho se rozhodnout, jaké opatření podniknout. Agent má na výběr dvě možnosti:
 - **Sunscreen (opalovací krém)**: vhodné pro slunečné dny.
 - **Umbrella (deštník)**: vhodné pro deštivé dny.
- V příkladu máme sérii dní, kde počasí je buď slunečné nebo deštivé. Počasí je znázorněno v prvním řádku tabulky s ikonami (slunce pro slunečno a mrak pro déšť).
- Agent vybírá mezi opalovacím krémem a deštníkem, snaží se minimalizovat svůj **loss** (ztrátu), která závisí na správnosti jeho volby v porovnání se skutečným počasím.

Tabulka ztrát

- Každá kombinace akce a počasí má přiřazenou hodnotu ztráty:
 - Pokud je vybrán opalovací krém a je slunečno, ztráta je 0 (správné rozhodnutí).

- Pokud je vybrán opalovací krém a prší, ztráta je 1 (špatné rozhodnutí).
 - Pokud je vybrán deštník a prší, ztráta je 0 (správné rozhodnutí).
 - Pokud je vybrán deštník a je slunečno, ztráta je 1 (špatné rozhodnutí).
- Cílem agenta je minimalizovat svoji ztrátu během více kroků (dnů) tak, aby žádná konkrétní akce dlouhodobě nevedla k nižší ztrátě než dynamický výběr akcí na základě předpovědí.

Interpretace výsledků

Na konci každého dne agent zjistí, jakou ztrátu utrpěl na základě skutečného počasí. Postupným učením a úpravou své strategie se agent snaží minimalizovat kumulativní ztrátu, čímž dosahuje toho, že žádná jednotlivá akce (opalovací krém nebo deštník) dlouhodobě nepřekonává jeho adaptivní přístup.

Je tento model příliš restriktivní?

Na první pohled se může zdát, že třída agentů \mathcal{A}_X obsahuje příliš jednoduché agenty. Nicméně ukážeme, že rozšíření této třídy na větší srovnávací třídy vede k výrazně vyššímu regretru.

- Označme \mathcal{A}_{all} množinu agentů, kteří v každém kroku přiřadí pravděpodobnost 1 nějaké akci z množiny X .
 - V třídě \mathcal{A}_X musí každý agent volit vždy stejnou akci ve všech krocích.

Pozorování 2.45

Pro libovolného agenta A a každé $T \in \mathbb{N}$ existuje posloupnost T vektorů ztrát a agent $B \in \mathcal{A}_{all}$ takový, že

$$L_A^T - L_B^T \geq T \left(1 - \frac{1}{N} \right).$$

To znamená, že rozdíl mezi kumulativní ztrátou agenta A a optimální kumulativní ztrátou agenta z třídy \mathcal{A}_{all} může být téměř maximální možný.

Důkaz:

- Pro každý časový krok t označme i_t akci s nejnižší pravděpodobností $p_{i_t}^t$.
- Nastavíme $\ell_{i_t}^t = 0$ a $\ell_i^t = 1$ pro každé $i \neq i_t$.
- Protože $p_{i_t}^t \leq \frac{1}{N}$, máme $\ell_A^t \geq 1 - \frac{1}{N}$, a tedy kumulativní ztráta L_A^T po T krocích je alespoň $T \left(1 - \frac{1}{N} \right)$.
- Algoritmus $B \in \mathcal{A}_{all}$, který v každém kroku volí akci i_t s pravděpodobností 1, má pak kumulativní ztrátu $L_B^T = 0$.

Intuitivní vysvětlení

V tomto pozorování jde o ukázku toho, že pokud rozšíříme srovnávací třídu agentů na \mathcal{A}_{all} (tj. umožníme agentům v každém kroku vybrat libovolnou akci s pravděpodobností 1), může agent dosáhnout výrazně nižší ztráty. To vede k vysokému regretnímu pro našeho agenta A , který musí postupně volit akce dle pravděpodobnostní distribuce. Naopak, agent z třídy \mathcal{A}_{all} může v každém kroku bezchybně vybrat akci, což mu zajistí minimální kumulativní ztrátu.

Greedy algoritmus

Nyní máme srovnávací třídu \mathcal{A}_X a chceme navrhnout agenta A , který se s touto třídou dobře srovná. Zkusíme tedy použít přirozený přístup greedy (chamtvý) strategie: zvolíme akci $i \in X$, pro kterou je kumulativní ztráta L_i^{t-1} v kroku $t - 1$ nejmenší.

Algorithm 2 Greedy algoritmus (X, T)

Input: Množina akcí $X = \{1, \dots, N\}$ a počet kroků $T \in \mathbb{N}$.

Output: Pravděpodobnostní distribuce p^t pro každý krok $t \in \{1, \dots, T\}$.

```

1:  $p^1 \leftarrow (1, 0, \dots, 0)$ 
2: for  $t = 2, \dots, T$  do
3:    $L_{\min}^{t-1} \leftarrow \min_{j \in X} \{L_j^{t-1}\}$ 
4:    $S^{t-1} \leftarrow \{i \in X : L_i^{t-1} = L_{\min}^{t-1}\}$ 
5:    $k \leftarrow \min S^{t-1}$ 
6:    $p_k^t \leftarrow 1, p_i^t \leftarrow 0$  pro  $i \neq k$ 
7: end for
8: Output  $\{p^t : t \in \{1, \dots, T\}\}$ 
```

Intuitivní vysvětlení

Greedy algoritmus funguje na základě jednoduchého principu: v každém kroku zvolíme tu akci, která měla dosud nejnižší kumulativní ztrátu. Tím se snažíme minimalizovat celkovou ztrátu postupem času, protože dáváme přednost akcím, které se dosud ukázaly jako méně nákladné. Tento přístup však může být krátkozraký, protože nebude v úvalu možné dlouhodobé dopady volby akcí. Greedy algoritmus je tedy jednoduchý a efektivní pro některé situace, ale nemusí být optimální ve všech případech.

Analýza Greedy algoritmu

Pro libovolnou posloupnost vektorů ztrát s hodnotami z $\{0, 1\}$, kumulativní ztráta L_{Greedy}^T Greedy algoritmu v čase $T \in \mathbb{N}$ splňuje:

$$L_{Greedy}^T \leq N \cdot L_{\min}^T + (N - 1).$$

Důkaz:

V kroku t , pokud Greedy algoritmus způsobí ztrátu 1 a L_{min}^t se nezvýší, pak alespoň jedna akce zmizí z množiny S^t v dalším kroku. K tomu může dojít maximálně N -krát, než se L_{min}^t zvýší o 1.

Tímto způsobem Greedy algoritmus způsobí maximální ztrátu N mezi po sobě jdoucími nárůsty L_{min}^t o 1. Z toho vyplývá:

$$L_{Greedy}^T \leq N \cdot L_{min}^T + N - |S^T| \leq N \cdot L_{min}^T + (N - 1).$$

Intuitivní vysvětlení:

Tato nerovnost ukazuje, že Greedy algoritmus může dosáhnout kumulativní ztráty, která je zhruba N -krát horší než ztráta nejlepší akce. Tento výsledek je relativně slabý, protože ukazuje, že Greedy algoritmus nemusí být vždy optimální, a jeho výkon může být výrazně horší než u nejlepší akce v dlouhodobém horizontu.

Randomizovaný Greedy algoritmus

- Špatné chování Greedy algoritmu má své opodstatnění. Žádný deterministický algoritmus nemůže dosáhnout výrazně lepšího výkonu.
- Proto dává smysl zavést do algoritmu náhodnost. Při shodných hodnotách vybíráme náhodně a rozdělujeme váhy mezi aktuálně nejlepší akce.

Algorithm 3 Randomizovaný Greedy algoritmus (X, T)

Require: Množina akcí $X = \{1, \dots, N\}$ a počet kroků $T \in \mathbb{N}$

Ensure: Pravděpodobnostní distribuce p^t pro každý $t \in \{1, \dots, T\}$

```

1:  $p^1 \leftarrow (\frac{1}{N}, \dots, \frac{1}{N})$ 
2: for  $t = 2, \dots, T$  do
3:    $L_{min}^{t-1} \leftarrow \min_{j \in X} \{L_j^{t-1}\}$ 
4:    $S^{t-1} \leftarrow \{i \in X : L_i^{t-1} = L_{min}^{t-1}\}$ 
5:   for  $i \in S^{t-1}$  do
6:      $p_i^t \leftarrow \frac{1}{|S^{t-1}|}$ 
7:   end for
8:   for  $i \notin S^{t-1}$  do
9:      $p_i^t \leftarrow 0$ 
10:  end for
11: end for
12: return  $\{p^t : t \in \{1, \dots, T\}\}$ 

```

Intuitivní vysvětlení:

Randomizovaný Greedy algoritmus se snaží vylepšit výsledky tím, že přidává náhodnost při výběru mezi akcemi s minimální kumulativní ztrátou. Tímto

způsobem algoritmus lépe zvládá situace, kdy více akcí má stejnou nejnižší ztrátu, a snaží se tím omezit opakování vybírání stejných akcí, což by mohlo vést k horším výsledkům. Tento přístup má za cíl lépe vyvážit rizika a přínosy výběru akcí.

Analýza Randomizovaného Greedy algoritmu

[2.48] Pro libovolnou posloupnost vektorů ztrát s hodnotami v $\{0, 1\}$, kumulativní ztráta L_{RG}^T Randomizovaného Greedy algoritmu v čase $T \in \mathbb{N}$ splňuje

$$L_{RG}^T \leq (1 + \ln N) \cdot L_{min}^T + \ln N.$$

Důkaz (náčrt): Postupujeme podobně jako v předchozím důkazu. Pro $j \in \mathbb{N}$ označme t_j jako krok t , ve kterém ztráta L_{min}^t poprvé dosáhne hodnoty j . Odhadujeme ztrátu algoritmu mezi kroky t_j a t_{j+1} .

- Poznamenejme, že $1 \leq |S^t| \leq N$. Pokud se velikost S^t sníží o k z n' na $n' - k$ v nějakém čase $t \in (t_j, t_{j+1}]$, pak je ztráta algoritmu v kroku t rovna $\frac{k}{n'}$, protože váha každé takové akce je $\frac{1}{n'}$.
- Zjevně platí, že $\frac{k}{n'} \leq \frac{1}{n'} + \frac{1}{(n'-1)} + \cdots + \frac{1}{(n'-k+1)}$, takže ztráta za celý časový interval $(t_j, t_{j+1}]$ je nejvýše

$$\frac{1}{N} + \frac{1}{(N-1)} + \cdots + \frac{1}{1} \leq 1 + \ln N.$$

Z toho plyne, že

$$L_{RG}^T \leq (1 + \ln N) \cdot L_{min}^T + \left(\frac{1}{N} + \frac{1}{(N-1)} + \cdots + \frac{1}{|S^T| + 1} \right).$$

Intuitivní vysvětlení: Tato analýza ukazuje, že Randomizovaný Greedy algoritmus má ztrátu, která je limitována logaritmickou funkcí velikosti akčního prostoru N a minimální ztrátou L_{min}^T . Jinými slovy, algoritmus se snaží minimalizovat svou kumulativní ztrátu tím, že náhodně volí mezi akcemi s nejnižší ztrátou. Tento přístup vede k tomu, že celková ztráta algoritmu roste pouze velmi pomalu s počtem možných akcí, což je výhodné při velkých hodnotách N .

Polynomial Weights Algoritmus

Motivace: I když Randomizovaný Greedy algoritmus vylepsil původní Greedy algoritmus, stále není optimální. Ztráty jsou největší, když jsou množiny S^t malé, protože ztrátu lze považovat za úměrnou $1/|S^t|$. Tento problém překonáváme přiřazením větších vah akcím, které jsou blízko nejlepší akce.

Algoritmus 0.37: Polynomial Weights Algoritmus

Algorithm 4 Polynomial Weights Algorithm (X, T, η)

Require: Množina akcí $X = \{1, \dots, N\}$, počet kroků $T \in \mathbb{N}$, a $\eta \in (0, 1/2]$

Ensure: Pravděpodobnostní distribuce p^t pro každý $t \in \{1, \dots, T\}$

- 1: $w_i^1 \leftarrow 1$ pro každé $i \in X$
 - 2: $p^1 \leftarrow (1/N, \dots, 1/N)$
 - 3: **for** $t = 2, \dots, T$ **do**
 - 4: $w_i^t \leftarrow w_i^{t-1} (1 - \eta \ell_i^{t-1})$ pro každé $i \in X$
 - 5: $W^t \leftarrow \sum_{i \in X} w_i^t$
 - 6: $p_i^t \leftarrow w_i^t / W^t$ pro každé $i \in X$
 - 7: **end for**
 - 8: **return** $\{p^t : t \in \{1, \dots, T\}\}$
-

Intuitivní vysvětlení: Polynomial Weights algoritmus přiřazuje vyšší váhy těm akcím, které mají lepší výkon (tj. menší ztrátu) v předchozích krocích. Každý krok t aktualizuje váhy w_i^t na základě ztráty ℓ_i^{t-1} a nastavuje novou pravděpodobnostní distribuci p^t pro výběr akcí v následujícím kroku. Tento přístup zajišťuje, že akce s nižší kumulativní ztrátou mají vyšší pravděpodobnost výběru, což vede k lepším výsledkům v dlouhodobém horizontu.

31.3 Analýza algoritmu s polynomiálními váhami

31.3.1 Věta 2.49

Pro $\eta \in (0, 1/2]$ a pro každou sekvenci hodnot ztrát $\ell^t \in [-1, 1]$ pro $t = 1, \dots, T$ platí, že kumulativní ztráta algoritmu s polynomiálními váhami, označená jako L_{PW}^T , splňuje:

$$L_{PW}^T \leq L_k^T + \eta Q_k^T + \frac{\ln N}{\eta},$$

kde $Q_k^T = \sum_{t=1}^T (\ell_k^t)^2$. V případě, že $T \geq 4 \ln N$, při volbě $\eta = \sqrt{\frac{\ln N}{T}}$ a s vědomím, že $Q_k^T \leq T$, dostaneme následující odhad:

$$L_{PW}^T \leq L_{\min}^T + 2\sqrt{T \ln N}.$$

31.3.2 Důkaz (náčrt)

Dokážeme, že pokud je ztráta významná, pak celková váha W^t musí výrazně klesnout.

- Pro krok t definujeme $\ell_{PW}^t = \sum_{i=1}^N w_i^t \ell_i^t / W^t$, což je očekávaná ztráta v kroku t .
- Váha w_i^t každé akce i je násobena $(1 - \eta \ell_i^{t-1})$ v kroku t . Proto platí

$$W^{t+1} = W^t - \sum_{i=1}^N \eta w_i^t \ell_i^t = W^t (1 - \eta \ell_{PW}^t).$$

31.3.3 Intuitivní vysvětlení

V podstatě jde o to, že pokud některé akce způsobují vysoké ztráty, jejich váhy se postupně snižují, což znamená, že se budou vybírat méně často. Tento proces zajišťuje, že algoritmus se přizpůsobí a bude preferovat akce, které vedou k nižším ztrátám. Tento mechanismus je klíčový pro minimalizaci kumulativní ztráty.

31.3.4 Pokračování důkazu: Horní a dolní mez pro W^{T+1}

Použitím vlastností váhy a ztráty získáme pro horní mez:

$$\ln W^{T+1} \leq \ln N - \eta \sum_{t=1}^T \ell_{PW}^t.$$

Pro dolní mez platí:

$$\ln W^{T+1} \geq \ln w_k^{T+1} = \sum_{t=1}^T \ln(1 - \eta \ell_k^t) \geq -\eta L_k^T - \eta^2 Q_k^T.$$

31.3.5 Kombinace výsledků

Kombinací horní a dolní meze dostáváme:

$$-\eta L_k^T - \eta^2 Q_k^T \leq \ln N - \eta L_{PW}^T,$$

což dokazuje, že algoritmus s polynomiálními váhami skutečně efektivně minimalizuje kumulativní ztrátu v porovnání s nejlepší možnou akcí.

31.4 Poznámky k algoritmu s polynomiálními váhami

- Tento algoritmus dosahuje velmi dobrých výsledků z hlediska vnějšího regredu. Časově zprůměrovaný vnější regret se blíží nule.
- Nerovnost

$$L_{PW}^T \leq L_{\min}^T + 2\sqrt{T \ln N}$$

je v zásadě optimální.

31.4.1 Propozice 2.50

Pro celá čísla N a T , kde $T < \lfloor \log_2 N \rfloor$, existuje stochastické generování ztrát, takové že pro každý online algoritmus A platí:

$$\mathbb{E}[L_A^T] \geq \frac{T}{2} \quad \text{a současně} \quad L_{\min}^T = 0.$$

31.4.2 Propozice 2.51

V případě, že $N = 2$ (dvě akce), existuje stochastické generování ztrát, takové že pro každý online algoritmus A platí:

$$\mathbb{E}[L_A^T - L_{\min}^T] \geq \Omega(\sqrt{T}).$$

- Podrobnosti důkazů naleznete v poznámkách k přednáškám.
- K tomu, aby algoritmus fungoval, není nutné znát hodnotu T předem (cvičení).

31.5 Aplikace a další algoritmy

- Kromě teorie her má *metoda násobného přepočtu vah* mnoho aplikací v různých oblastech vědy, například v optimalizaci, teoretické informatice a strojovém učení.
- Více informací lze nalézt na stránce Wikipedie o metodě násobného přepočtu vah.
- Existují i další algoritmy, které produkují malý vnější regret, například *algoritmus párování regretru* (Regret matching algorithm).

31.6 Dynamika bez regretru

- Hráči v normální formě hry hrají proti sobě a volí akce podle algoritmu s polynomiálními vahami.

Algorithm 5 No-regret dynamics(G, T, ϵ)

Require: Normální forma hry $G = (P, A, C)$ s n hráči, $T \in \mathbb{N}$ a $\epsilon > 0$

Ensure: Pravděpodobnostní distribuce p_i^t na A_i pro každého $i \in P$ a $t \in \{1, \dots, T\}$

- 1: **for** každý krok $t = 1, \dots, T$ **do**
- 2: **for** každého hráče $i \in P$ nezávisle **do**
- 3: Hráč i si vybere smíšenou strategii p_i^t pomocí algoritmu s průměrným regretem maximálně ϵ , přičemž akce odpovídají čistým strategiím.
- 4: **end for**
- 5: **for** každého hráče $i \in P$ **do**
- 6: Hráč i obdrží vektor ztrát $\ell_i^t = (\ell_i^t(a_i))_{a_i \in A_i}$, kde
$$\ell_i^t(a_i) \leftarrow \mathbb{E}_{a_{-i} \sim p_{-i}^t}[C_i(a_i; a_{-i})]$$
pro produktovou distribuci
$$p_{-i}^t = \prod_{j \neq i} p_j^t.$$
- 7: **end for**
- 8: **end for**
- 9: **return** $\{p^t : t \in \{1, \dots, T\}\}$

Tento algoritmus umožňuje hráčům v normální formě hry, aby si vybírali své strategie nezávisle tak, aby minimalizovali svůj regret. Každý hráč postupuje podle smíšené strategie a aktualizuje své rozhodnutí na základě aktuálních ztrát (loss), které obdrží od ostatních hráčů ve hře.

31.7 Moderní důkaz Minimaxové věty

Minimaxová věta je klíčovým výsledkem teorie her, který popisuje optimální strategie v *zero-sum* hrách (hry s nulovým součtem). Tyto hry jsou charakterizovány tím, že zisk jednoho hráče je přesnou ztrátou druhého. Moderní důkaz Minimaxové věty přistupuje k problému bez využití lineárního programování (LP).

Definice hry: Hru G lze reprezentovat jako:

$$G = (\{1, 2\}, A, C),$$

kde:

- $\{1, 2\}$ jsou hráči.
- $A_1 = \{a_1, \dots, a_m\}$ je množina akcí hráče 1.
- $A_2 = \{b_1, \dots, b_n\}$ je množina akcí hráče 2.

- Matice M velikosti $m \times n$ je definována jako $M_{i,j} = -C_1(a_i, b_j) = C_2(a_i, b_j)$, kde $M_{i,j} \in [-1, 1]$.

Smíšené strategie hráčů jsou popsány vektory pravděpodobností \mathbf{x} (pro hráče 1) a \mathbf{y} (pro hráče 2). Očekávané náklady hráče 2 jsou dány vztahem:

$$C_2(s) = \mathbf{x}^\top M \mathbf{y}.$$

Tvrzení Minimaxové věty:

$$\max_{\mathbf{x} \in S_1} \min_{\mathbf{y} \in S_2} \mathbf{x}^\top M \mathbf{y} = \min_{\mathbf{y} \in S_2} \max_{\mathbf{x} \in S_1} \mathbf{x}^\top M \mathbf{y}.$$

Toto tvrzení říká, že maximální minimalizovaný zisk hráče 1 je stejný jako minimální maximalizovaný zisk hráče 2.

Poznámka: Tento důkaz nevyužívá lineární programování, což je odlišný přístup oproti klasickým metodám.

31.8 Moderní důkaz Minimaxové věty I

Intuitivní vysvětlení: Moderní důkaz Minimaxové věty začíná ukázáním dvou nerovností, které dohromady prokazují tvrzení věty. První nerovnost $\max_{\mathbf{x}} \min_{\mathbf{y}} \mathbf{x}^\top M \mathbf{y} \leq \min_{\mathbf{y}} \max_{\mathbf{x}} \mathbf{x}^\top M \mathbf{y}$ vyplývá přímo z vlastnosti maxim a minim – jít jako první je vždy nevhodné. Druhá nerovnost $\max_{\mathbf{x}} \min_{\mathbf{y}} \mathbf{x}^\top M \mathbf{y} \geq \min_{\mathbf{y}} \max_{\mathbf{x}} \mathbf{x}^\top M \mathbf{y}$ se dokazuje pomocí dynamiky algoritmů s nízkým regretem (no-regret dynamics). Pomocí parametrů algoritmu se hráči přiblíží optimálním strategiím.

Formální zápis:

- Nejprve dokážeme nerovnost:

$$\max_{\mathbf{x}} \min_{\mathbf{y}} \mathbf{x}^\top M \mathbf{y} \leq \min_{\mathbf{y}} \max_{\mathbf{x}} \mathbf{x}^\top M \mathbf{y}.$$

Tato nerovnost plyne přímo z vlastnosti maxim a minim, protože jít jako první (maximalizace před minimalizací) je nevhodné.

- Nyní dokážeme opačnou nerovnost:

$$\max_{\mathbf{x}} \min_{\mathbf{y}} \mathbf{x}^\top M \mathbf{y} \geq \min_{\mathbf{y}} \max_{\mathbf{x}} \mathbf{x}^\top M \mathbf{y}.$$

- Zvolíme parametr $\varepsilon \in (0, 1]$ a necháme algoritmus No-regret dynamics běžet po dostatečný počet kroků T , aby oba hráči měli průměrný externí regret nejvýše ε .

- S algoritmem PW můžeme nastavit $T = \frac{4 \ln(\max\{m, n\})}{\varepsilon^2}$.

- Označíme strategie hráčů 1 a 2 v jednotlivých krocích jako $\mathbf{p}^1, \dots, \mathbf{p}^T$ a $\mathbf{q}^1, \dots, \mathbf{q}^T$.

- Průměrné strategie hráčů jsou:

$$\bar{\mathbf{x}} = \frac{1}{T} \sum_{t=1}^T \mathbf{p}^t, \quad \bar{\mathbf{y}} = \frac{1}{T} \sum_{t=1}^T \mathbf{q}^t.$$

- Vektory výplat (payoff vectors), které hráči získají, jsou $(M\mathbf{q}^t)$ a $(-(\mathbf{p}^t)^\top M)$.
- Průměrná očekávaná výplata hráče 1 je:

$$\nu = \frac{1}{T} \sum_{t=1}^T (\mathbf{p}^t)^\top M \mathbf{q}^t.$$

31.9 Moderní důkaz Minimaxové věty II

Intuitivní vysvětlení: Druhý krok moderního důkazu Minimaxové věty využívá externí regret hráčů k odvození nerovnosti. Použitím vlastností lineárních kombinací strategií a linearity očekávání dokážeme, že časově průměrné strategie se blíží minimaxové rovnováze. Při dostatečně dlouhém běhu algoritmu ($T \rightarrow \infty$) se dosažené výsledky přiblížují optimálním hodnotám.

Formální zápis:

- Pro $i = 1, \dots, m$ označme $\mathbf{e}_i = (0, \dots, 0, 1, 0, \dots, 0)$ jako směrodatný vektor pro čistou strategii a_i . Jelikož externí regret hráče 1 je nejvýše ε , platí:

$$\mathbf{e}_i^\top M \bar{\mathbf{y}} = \frac{1}{T} \sum_{t=1}^T \mathbf{e}_i^\top M \mathbf{q}^t \leq \frac{1}{T} \sum_{t=1}^T (\mathbf{p}^t)^\top M \mathbf{q}^t + \varepsilon = \nu + \varepsilon.$$

- Protože každá strategie $\mathbf{x} \in S_1$ je konvexní kombinací vektorů \mathbf{e}_i , linearita očekávání dává:

$$\mathbf{x}^\top M \bar{\mathbf{y}} \leq \nu + \varepsilon.$$

- Analogicky:

$$\bar{\mathbf{x}}^\top M \mathbf{y} \geq \nu - \varepsilon \quad \text{pro každé } \mathbf{y} \in S_2.$$

- Spojením těchto tvrzení získáváme:

$$\max_{\mathbf{x} \in S_1} \min_{\mathbf{y} \in S_2} \mathbf{x}^\top M \mathbf{y} \geq \min_{\mathbf{y} \in S_2} \bar{\mathbf{x}}^\top M \mathbf{y} \geq \nu - \varepsilon \geq \max_{\mathbf{x} \in S_1} \bar{\mathbf{x}}^\top M \bar{\mathbf{y}} - 2\varepsilon \geq \min_{\mathbf{y} \in S_2} \max_{\mathbf{x} \in S_1} \mathbf{x}^\top M \mathbf{y} - 2\varepsilon.$$

- Pro $T \rightarrow \infty$ se $\varepsilon \rightarrow 0$, a tedy dosáhneme požadované nerovnosti.

31.10 Hrubá korelovaná rovnováha

Intuitivní vysvětlení: Korelovaná rovnováha (Correlated Equilibrium) je koncept rovnováhy, kde hráči mohou sdílet společný signál (doporučení), který ovlivňuje jejich strategie. Podmínkou je, že žádný hráč nemůže získat vyšší výplatu změnou své strategie, pokud ostatní hráči následují doporučení. Hrubá korelovaná rovnováha (Coarse Correlated Equilibrium) je slabší koncept, kde hráči hodnotí své strategie bez ohledu na informace o společném signálu.

Formální definice: Rozložení pravděpodobnosti p na A je korelovanou rovnováhou (CE), pokud pro každého hráče $i \in P$ a všechny čisté strategie $a_i, a'_i \in A_i$ platí:

$$\sum_{a_{-i} \in A_{-i}} C_i(a_i; a_{-i}) p(a_i; a_{-i}) \leq \sum_{a_{-i} \in A_{-i}} C_i(a'_i; a_{-i}) p(a_i; a_{-i}),$$

kde a_{-i} jsou strategie ostatních hráčů.

Jinými slovy:

$$\mathbb{E}_{a \sim p}[C_i(a) | a_i] \leq \mathbb{E}_{a \sim p}[C_i(a'_i; a_{-i}) | a_i].$$

Hrubá korelovaná rovnováha je slabší koncept, kde podmínka je definována pouze v globálním měřítku:

$$\mathbb{E}_{a \sim p}[C_i(a)] \leq \mathbb{E}_{a \sim p}[C_i(a'_i; a_{-i})].$$

Rozdíl na příkladu: Představme si dvě firmy (hráče), které dostávají doporučení, jak nastavit své ceny (strategie). V případě korelované rovnováhy každá firma zvažuje doporučení i ve vztahu k doporučení druhé firmy (mají přístup k signálu). Hrubá korelovaná rovnováha ale umožňuje, aby hráči rozhodovali pouze na základě svých vlastních strategií, bez ohledu na informace o doporučení druhé firmy.

Vysvětlení pomocí příkladu: - *Korelovaná rovnováha:* Hráči mají přístup ke společnému signálu, například semaforu, který jim radí, jakou strategii zvolit (např. červená = vysoká cena, zelená = nízká cena). Hráči následují tento signál, protože žádný z nich nemá motivaci jednat jinak. - *Hrubá korelovaná rovnováha:* Hráči neberou v úvahu signál, ale stále hodnotí, zda jejich strategie přinese vyšší výplatu při daných pravděpodobnostech ostatních strategií.

Závěr: Hrubá korelovaná rovnováha je méně restriktivní, protože nezohledňuje závislost na signálu. No-regret dynamika může konvergovat do hrubé korelované rovnováhy díky slabším podmírkám optimality.

31.11 Hrubá korelovaná rovnováha (Coarse Correlated Equilibrium - CCE)

Definice: Pro hru v normální formě $G = (P, A, C)$ s n hráči je rozdělení pravděpodobnosti p na množině akcí A hrubou korelovanou rovnováhou (CCE)

právě tehdy, pokud pro každého hráče $i \in P$ a každou alternativní akci $a'_i \in A_i$ platí:

$$\sum_{a \in A} C_i(a)p(a) \leq \sum_{a \in A} C_i(a'_i; a_{-i})p(a),$$

kde a_{-i} označuje akce všech hráčů kromě i .

Ekvivalentně:

$$\mathbb{E}_{a \sim p}[C_i(a)] \leq \mathbb{E}_{a \sim p}[C_i(a'_i; a_{-i})].$$

Rozdíl mezi korelovanou (CE) a hrubou korelovanou rovnováhou (CCE): Hlavní rozdíl spočívá v tom, že CCE požaduje, aby následování doporučené akce a_i bylo nejlepší odpověďí v očekávání, ještě předtím, než hráč uvidí svou doporučenou akci. Naproti tomu CE zahrnuje dodatečnou podmínu, že hráč stále nebude mít motivaci změnit svou akci ani poté, co obdrží doporučení (signál).

Intuitivní vysvětlení: Hrubá korelovaná rovnováha je slabší koncept než korelovaná rovnováha, protože hráči hodnotí výplaty na základě očekávání všech akcí (globálně), nikoliv s ohledem na konkrétní signál, který jim byl doporučen. Jinými slovy, hráči se musí rozhodnout, zda budou následovat doporučené strategie, ještě než uvidí, co jim bylo doporučeno.

Příklad: Představme si semafor: - *CCE*: Semafor rozdává doporučení hráčům. Hráči hodnotí, zda je jejich doporučení výhodné v porovnání s jinými možnými strategiemi, ale rozhodují se ještě předtím, než vidí své konkrétní doporučení. - *CE*: Hráči znají doporučení a rozhodují se, zda jej následovat, nebo změnit strategii na základě konkrétního doporučení.

Shrnutí: CCE je více obecný koncept, který pokryvá situace, kdy hráči nevidí konkrétní doporučení, ale stále analyzují očekávané výplaty ze všech možných strategií. Díky tomu je jednodušší dosáhnout CCE pomocí algoritmů jako no-regret dynamika.

31.12 Hierarchie Nashových rovnováh

Shrnutí hierarchie Nashových rovnováh:

- **Čisté Nashovy rovnováhy (PNE):** - Tyto rovnováhy nemusí existovat ve všech hrách. - Jsou velmi specifické a často obtížné na dosažení. - Hráči volí čisté strategie, což znamená, že každá strategie má pravděpodobnost 1.
- **Smíšené Nashovy rovnováhy (MNE):** - Existují v každé konečné hře díky Nashovu teorému. - Hráči mohou hrát smíšené strategie (pravděpodobnostní rozdělení přes čisté strategie). - Výpočet těchto rovnováh je však náročnější.
- **Korelované rovnováhy (CE):** - Jsou snadnější na výpočet ve srovnání s MNE. - Hráči dostávají doporučení, jakou strategii hrát, a musí následovat doporučení, aby maximalizovali svůj užitek.

- **Hrubé korelované rovnováhy (CCE):** - Nejjednodušší na výpočet. - Hráči nemusí vidět konkrétní doporučení předem, ale hodnotí užitek z celé distribuce strategií. - No-regret dynamika v obecných hrách konverguje právě do CCE.

ε -**Hrubá korelovaná rovnováha (ε -CCE):** Pro $\varepsilon > 0$ je pravděpodobnostní rozdělení p na A označeno jako ε -CCE, pokud platí:

$$\mathbb{E}_{a \sim p}[C_i(a)] \leq \mathbb{E}_{a \sim p}[C_i(a'_i; a_{-i})] + \varepsilon.$$

Shrnutí: Tato hierarchie ukazuje, že každá další vrstva rovnováhy je obecnější a snadnější na výpočet, přičemž hrubé korelované rovnováhy jsou nejslabším a nejjednodušším konceptem, zatímco čisté Nashovy rovnováhy jsou nejsilnějším a často nejnáročnějším konceptem na výpočet i nalezení.

31.13 Konvergence k ε -CCE

Intuitivní vysvětlení

Věta tvrdí, že pokud hráči minimalizují své "regrety" (tj. rozdíl mezi tím, co mohli získat nejlepší strategií, a tím, co skutečně získali), mohou dosáhnout ε -Hrubého korelovaného ekvilibria (ε -CCE). To znamená, že strategie hráčů jsou dlouhodobě téměř optimální, s tolerancí ε .

Probíhá to takto: 1. Hráči iterativně zlepšují své strategie podle dynamiky minimalizace regredu. 2. Po dostatečně dlouhé době (T) se jejich kombinované strategie (průměr z iterací) ustálí na ε -CCE. 3. Průměrné rozdělení strategií p je vypočteno z distribucí strategií p^t v každé iteraci t .

Formální znění věty

Pro každou normální hru $G = (P, A, C)$, $\varepsilon > 0$, a $T = T(\varepsilon) \in \mathbb{N}$, platí, že pokud po T krocích dynamiky minimalizace regredu má každý hráč $i \in P$ časově průměrný regret nejvýše ε , pak p je ε -CCE, kde:

$$p = \frac{1}{T} \sum_{t=1}^T p^t.$$

Důkaz

Chceme dokázat:

$$\mathbb{E}_{a \sim p}[C_i(a)] \leq \mathbb{E}_{a \sim p}[C_i(a'_i, a_{-i})] + \varepsilon,$$

pro každého hráče $i \in P$ a každou alternativní strategii $a'_i \in A_i$.

Postup důkazu:

1. Podle definice p máme:

$$\mathbb{E}_{a \sim p}[C_i(a)] = \frac{1}{T} \sum_{t=1}^T \mathbb{E}_{a \sim p^t}[C_i(a)],$$

a

$$\mathbb{E}_{a \sim p}[C_i(a'_i, a_{-i})] = \frac{1}{T} \sum_{t=1}^T \mathbb{E}_{a \sim p^t}[C_i(a'_i, a_{-i})].$$

2. Pravá strana vyjadřuje časově průměrné očekávané náklady hráče i , když hraje a'_i místo své skutečné akce.
3. Jelikož každý hráč má regret nejvýše ε , platí:

$$\frac{1}{T} \sum_{t=1}^T \mathbb{E}_{a \sim p^t}[C_i(a)] \leq \frac{1}{T} \sum_{t=1}^T \mathbb{E}_{a \sim p^t}[C_i(a'_i, a_{-i})] + \varepsilon.$$

4. To přímo potvrzuje podmínku ε -CCE:

$$\mathbb{E}_{a \sim p}[C_i(a)] \leq \mathbb{E}_{a \sim p}[C_i(a'_i, a_{-i})] + \varepsilon.$$

Tím je věta dokázána. \square

Další koncepty regredu: Modifikace strategie

Intuitivní vysvětlení

Pro detailnější analýzu regredu uvažujeme "vnitřní nastavení" (internal setting), kde porovnáváme chování hráče vůči upraveným strategiím. Uvažujeme: 1. **Pravidlo modifikace:** Funkce $F : X \rightarrow X$, která říká, jak měnit akce hráče. 2. **Upravené strategie:** Místo pravděpodobnostního rozdělení strategií p^t hráče použijeme f^t , kde každá akce je změněna podle pravidla F .

Například: - Pokud hráč často volil akci i , F může určit, že tyto akce nahradí j . Výpočet regredu ukazuje, zda taková modifikace mohla zlepšit výsledek.

Formální znění

- **Pravidlo modifikace:** Funkce $F : X \rightarrow X$.
- **Modifikované rozdělení:** Sekvenci $(p^t)_{t=1}^T$ nahradíme sekvencí $(f^t)_{t=1}^T$, kde:

$$f_i^t = \sum_{j:F(j)=i} p_j^t.$$

Upravený hráč volí $F(i)$, kdykoli původní hráč volí i .

- **Kumulativní ztráta:** Celková ztráta hráče A modifikovaného pravidlem F je:

$$L_{A,F}^T = \sum_{t=1}^T \sum_{i=1}^N f_i^t \ell_i^t.$$

- **Regret vůči modifikaci:** Pro množinu modifikačních pravidel \mathcal{F} porovnáváme hráče s jeho modifikacemi, čímž získáváme různé typy regretru.
-

Jiné pojmy regretru

Intuitivní vysvětlení

Při analýze regretru se můžeme zaměřit na různé situace, které porovnávají chování hráče vůči různým "modifikacím" jeho strategií: - **Externí regret:** Měří, o kolik lépe by hráč dopadl, kdyby celou dobu hrál jednu pevnou akci. - **Interní regret:** Zjišťuje, zda by změna jedné konkrétní akce na jinou v jeho strategii vedla k lepšímu výsledku. - **Swap regret:** Zobecňuje interní regret tím, že umožňuje libovolné pravidlo, které přemapovává akce hráče na jiné akce (např. podle kontextu nebo historie hry).

Každý typ regretru se zaměřuje na jiné otázky o efektivitě strategie hráče a dává nám odlišné informace o tom, jak by mohl zlepšit své rozhodování.

Formální znění

Externí regret:

$$R_{A,\mathcal{F}^{\text{ex}}}^T = \max_{F \in \mathcal{F}^{\text{ex}}} \{L_A^T - L_{A,F}^T\} = \max_{j \in X} \left\{ \sum_{t=1}^T \left(\sum_{i \in X} p_i^t \ell_i^t - \ell_j^t \right) \right\}.$$

- Zde $\mathcal{F}^{\text{ex}} = \{F_i : i \in X\}$, kde F_i vždy volí pevnou akci i .

Interní regret:

$$R_{A,\mathcal{F}^{\text{in}}}^T = \max_{F \in \mathcal{F}^{\text{in}}} \{L_A^T - L_{A,F}^T\} = \max_{i,j \in X} \left\{ \sum_{t=1}^T p_i^t (\ell_i^t - \ell_j^t) \right\}.$$

- Zde $\mathcal{F}^{\text{in}} = \{F_{i,j} : (i,j) \in X \times X, i \neq j\}$, kde $F_{i,j}(i) = j$ a $F_{i,j}(i') = i'$ pro $i' \neq i$.

Swap regret:

$$R_{A,\mathcal{F}^{\text{sw}}}^T = \max_{F \in \mathcal{F}^{\text{sw}}} \{L_A^T - L_{A,F}^T\} = \sum_{i=1}^N \max_{j \in X} \left\{ \sum_{t=1}^T p_i^t (\ell_i^t - \ell_j^t) \right\}.$$

- Zde \mathcal{F}^{sw} obsahuje všechna možná pravidla modifikace akcí.

Hierarchie regretů:

$$R_{A,\mathcal{F}^{\text{ex}}}^T \leq R_{A,\mathcal{F}^{\text{in}}}^T \leq R_{A,\mathcal{F}^{\text{sw}}}^T.$$

31.14 Redukce z externí lítosti na swap lítost

Intuitivní vysvětlení: Externí lítost (*external regret*) nám říká, jak dobře jsme hráli ve srovnání s nejlepší fixní akcí (nebo strategií), kterou jsme mohli zvolit od začátku hry. Swap lítost (*swap regret*) jde o krok dále a zohledňuje možnost, že bychom mohli retrospektivně změnit akce dle určité funkce (např. pokud pokaždé, když zvolím akci i , přepnu na akci j).

Cílem této sekce je ukázat, že pokud máme algoritmus s malou externí lítostí, můžeme z něj vytvořit algoritmus, který má malou swap lítost. Toho je dosaženo pomocí tzv. *black-box* redukce, která transformuje algoritmus zaměřený na externí lítost do algoritmu s kontrolou swap lítosti.

Formální znění a věta:

Věta 2.55: Pro každý algoritmus A s externí lítostí nejvýše R existuje algoritmus $M = M(A)$ takový, že pro každou funkci $F : X \rightarrow X$ a $T \in \mathbb{N}$ platí:

$$L_M^T \leq L_{M,F}^T + NR.$$

To znamená, že swap lítost algoritmu M je nejvýše NR , kde N je počet možných akcí.

31.15 Důkaz Redukce z externí lítosti na swap lítost

Intuitivní vysvětlení: Cílem důkazu je ukázat, že algoritmus s nízkou externí lítostí lze pomocí redukce přeměnit na algoritmus s nízkou swap lítostí. Myšlenka spočívá v tom, že použijeme kopie daného algoritmu A (s nízkou externí lítostí) a vytvoříme takzvaný "master" algoritmus M , který kombinuje výstupy jednotlivých kopií. Tento master algoritmus pracuje s přechodovou maticí Markovova řetězce, čímž umožňuje efektivní výběr akcí na základě pravděpodobnostního rozdělení. V průběhu důkazu se ukazuje, že celková ztráta všech algoritmů A_i odpovídá ztrátě algoritmu M , a díky vlastnostem externí lítosti algoritmu A se swap lítost algoritmu M ukáže být omezená.

Formální důkaz:

Důkaz - Část I:

- Předpokládejme, že A_1, \dots, A_N jsou kopie algoritmu A . V každém kroku t algoritmus A_i vrací pravděpodobnostní rozdělení $q_i^t = (q_{i,1}^t, \dots, q_{i,N}^t)$, kde $q_{i,j}^t$ udává pravděpodobnost, kterou A_i přiřadí akci $j \in X$.
- Sestavíme master algoritmus M kombinací těchto kopií A .

- Definujeme pravděpodobnostní rozdělení $p^t = (p_1^t, \dots, p_N^t)$ následovně:

$$p_j^t = \sum_{i=1}^N p_i^t q_{i,j}^t,$$

kde Q^t je $N \times N$ matice s prvky $Q_{i,j}^t = q_{i,j}^t$.

- Matematicky tedy:

$$(p^t)^\top = (p^t)^\top Q^t.$$

- Tato volba p^t zaručuje, že výběr akcí lze interpretovat dvěma způsoby:

- Akce $j \in X$ je přímo vybrána s pravděpodobností p_j^t .
- Nebo nejprve vybereme algoritmus A_i s pravděpodobností p_i^t a poté akci j podle $q_{i,j}^t$.

Důkaz - Část II:

- Ukážeme, že celková ztráta všech algoritmů A_i v kroku t odpovídá ztrátě algoritmu M , tj. $p^t \cdot \ell^t$.
- Po obdržení vektoru ztrát ℓ^t přiřadíme každému A_i vektor ztrát $p_i^t \ell^t$.
- Algoritmus A_i pak zaznamená ztrátu $p_i^t(\ell^t \cdot q_i^t) = p_i^t(q_i^t \cdot \ell^t)$.
- Protože A_i má externí lítost R , platí pro každé $j \in X$:

$$\sum_{t=1}^T p_i^t(q_i^t \cdot \ell^t) \leq \sum_{t=1}^T p_i^t \ell_j^t + R.$$

- Sečteme ztráty všech algoritmů A_i přes $i \in X$, což dává:

$$\sum_{i=1}^N \sum_{t=1}^T p_i^t(q_i^t \cdot \ell^t) = \sum_{t=1}^T (p^t)^\top Q^t \ell^t.$$

- Výběrem p^t platí $(p^t)^\top = (p^t)^\top Q^t$, což ukazuje, že jsme dosáhli kýženého výsledku.

Důkaz - Část III:

- Sečteme přes všechny akce $i \in X$:

$$\sum_{t=1}^T p_i^t(q_i^t \cdot \ell^t) \leq \sum_{t=1}^T p_j^t \ell_j^t + R.$$

- Pravá strana této nerovnosti platí pro každou funkci $F : X \rightarrow X$, takže dostáváme:

$$L_M^T \leq \sum_{i=1}^N \sum_{t=1}^T p_i^t \ell_{F(i)}^t + NR = L_{M,F}^T + NR.$$

- Použitím algoritmu PW jako A získáme algoritmus s lítostí typu swap, která konverguje k 0, když $T \rightarrow \infty$.

31.16 Redukce externí lítostí na swap lítost

Cílem této sekce je intuitivně a jednoduše vysvětlit důkaz, že algoritmus s nízkou externí lítostí lze převést na algoritmus s nízkou swap lítostí. Klíčovou myšlenkou je použít kopie algoritmu s externí lítostí a zkombinovat jejich výstupy tak, aby výsledný algoritmus dosahoval nízké swap lítosti.

31.17 Myšlenka důkazu

Externí lítost (external regret) měří, jak dobře by si hráč vedl, kdyby celou dobu hrál jednu fixní akci. Naproti tomu swap lítost (swap regret) je složitější a zohledňuje možnost retrospektivně měnit jednotlivé akce podle nějakého pravidla (např. "pokud jsem hrál akci i , měl jsem místo toho hrát j ").).

- Hlavní idea důkazu je:
- Použít *kopie algoritmu* s nízkou externí lítostí.
- Kombinovat jejich výstupy pomocí tzv. *master algoritmu*, který rozhoduje o výběru akce.
- Tím zajistíme, že výsledný algoritmus dosáhne nízké swap lítosti.

31.18 Kroky důkazu

Důkaz lze rozdělit na několik intuitivních kroků:

Krok 1: Kopie algoritmů. Nechť máme N kopií algoritmu A s nízkou externí lítostí. Každý algoritmus A_i (pro $i \in \{1, \dots, N\}$) je zodpovědný za výběr akce i .

Každý algoritmus A_i v čase t vrací pravděpodobnostní rozdělení q_i^t , které říká, jak pravděpodobně má být zvolena akce j , pokud byl algoritmus A_i aktivní.

Krok 2: Master algoritmus. Zavedeme *master algoritmus*, který rozhoduje, který z algoritmů A_i bude v každém čase aktivní. Pravděpodobnost, že master algoritmus zvolí algoritmus A_i , označíme p_i^t .

Master algoritmus tedy kombinuje výstupy všech A_i pomocí pravděpodobnostního vektoru $p^t = (p_1^t, \dots, p_N^t)$.

Krok 3: Kombinované rozdělení akcí. Master algoritmus definuje výsledné pravděpodobnostní rozdělení akcí p_j^t v čase t takto:

$$p_j^t = \sum_{i=1}^N p_i^t q_{i,j}^t,$$

kde $q_{i,j}^t$ je pravděpodobnost, že algoritmus A_i zvolí akci j .

Krok 4: Výpočet ztrát. Každý algoritmus A_i zaznamenává svou ztrátu $\ell^t \cdot q_i^t$ v čase t , kde ℓ^t je vektor ztrát pro všechny akce.

Celková ztráta všech algoritmů A_i je kombinována master algoritmem podle pravděpodobností p_i^t .

Krok 5: Redukce z externí na swap lítost. Vlastnosti algoritmů A_i s nízkou externí lítostí zaručují, že celková ztráta výsledného master algoritmu je omezená. Pro každé pravidlo přemapování akcí (swap pravidlo) lze ukázat, že:

$$L_T^M \leq L_T^{M,F} + NR,$$

kde L_T^M je ztráta master algoritmu, $L_T^{M,F}$ je ztráta při použití swap pravidla F a NR je člen penalizace závislý na počtu akcí N a externí lítosti R .

Použitím algoritmů A_i s externí lítostí konvergující k 0 (pro $T \rightarrow \infty$) zajistíme, že swap lítost výsledného algoritmu také konverguje k 0.

31.19 Závěr

Důkaz ukazuje, že algoritmus s nízkou externí lítostí lze transformovat na algoritmus s nízkou swap lítostí použitím kopí algoritmu a master algoritmu, který kombinuje jejich výstupy. Tento postup zajišťuje efektivní kontrolu nad swap lítostí i v situacích, kdy jsou přemapování akcí složitější.

Dynamika No-swap-regret

Intuitivní vysvětlení

Dynamika No-swap-regret je algoritmus, který umožňuje hráčům v iterativní hře minimalizovat svůj swap regret. Swap regret měří, zda by hráč mohl zlepšit svůj výsledek tím, že by změnil své tahy na základě určitého pravidla (např. vždy vyměnil akci i za j podle historie hry).

Cílem algoritmu je vést hráče k distribuci strategií, která je v korelovaném ekvilibriu (CE), což znamená, že žádný hráč nemá motivaci odchýlit se od navrhované strategie, a to ani po zvážení možných "výměn" svých tahů.

Algoritmus funguje takto: 1. Každý hráč iterativně vybírá smíšenou strategii, která minimalizuje jeho swap regret. 2. Hráči dostávají zpětnou vazbu o svých ztrátách (loss vector) na základě společné distribuce strategií všech hráčů. 3. Proces se opakuje po T iteracích, přičemž každá iterace upravuje strategie směrem k minimalizaci swap regredu. 4. Nakonec se distribuce strategií ustálí a konverguje ke korelovanému ekvilibriu.

Algoritmus 0.15: Dynamika No-swap-regret

31.20 Konvergence k korelované rovnováze (CE)

Intuitivní vysvětlení: Tato věta ukazuje, že pokud všichni hráči používají dynamiku s malou swap lítostí (no-swap-regret), pak po dostatečném počtu kroků dojde k tomu, že časově průměrné strategie hráčů konvergují k ε -korelované rovnováze (ε -CE). To znamená, že při průměrné ztrátě nedává hráči smysl "měnit" svoji strategii dle libovolné funkce F , protože by si tím nezlepšil výsledek o více než ε . Důkaz je založen na vlastnosti swap lítosti, která zaručuje, že časově

Algorithm 6 Dynamika No-swap-regret (No-SWAP-REGRET DYNAMICS)

Vstup: Normální forma hry $G = (P, A, C)$ s n hráči, počet iterací $T \in \mathbb{N}$, tolerance swap regretem $\varepsilon > 0$.

Výstup: Pravděpodobnostní distribuce p_i^t pro každého hráče $i \in P$ a $t \in \{1, \dots, T\}$.

- 1: **for** $t = 1, \dots, T$ **do**
 - 2: **for** každého hráče $i \in P$ **do**
 - 3: Každý hráč i nezávisle vybírá smíšenou strategii p_i^t pomocí algoritmu s průměrným swap regretem nejvýše ε , kde akce odpovídají čistým strategiím.
 - 4: Hráč i obdrží vektor ztrát $\ell_i^t = (\ell_i^t(a_i))_{a_i \in A_i}$, kde:
$$\ell_i^t(a_i) \leftarrow \mathbb{E}_{a_{-i} \sim p_{-i}^t} [C_i(a_i; a_{-i})],$$

s distribucí $p_{-i}^t = \prod_{j \neq i} p_j^t$.
 - 5: **end for**
 - 6: **end for**
 - 7: **Výstup:** Sekvence distribucí strategií $\{p^t : t \in \{1, \dots, T\}\}$.
-

průměrné očekávané náklady při hraní aktuální strategie nejsou výrazně horší než při aplikování jiné strategie odvozené funkcí F .

Věta 2.57: Pro každou hru v normální formě $G = (P, A, C)$, kde $\varepsilon > 0$ a $T = T(\varepsilon) \in \mathbb{N}$, platí: Pokud po T krocích no-swap-regret dynamiky má každý hráč $i \in P$ časově průměrnou očekávanou lítost maximálně ε , pak p je ε -CE, kde

$$p^t = \prod_{i=1}^n p_i^t \quad \text{a} \quad p = \frac{1}{T} \sum_{t=1}^T p^t.$$

Důkaz:

- Chceme dokázat, že:

$$\mathbb{E}_{a \sim p} [C_i(a)] \leq \mathbb{E}_{a \sim p} [C_i(F(a_i); a_{-i})] + \varepsilon,$$

kde $F : X \rightarrow X$ je libovolná funkce.

- Podle definice p platí, že pro každého hráče $i \in P$ a každou funkci $F : X \rightarrow X$:

$$\mathbb{E}_{a \sim p} [C_i(a)] = \frac{1}{T} \sum_{t=1}^T \mathbb{E}_{a \sim p^t} [C_i(a)],$$

$$\mathbb{E}_{a \sim p} [C_i(F(a_i); a_{-i})] = \frac{1}{T} \sum_{t=1}^T \mathbb{E}_{a \sim p^t} [C_i(F(a_i); a_{-i})].$$

- Pravé strany představují časově průměrné očekávané náklady hráče i , když hraje podle aktuální strategie nebo podle funkce F .

- Protože každý hráč má swap lítost nejvýše ε , získáme:

$$\frac{1}{T} \sum_{t=1}^T \mathbb{E}_{a \sim p^t}[C_i(a)] \leq \frac{1}{T} \sum_{t=1}^T \mathbb{E}_{a \sim p^t}[C_i(F(a_i); a_{-i})] + \varepsilon.$$

- Tím ověřujeme podmínu ε -CE pro $p = \frac{1}{T} \sum_{t=1}^T p^t$.

32 Hry v rozšířené formě

32.1 Úvod do her v rozšířené formě

Hry v normální formě popisují situace, kde hráci jednají současně. Výsledkem je statický popis hry, který zobrazuje všechny možné strategie a jejich výplaty prostřednictvím tabulek.

V této části představujeme jinou reprezentaci her, která poskytuje *dynamický* popis her, kde hráci jednají sekvenčně. Namísto tabulek se pro popis těchto her používají *stromy*.

Příklad: U některých těchto her ukážeme, jak vypočítat Nashovo ekvilibrium.

32.2 Šach jako příklad normální formy

Šach jako hra v normální formě: Každá akce hráče $i \in \{\text{bílý, černý}\}$ odpovídá seznamu všech možných situací, které mohou nastat na šachovnici, společně s tahy, které by hráč i učinil v daných situacích. Celou hru šachu tak můžeme simulovat v jednom tahu.

32.3 Šach jako příklad rozšířené formy

Šach jako hra v rozšířené formě:

- Kořen stromu odpovídá počáteční pozici na šachovnici.
- Každý rozhodovací uzel reprezentuje nějakou pozici na šachovnici.
- Odchozí hrany uzlů odpovídají možným tahům, které lze z dané pozice zahrát.

32.4 Základní definice

Extenzivní hra je reprezentována orientovaným stromem, kde uzly představují stavy hry. Tento strom zaznamenává celou historii hry.

- Hra začíná v kořeni stromu a končí v listu, kde každý hráč získává výplatu. Hrana stromu odpovídá přechodu hráče z jednoho stavu hry do jiného stavu.

- Každý uzel, který není listem, se nazývá **rozhodovací uzel**.
- Možné tahy hráče ve stavu odpovídají hranám, které vycházejí z odpovídajícího rozhodovacího uzlu.
- **Perfektně informovaná hra** je hra, ve které všichni hráči znají aktuální uzel, ve kterém se nachází, a tedy celou historii tahu, která je do tohoto uzlu dovedla.
 - Příklad: Šachy.
- **Hra s neúplnou informací** je hra, kde hráči nemají úplné informace o aktuálním stavu hry.
 - Příklad: Poker.
 - Rozhodovací uzly v takové hře rozdělujeme do **informačních množin**, kde všechny uzly v množině patří stejnemu hráči a mají stejné možnosti tahů.
 - Pro hráče i označme H_i množinu jeho informačních množin a pro $h \in H_i$ označme C_h množinu tahů, které jsou možné ve stavu h .

32.5 Strategie v extenzivních hrách

- **Čistá strategie** pro hráče i je kompletní specifikace deterministického tahu, který má hráč provést v každé informační množině patřící hráci i .
 - Formálně: Čistá strategie hráče i je vektor $(c_h)_{h \in H_i}$ z kartézského součinu $\prod_{h \in H_i} C_h$.
 - Pomocí čistých strategií můžeme transformovat extenzivní hru G na normální formu hry G' , kde tabulkujeme všechny čisté strategie hráčů a zaznamenáváme očekávané výplaty.
- **Smíšené strategie** extenzivní hry G odpovídají smíšeným strategiím hry G' .
- Stejným způsobem definujeme i množinu **Nashových ekvilibrií** hry G .
- **Chování strategie** hráče i je pravděpodobnostní rozdělení na C_h pro každé $h \in H_i$.
 - Jde o strategii, ve které hráč provádí tahy nezávisle v každé své informační množině.
 - Chování strategie je vektor pravděpodobnostních rozdělení na množině C_h , zatímco smíšená strategie je pravděpodobnostní rozdělení na vektorech.
 - Na rozdíl od smíšené strategie hráč může v různých setkáních h provádět odlišné tahy.

32.5.1 Úvod do her v extenzivní formě (opakování)

Hry v extenzivní formě jsou reprezentovány pomocí stromů, kde uzly odpovídají stavům hry a hrany jednotlivým možným tahům hráčů. Tento typ her umožňuje detailnější pohled na posloupnost tahů ve hře.

Intuice: Tento typ modelování je obzvlášť užitečný v hrách s více kroky, jako je například šachová partie, kde každý krok hráče ovlivňuje následující rozhodnutí.

32.5.2 Základní definice

- **Extenzivní hra:** Reprezentována orientovaným stromem, jehož uzly odpovídají stavům hry. Hra začíná v kořeni stromu a končí v listu, kde hráči získají výplaty.
- **Tah hráče:** Odpovídá hranám vycházejícím z rozhodovacího uzlu.
- **Informační množiny:** Rozhodovací uzly seskupené podle znalostí hráče, který je na tahu. Uzly v jedné množině sdílejí stejné dostupné tahy.
- **Dokonalá a nedokonalá informace:** Ve hrách s dokonalou informací zná hráč přesný stav hry. Naopak ve hrách s nedokonalou informací má pouze částečnou znalost stavu.

Intuice: Extenzivní hry zobrazují rozhodovací procesy hráčů v průběhu celé hry a umožňují modelovat různé úrovně znalostí hráčů o aktuálním stavu hry.

32.5.3 Příklad: Věžovo dilema

- Věžovo dilema lze znázornit v extenzivní formě jako strom (část (a)) i v normální formě pomocí výplatní matici (část (b)).
- Stromová reprezentace ukazuje posloupnost rozhodnutí hráčů 1 a 2, přičemž každé rozhodnutí vede k určitému výsledku (výplatám).
- Každou hru v normální formě lze reprezentovat jako hru s nedokonalou informací v extenzivní formě.

Intuice: Extenzivní forma věžova dilematu umožňuje lépe pochopit dynamiku rozhodování mezi hráči a dopady jejich tahů na celkové výplaty.

32.5.4 Strategie v extenzivních hrách

- **Čistá strategie:** Určuje konkrétní akci hráče v každé jeho informační množině. Formálně je to vektor $(c_h)_{h \in H_i}$, kde H_i jsou informační množiny hráče i .
- **Smíšená strategie:** Pravděpodobnostní kombinace čistých strategií. Smíšené strategie v extenzivní hře G odpovídají smíšeným strategiím v normální formě G' .

- **Behaviorální strategie:** Pravděpodobnostní rozdělení na tahy v každé informační množině $h \in H_i$. Rozdíl oproti smíšeným strategiím spočívá v tom, že rozhodnutí v různých množinách jsou na sobě nezávislá.
- Lze definovat **Nashovy ekvilibria** pro extenzivní hry podobně jako pro normální formu.

Intuice: Různé typy strategií umožňují analyzovat rozhodování hráčů s různou mírou neurčitosti, a behaviorální strategie jsou často výhodné v hrách s nedokonalou informací.

32.5.5 Příklad: Behaviorální strategie

- Hra s dokonalou informací může být reprezentována v extenzivní formě jako strom (část (a)) a v normální formě jako tabulka výplat (část (b)).
- Strategie hráče 1, kdy volí A s pravděpodobností $\frac{1}{2}$ a G s pravděpodobností $\frac{1}{3}$, je **behaviorální strategie**.
- Smíšená strategie $\frac{3}{5}(A, G) + \frac{2}{5}(B, H)$ však **není behaviorální strategie**, protože rozhodnutí hráče na různých uzlech nejsou nezávislá.

Intuice: Behaviorální strategie poskytují flexibilitu hráčům, aby mohli měnit svá rozhodnutí v závislosti na informační množině, na rozdíl od smíšených strategií, které kombinují celé sekvence tahů.

32.5.6 Hry s dokonalou pamětí (perfect recall)

- Výrazová síla behaviorálních strategií a smíšených strategií je obecně neporovnatelná.
- Existuje však třída her, kde se tyto definice shodují – jedná se o hry s **dokonalou pamětí**.
- Hráč i má dokonalou paměť, pokud:
 - Pro každou informační množinu $h \in H_i$ a libovolné uzly $t, t' \in h$ platí, že posloupnost tahů $\sigma_i(t) = \sigma_i(t')$.
- Hry s dokonalou pamětí zaručují, že žádný hráč nezapomene, co se stalo v předchozích tazích.
- Každáhra s dokonalou informací je hrou s dokonalou pamětí.

Intuice: Dokonalá paměť zajišťuje, že hráči mohou využít plnou znalost historie hry, což umožňuje přesnější analýzu strategií a výsledků.

32.5.7 Kuhnův teorém

- V hrách s dokonalou pamětí (*perfect recall*) jsou smíšené strategie a behaviorální strategie ekvivalentní.
- **Kuhnův teorém:** Ve hře s dokonalou pamětí může být každá smíšená strategie nahrazena ekvivalentní behaviorální strategií a naopak.

Intuice: Tento teorém ukazuje, že ve hrách s dokonalou pamětí můžeme pracovat s libovolným typem strategií bez ztráty obecnosti, což zjednoduší analýzu.

32.5.8 Hledání Nashových ekvilibrií v extenzivních hrách

- Každá extenzivní hra G může být převedena na ekvivalentní hru v normální formě G' .
- Nashova ekvilibria (*NE*) hry G lze nalézt použitím algoritmu Lemke–Howson na G' .
- Tento přístup je však **neefektivní**, protože počet tahů v G' roste exponenciálně s velikostí G , což může vést k dvojitě exponenciální složitosti.
- Abychom se tomuto problému vyhnuli, budeme pracovat přímo s G pomocí tzv. **sekvenční formy**.
- Dále budeme uvažovat pouze hry s dokonalou pamětí, kde díky Kuhnovo teorému nemusíme rozlišovat mezi smíšenými a behaviorálními strategiemi.

Intuice: Přímá práce s extenzivní hrou v sekvenční formě umožňuje efektivnější analýzu Nashových ekvilibrií, aniž bychom museli přecházet na normální formu.

32.5.9 Sekvenční forma

- Sekvenční forma hry s nedokonalou informací G je čtyřice (P, S, u, \mathcal{C}) , kde:
 - P je množina n hráčů,
 - $S = (S_1, \dots, S_n)$, kde S_i je množina sekvencí tahů hráče i ,
 - $u = (u_1, \dots, u_n)$, kde $u_i : S \rightarrow \mathbb{R}$ je výplatní funkce hráče i ,
 - $\mathcal{C} = (C_1, \dots, C_n)$, což je množina lineárních omezení na realizované pravděpodobnosti hráče i .
- Množinu sekvencí S_i lze definovat podrobněji:
 - Každá sekvence $\sigma \in S_i$ je buď prázdná sekvence \emptyset , nebo je jednoznačně určena posledním tahem c v informační množině h , tedy $\sigma = \sigma_h c$.

- Platí $S_i = \{\emptyset\} \cup \{\sigma_h c : h \in H_i, c \in C_h\}$.
- Počet sekvencí v S_i je lineární vůči velikosti stromu hry G .

Intuice: Sekvenční forma poskytuje efektivnější reprezentaci strategií v hrách s nedokonalou informací, přičemž využívá lineární omezení pro realizaci pravděpodobnosti.

32.5.10 Detailní popis výplatní funkce

- Výplatní funkce u pro hráče i a sekvenční formu $\sigma = (\sigma_1, \dots, \sigma_n) \in S$:
 - $u_i(\sigma) = u_i(\ell)$, kde ℓ je list stromu dosažený, pokud každý hráč j hrál svou sekvenční formu σ_j .
 - Jinak $u_i(\sigma) = 0$.
- Výplaty lze reprezentovat pomocí matic, kde prvky odpovídají dvojicím sekvencí z S . Tyto matice jsou řídké, protože většina prvků je 0.
- Pro dva hráče lze výplaty reprezentovat pomocí matic A a B .

Intuice: Výplatní funkce spojuje konkrétní sekvenční formy hráčů s výslednými výplatami a využívá maticovou reprezentaci, což zjednoduší výpočty v hrách s více hráči.

32.5.11 Příklad: Výplatní matice v sekvenční formě

- Příklad hry s nedokonalou informací znázorněný v extenzivní formě (část (a)) a její výplatní matice v sekvenční formě (část (b)).
- Výplatní matice A a B :
 - Matice A odpovídá výplatám hráče 1 pro různé sekvenční formy tahů.
 - Matice B odpovídá výplatám hráče 2.
 - Řádky a sloupce reprezentují možné sekvenční formy tahů pro oba hráče.

Intuice: Výplatní matice v sekvenční formě umožňují efektivní výpočty a analýzu strategie pomocí strukturované reprezentace výplat pro různé kombinace tahů.

32.5.12 Dokončení definice sekvenční formy

- Pro přesný popis hry G je nutné definovat lineární omezení \mathcal{C} podrobněji.
- **Realizační plán** behaviorální strategie β_i hráče i je zobrazení $x : S_i \rightarrow [0, 1]$, kde:
 - $x(\sigma_i) = \prod_{c \in \sigma_i} \beta_i(c)$, což je pravděpodobnost, že daná sekvenční forma vznikne pod strategií β_i .

- Realizační plán lze popsat pomocí lineárních rovnic:
 - * $x(\emptyset) = 1$, což značí, že prázdná sekvence má pravděpodobnost 1.
 - * $\sum_{c \in C_h} x(\sigma_h c) = x(\sigma_h)$ pro každou informační množinu $h \in H_i$.
- C_i je množina omezení pro hráče i .

Intuice: Realizační plán umožňuje modelovat pravděpodobnosti jednotlivých sekvencí tahů hráčů pomocí behaviorálních strategií a lineárních omezení, což zjednodušuje výpočty v sekvenční formě.

32.5.13 Použití sekvenční formy

- Uvažujme extenzivní hru G se dvěma hráči.
- Realizační plány hráčů jsou reprezentovány vektory:
 - $x = (x_\sigma)_{\sigma \in S_1} \in \mathbb{R}^{|S_1|}$ pro hráče 1,
 - $y = (y_\tau)_{\tau \in S_2} \in \mathbb{R}^{|S_2|}$ pro hráče 2.
- Lineární omezení z \mathcal{C} lze zapsat jako:

$$Ex = e, \quad x \geq 0, \quad Fy = f, \quad y \geq 0,$$

kde:

- Matice E a F mají $1 + |H_1|$ a $1 + |H_2|$ řádků,
- První řádky odpovídají $x(\emptyset) = 1$ pro E a $y(\emptyset) = 1$ pro F ,
- Další řádky reprezentují podmínky $-x(\sigma_h) + \sum_{c \in C_h} x(\sigma_h c) = 0$ a obdobně pro y .

Intuice: Použití sekvenční formy umožňuje efektivní reprezentaci a řešení hry pomocí lineárních omezení na realizační plány hráčů.

32.5.14 Příklad: Omezení v sekvenční formě

- Příklad hry s nedokonalou informací v extenzivní formě (část (a)) a lineárních omezení v její sekvenční formě (část (b)).
- Omezení:

$$Ex = e, \quad Fy = f,$$

kde:

- Matice E a F definují omezení na realizační plány hráčů,
- Vektor e odpovídá inicializaci $x(\emptyset) = 1$,
- Hodnoty $\beta_i(h, c)$ lze obnovit z realizačního plánu jako:

$$\beta_i(h, c) = \frac{x(\sigma_h c)}{x(\sigma_h)} \quad \text{pro každé } h \in H_i \text{ a } c \in C_h.$$

Intuice: Tento příklad ukazuje, jak se omezení v sekvenční formě odvozují z extenzivní hry a jak lze z realizačního plánu obnovit původní behaviorální strategii.

32.5.15 Použití sekvenční formy pro nalezení nejlepší odpovědi

- Ukážeme, jak nalézt nejlepší odpovědi (*best responses*) pomocí sekvenční formy.
- Pro pevný realizační plán y hráče 2 je nejlepší odpověď hráče 1 realizační plán x , který maximalizuje očekávanou výplatu.
- Tento problém lze formulovat jako následující lineární program P :

$$\max x^\top A y \quad \text{za podmínek} \quad Ex = e, \quad x \geq 0.$$

- Dvojduální program D pro P využívá volné proměnné u a má tvar:

$$\min e^\top u \quad \text{za podmínek} \quad E^\top u \geq Ay.$$

- Analogické lineární programy lze použít pro výpočet nejlepší odpovědi hráče 2.

Intuice: Nejlepší odpovědi hráčů lze nalézt řešením optimalizačních problémů, které berou v úvahu očekávané výplaty a omezení realizačních plánů.

32.5.16 Použití sekvenční formy pro nalezení Nashova ekvilibria v nulových hrách

- Uvažujme hru G , která je nulová, tj. $A = -B$.
- Pomocí dualitního teorému lze formulovat lineární program (LP) pro nalezení Nashova ekvilibria v G .
- Hráč 2 minimalizuje $x^\top Ay$, zatímco hráč 1 maximalizuje stejný výraz $x^\top Ay$ podle duality.
- **Teorém 2.65:** Nashova ekvilibria dvouhráčové nulové extenzivní hry s dokonalou pamětí jsou řešením LP:

$$\min_{u,y} e^\top u \quad \text{za podmínek} \quad Fy = f, \quad E^\top u - Ay \geq 0, \quad y \geq 0.$$

- Dvojduální program používá proměnné x a v a má tvar:

$$\max_{v,x} f^\top v \quad \text{za podmínek} \quad Ex = e, \quad F^\top v - A^\top x \leq 0, \quad x \geq 0.$$

- Počet nenulových prvků v maticích E, F, A, B je lineární vzhledem k velikosti stromu hry, takže LP lze řešit v polynomiálním čase.

Intuice: Nulové hry umožňují efektivní nalezení Nashova ekvilibria pomocí optimalizačních metod díky vlastnostem jejich maticové reprezentace.

32.5.17 Použití sekvenční formy pro nalezení Nashova ekvilibria v nulových hrách

- Podobně jako u dualitního teorému a principu komplementární volnosti lze Nashova ekvilibria v extenzivních hrách s dvěma hráči nalézt pomocí sekvenční formy.
- **Teorém 2.66:** Dvojice realizačních plánů (x, y) tvoří Nashovo ekvilibrium v extenzivní hře s dokonalou pamětí, pokud existují vektory u a v takové, že:

$$\begin{aligned} x^\top (E^\top u - Ay) &= 0, & y^\top (F^\top v - B^\top x) &= 0, \\ Ex = e, \quad x \geq 0, \quad Fy = f, \quad y \geq 0, \\ E^\top u - Ay \geq 0, \quad F^\top v - B^\top x \geq 0. \end{aligned}$$

- Toto není lineární program (LP), ale tzv. **lineární komplementární problém**.
- Tyto problémy lze řešit pomocí Lemkeho algoritmu, který má podobnou složitost jako Lemke–Howsonův algoritmus.
- Navzdory exponenciálnímu růstu je tento přístup rychlejší než aplikace Lemke–Howsonova algoritmu na normální formu hry.

Intuice: Tato metoda umožňuje efektivní řešení Nashových ekvilibrií využitím vlastností sekvenční formy a struktury nulových her.

32.5.18 Další zdroje

- Více informací o hrách v extenzivní formě a implementaci algoritmů lze nalézt v přednáškách Martina Schmidha.

Intuice: Pokud je třeba podrobněji nastudovat algoritmy pro extenzivní hry, doporučuje se studovat specializované přednášky.

33 Návrh mechanismů (Mechanism Design)

33.1 Obecný koncept návrhu mechanismů

- Návrh herních pravidel tak, aby vedly k dosažení požadovaných cílů.
- Snažíme se navrhnout pravidla hry tak, aby strategické chování účastníků vedlo k žádoucímu výsledku.
- Proces zahrnuje několik kroků:
 1. **Aktéři (Actors):** Jednotlivci nebo organizace, kteří se hry účastní.
 2. **Zprávy (Messages):** Informace nebo volby, které účastníci sdělují.

3. **Mechanismus (Mechanism):** Instituce nebo pravidla hry.
 4. **Výsledek (Outcome):** Rozhodnutí nebo alokace zdrojů.
- Začínáme u jednoduchých aukcí na jeden předmět (single item auctions).
 - Následně tyto principy rozšiřujeme na obecnější prostředí s jedním parametrem (single-parameter environments) pomocí tzv. Myersonovy věty.

33.2 Aukce na jeden předmět (Single Item Auctions)

- **Situace:**
 - Prodávající (seller) nabízí jeden předmět (například obraz).
 - Počet zájemců (n) je libovolný a každý má zájem o daný předmět.
- **Hodnoty účastníků:**
 - Každý zájemce i má svou vlastní hodnotu v_i , kterou je ochoten za předmět zaplatit.
 - Hodnoty v_i nejsou známé ostatním účastníkům ani prodávajícímu.
- **Proces aukce:**
 - Každý zájemce soukromě sdělí svou nabídku b_i prodávajícímu.
 - Prodávající poté rozhodne, kdo předmět získá, a určí prodejní cenu p .
- **Užitky zájemců:**
 - Pokud zájemce aukci nevyhraje, jeho užitek $u_i = 0$.
 - Pokud vyhraje za cenu p , jeho užitek $u_i = v_i - p$.
- **Cíl návrhu mechanismu:**
 - Navrhnut mechanismus tak, aby alokace předmětu nebyla strategicky zmanipulovatelná.
 - Správně implementovat pravidla, která určují vítěze a prodejní cenu.

33.3 Jak nenavrhovat aukce na jeden předmět (How Not to Design a Single Item Auction)

- Ne každá volba pravidel vede k žádoucí aukci.
- Příklady nevhodných návrhů:
 - **Prodej předmětu zájemci i s nejvyšší nabídkou b_i zdarma:**
 - * Toto není dobrá volba, protože zájemci by měli motivaci nadzakovat své hodnoty v_i .

- * Aukce se tak změní na hru "kdo uvede nejvyšší číslo".
- **Prodej předmětu zájemci i s nejvyšší nabídkou b_i za cenu b_i :**
 - * Tento přístup je běžný, ale má nevýhody.
 - * Zájemci mají problém určit, jaká nabídka b_i je optimální. Pokud $b_i = v_i$, jejich užitek je $v_i - b_i = 0$.
 - * Zájemci by tak měli motivaci nabízet méně než v_i , ale jaká by měla být hodnota b_i ?

33.4 Požadavky na ideální aukci (So What Do We Want?)

- Formalizujeme podmínky, které by měla naše aukce splňovat.
- **Dominantní strategie:** Strategie, která maximalizuje užitek zájemce i bez ohledu na chování ostatních.
- **Sociální přebytek:** $\sum_{i=1}^n v_i x_i$, kde $x_i = 1$, pokud zájemce i vyhraje, a $x_i = 0$ jinak. Platí $\sum_{i=1}^n x_i \leq 1$.
- **Požadavky na aukci:**
 - **Silné záruky incentiv:** Aukce je dominantně strategicky kompatibilní (DSIC), tzn. zájemci mají strategii "říct pravdu".
 - **Silné výkonnostní záruky:** Aukce maximalizuje sociální přebytek při pravdivých nabídkách.
 - **Výpočetní efektivita:** Aukce může být implementována v polynomiálním čase.

33.5 Proč chceme tyto vlastnosti? (Why Do We Want This?)

- Odůvodnění požadavků na aukci:
 - **Silné záruky incentiv:** Vlastnost DSIC usnadňuje účastníkům rozhodování o své nabídce ($b_i = v_i$). Prodávající může lépe předvídat výsledek aukce, protože účastníci budou podávat pravdivé nabídky.
 - **Silné výkonnostní záruky:** DSIC samotné nestačí (například dát předmět zdarma náhodnému účastníkovi je DSIC). Tato vlastnost identifikuje účastníka s nejvyšší hodnotou v_i , i když je tato informace neveřejná.
 - **Výpočetní efektivita:** Požadavek na zřejmou praktičnost.

33.6 Vickreyho aukce (Vickrey's Auction)

- Existuje ideální aukce, která je překvapivě jednoduchá.
- **Vickreyho druhocenová aukce:** Vítězem je účastník i s nejvyšší nabídkou b_i , ale platí cenu rovnou druhé nejvyšší nabídce $p = \max_{j \in \{1, \dots, n\} \setminus \{i\}} b_j$.
- **Teorém (Vickrey, 1961):** Vickreyho aukce splňuje požadavky DSIC, silné výkonnostní záruky i výpočetní efektivitu.
- **Jednoduchá intuice:** Účastníci nemají motivaci lhát o své skutečné hodnotě v_i , protože jejich platba nezávisí na jejich vlastní nabídce, ale na druhé nejvyšší.
- **Důkaz:**
 - **Silné záruky incentiv:** Ukážeme, že užitek účastníka i je maximalizován pro $b_i = v_i$. Nechť $B = \max_{j \in \{1, \dots, n\} \setminus \{i\}} b_j$.
 - * Pokud $b_i < B$, i prohrává a jeho užitek je 0.
 - * Pokud $b_i \geq B$, i vyhrává a jeho užitek je $v_i - B$.
 - * Pokud $v_i < B$, maximální užitek je $\max\{0, v_i - B\} = 0$.
 - * Pokud $v_i \geq B$, maximální užitek je $v_i - B \geq 0$. Účastník získá tento užitek při pravdivé nabídce $b_i = v_i$.
 - **Silné výkonnostní záruky:** Pokud je účastník i vítěz, pak $v_i \geq v_j$ pro každého $j \neq i$. Sociální přebytek je rovný v_i a je maximalizován.
 - **Výpočetní efektivita:** Aukce běží v lineárním čase.

33.7 Jednoprocesorové prostředí (Single Parameter Environments)

Jednoduchá intuice: Jednoprocesorová prostředí rozšiřují aukce na složitější problémy, kde každý účastník má určitou hodnotu za jednotku zboží (například pásmo kapacity) a celý mechanismus zahrnuje rozdelení zdrojů i platby.

- V jednoprocesorovém prostředí je n účastníků, každý má privátní hodnotu v_i (hodnota za jednotku zboží).
- Existuje množina přípustných výsledků $X \subseteq \mathbb{R}^n$, která obsahuje vektory $x = (x_1, \dots, x_n)$, kde x_i značí část zdroje, o kterou účastník i stojí.
- Sealed-bid aukce v tomto prostředí probíhá ve třech krocích:
 1. Sběr nabídek $b = (b_1, \dots, b_n)$, kde b_i je nabídka účastníka i .
 2. Pravidlo alokace: Vybere přípustný výsledek alokace $x = x(b)$ na základě nabídek b .
 3. Pravidlo plateb: Určí platby $p(b) = (p_1(b), \dots, p_n(b)) \in \mathbb{R}^n$ na základě nabídek b .
- Pár (x, p) tvoří mechanismus.
- Užitek účastníka i je $u_i(b) = v_i \cdot x_i(b) - p_i(b)$.

33.8 Poznámky k jednoprocesorovým prostředím (Single Parameter Environments: Remarks)

- Platby $p_i(b)$ patří do intervalu $[0, b_i \cdot x_i(b)]$ pro každého účastníka i a všechny nabídky b .
 - Protože $p_i(b) \geq 0$, prodávající nikdy neplatí účastníkům.
 - Podmínka $p_i(b) \leq b_i \cdot x_i(b)$ zajišťuje, že účastníci neplatí více, než je jejich hodnota b_i za alokované množství $x_i(b)$.
 - Tato podmínka zajišťuje, že účastníci mají nezáporný užitek, pokud podávají pravdivé nabídky.
- Základní dilema návrhu mechanismů spočívá v optimalizaci globálního cíle, například sociálního přebytku $\sum_{i=1}^n v_i \cdot x_i(b)$.

33.9 Jednoprocesorové prostředí: Aukce na jeden předmět (Single Parameter Environments: Single-Item Auctions)

Jednoduchá intuice: Jednoprocesorové prostředí obsahuje různé druhy aukcí. V případě aukce na jeden předmět soutěží účastníci o jediné zboží a pouze jeden z nich ho může získat.

- Jednoprocesorová prostředí zahrnují aukce na jeden předmět.
- V aukcích na jeden předmět soutěží n zájemců o jediný předmět prodávajícího.
- Každý zájemce buď předmět získá, nebo ne, ale pouze jeden může být vítěz.
- Toto prostředí lze modelovat nastavením $x_i \in \{0, 1\}$ pro každého zájemce i a výběrem přípustné množiny $X = \{(x_1, \dots, x_n) \in \{0, 1\}^n : \sum_{i=1}^n x_i \leq 1\}$.
- Cílem je navrhnout aukci tak, aby vítězem byl zájemce s nejvyšší hodnotou v_i .

33.10 Jednoprocesorové prostředí: Sponzorované odkazy (Single Parameter Environments: Sponsored-Search)

Jednoduchá intuice: Mechanismy pro sponzorované odkazy se používají v reálných situacích, například v internetových vyhledávačích, kde firmy soutěží o pozice na stránce výsledků vyhledávání.

- Uvažujme následující reálnou motivaci:
 - Stránka s výsledky vyhledávání obsahuje seznam organických výsledků a seznam k sponzorovaných odkazů, za které zaplatili inzerenti.

- Při každém zadání vyhledávacího dotazu probíhá v reálném čase aukce, která rozhodne, které sponzorované odkazy se zobrazí, v jakém pořadí a za jakou cenu.
- Pozice k prodeji jsou k ”slotů” pro sponzorované odkazy. Sloty výše na stránce jsou hodnotnější, protože lidé obvykle procházejí stránku odshora dolů.
- Zájemci jsou inzerenti, kteří mají nastavenou nabídku na klíčové slovo, na které se hledalo.
- Máme dvě předpoklady:
 - Čím výše je slot, tím vyšší je pravděpodobnost α_j , že bude kliknut.
 - Míra prokliku (click-through rate) nezávisí na tom, kdo je v daném slotu.
- Každá pozice j má hodnotu α_j , kde $\alpha_1 > \dots > \alpha_k > 0$.
- Přípustná množina X obsahuje vektory (x_1, \dots, x_n) , kde každý x_i leží v $\{\alpha_1, \dots, \alpha_k, 0\}$ a pokud $x_i = x_j$, pak $x_i = 0 = x_j$.
- Hodnota slotu j pro zájemce i je $v_i \alpha_j$.
- Cílem je maximalizace sociálního přebytku.

33.11 Navrhování skvělých mechanismů (Designing Awesome Mechanisms)

Jednoduchá intuice: Pro každý mechanismus (x, p) v jednoprocesorovém prostředí bychom měli zajistit, že splňuje vlastnost DSIC, což znamená, že pravidlivé nabízení je optimální strategií.

- Chceme navrhovat smysluplné mechanismy (x, p) pro dané jednoprocesorové prostředí, které jsou ideálně skvělé.
- Zejména bychom měli zajistit, že (x, p) má vlastnost DSIC.
- Pravidlo alokace x pro jednoprocesorové prostředí je implementovatelné, pokud existuje pravidlo plateb p takové, že (x, p) je DSIC.
 - Pravidlo alokace ”přiřad předmět zájemci s nejvyšší nabídkou” je implementovatelné v případě aukcí na jeden předmět, protože pravidlo druhé ceny poskytuje DSIC mechanismus.
 - Situace je mnohem méně jasná pro pravidlo alokace ”přiřad předmět zájemci s druhou nejvyšší nabídkou”.
- Pravidlo alokace x je monotónní, pokud pro každého zájemce i a všechny nabídky b_{-i} ostatních zájemců platí, že alokace $x_i(z; b_{-i})$ pro i je neklesající v jeho nabídce z .
 - První pravidlo výše je monotónní, zatímco druhé nikoliv.

34 Myersonova věta (Myerson's Lemma)

Jednoduchá intuice: Myersonova věta poskytuje klíčový nástroj pro návrh DSIC (dominantně strategicky kompatibilních) mechanismů. Spojuje monotónnost alokačního pravidla s jeho implementovatelností.

34.1 Formulace Myersonovy věty

- V jednoprocесорovém prostředí platí následující tři tvrzení:
 - [a] Alokační pravidlo je implementovatelné, pokud a pouze pokud je monotónní.
 - Pokud je alokační pravidlo x monotónní, pak existuje jedinečné pravidlo plateb p , takové že mechanismus (x, p) je DSIC (za předpokladu, že $b_i = 0$ implikuje $p_i(b) = 0$).
 - Pravidlo plateb p je dáno následující explicitní formulí:

$$p_i(b_i; b_{-i}) = \int_0^{b_i} z \cdot \frac{\partial}{\partial z} x_i(z; b_{-i}) dz,$$

pro každé $i \in \{1, \dots, n\}$.

34.2 Poznámka

- Důkaz věty bude prezentován později, nyní se zaměříme na aplikace Myersonovy věty.

34.3 Aplikace Myersonovy věty I: Aukce na jeden předmět

Jednoduchá intuice: Myersonova věta nám umožňuje ukázat, jak pravidlo druhé ceny zajišťuje DSIC mechanismus při aukci na jeden předmět.

- Uvažujme aukce na jeden předmět.
- Nechť zájemce i a nabídky b_{-i} ostatních zájemců jsou fixní a $B = \max_{j \in \{1, \dots, n\} \setminus \{i\}} b_j$.
- Pokud alokujeme předmět zájemci s nejvyšší nabídkou, pak alokační funkce $x_i(\cdot; b_{-i})$ je 0 až do B a poté 1.
- Zřejmé je, že toto alokační pravidlo je monotónní.
- Pokud je i nejvyšší zájemce, pak $b_i > B$ a (jedinečné) pravidlo plateb z Myersonovy věty je $p_i(b_i; b_{-i}) = B$ a užitek zájemce i je $v_i \cdot x_i(b_i; b_{-i}) - B = v_i - B$.
- Pokud $b_i \leq B$, pak platba i užitek zájemce i jsou nulové.
- Pokud $v_i > B$, jeho užitek je pozitivní a užitek všech ostatních zájemců je nulový.
- Celkově tím získáváme pravidlo druhé ceny.

34.4 Aplikace Myersonovy věty II: Aukce na sponzorované odkazy

Jednoduchá intuice: Myersonova věta se aplikuje i na složitější aukce, jako jsou aukce na sponzorované odkazy, kde více zájemců soutěží o různé pozice na stránce.

- Pokračujeme s aukcemi na sponzorované odkazy.
- Nechť x je alokační pravidlo, které přiřazuje i -tou nejlepší pozici i -tému nejvyššímu zájemci. Pravidlo x je monotónní a při pravdivém nabízení maximalizuje sociální přebytek.
- Z Myersonovy věty plyne, že existuje jedinečné a explicitní pravidlo plateb p , takové že mechanismus (x, p) je DSIC.
- Předpokládejme, že zájemce i učiní i -tou nejvyšší nabídku, tj. $b_1 \geq \dots \geq b_n$.
- Představme si zájemce 1, který zvyšuje svou nabídku z z 0 na b_1 , zatímco ostatní nabídky jsou fixní. Alokační funkce $x_1(z; b_{-1})$ roste z 0 na α_1 při z rostoucím z 0 na b_1 , s přeskokem $\alpha_j - \alpha_{j+1}$ v bodě, kde se z stává j -tou nejvyšší nabídkou v profilu $(z; b_{-1})$, tj. b_{j+1} .
- Obecně pro i -tého nejvyššího zájemce Myersonova věta poskytuje vzorec pro platby:

$$p_i(b) = \sum_{j=i}^k b_{j+1}(\alpha_j - \alpha_{j+1}),$$

kde $\alpha_{k+1} = 0$.

34.5 Co jsme se naučili posledně

Jednoduchá intuice: Tento přehled shrnuje klíčové body z předchozích lekcí o aukcích na jeden předmět a jejich rozšíření na obecnější prostředí s jedním parametrem. Důraz je kladen na vlastnosti DSIC mechanismů a jejich efektivitu.

- Pro aukce na jeden předmět jsme našli *skvělou aukci* (Vickreyho aukci) s následujícími vlastnostmi:
 - **DSIC:** Každý účastník má dominantní strategii nabízet pravdivě, což zaručuje nenegativní užitek.
 - **Silný výkon:** Maximalizuje sociální přebytek.
 - **Výpočetní efektivita:** Běží v polynomiálním čase.
- Poté jsme zobecnili aukce na jeden předmět na prostředí s jedním parametrem.

- 1 prodávající a n účastníků aukce, prodávající sbírá nabídky $b = (b_1, \dots, b_n)$.
- Prodávající volí alokaci $x(b) = (x_1(b), \dots, x_n(b))$ z množiny možných alokací X .
- Prodávající určuje platby $p(b) = (p_1(b), \dots, p_n(b))$.
- Hledali jsme mechanismy (x, p) s vlastností DSIC pro návrh těchto aukcí.

34.6 Maximalizace sociálního přebytku

Jednoduchá intuice: Zatímco DSIC mechanismy maximalizují sociální přebytek, nezaměřují se na příjem prodávajícího (*revenue*). Tento rozdíl má význam zejména v reálných aplikacích, jako jsou vládní aukce na bezdrátové spektrum.

- Dospod jsme navrhovali DSIC mechanismy pro prostředí s jedním parametrem, které maximalizují sociální přebytek $\sum_{i=1}^n v_i x_i(b)$.
- Nezabývali jsme se příjmem (*revenue*) $\sum_{i=1}^n p_i(b)$.
- Toto je smysluplné v reálných scénářích, kde příjem není hlavním cílem (např. vládní aukce pro prodej bezdrátového spektra).
- V aukcích na jeden předmět maximalizuje sociální přebytek Vickreyho aukce.
- V obecnějších prostředích s jedním parametrem využíváme Myersonovu větu.
- Dále se zaměříme na maximalizaci příjmu.

34.7 Maximalizace příjmu

Jednoduchá intuice: Nyní se zaměřujeme na to, jak optimalizovat příjem prodávajícího, což činí situaci složitější než při maximalizaci sociálního přebytku.

- Jak maximalizovat příjem $\sum_{i=1}^n p_i(b)$?
- Situace se stává složitější, ale dnes uvidíme zajímavé výsledky.

34.8 Proč je to složitější?

Jednoduchá intuice: Při maximalizaci příjmu narážíme na problém, že neznáme skutečné hodnoty účastníků aukce. To ztěžuje rozhodování, jak nastavit cenu.

- Zvažme aukci na jeden předmět s jedním účastníkem, který má hodnotu v .
- Jediná DSIC aukce: prodávající stanoví cenu r . Příjem je r , pokud $v \geq r$, a 0, pokud $v < r$.

- Maximalizace sociálního přebytku je triviální při nastavení $r = 0$.
- Při maximalizaci příjmu ale nevíme, jak nastavit r , protože neznáme hodnotu v .

34.9 Nový model

Jednoduchá intuice: Zavádíme pravděpodobnostní model, abychom mohli analyzovat kompromisy mezi různými vstupy a maximalizovat příjem v nejistém prostředí.

- Potřebujeme model, který umožní analyzovat kompromisy mezi výkonem na různých vstupech.
- Zavádíme prvky pravděpodobnostní teorie:
 - Pokud F je distribuční funkce hodnoty v , pak $F(z)$ je pravděpodobnost, že v má hodnotu nejvýše z .
 - Pokud F má hustotu f s podporou $[0, v_{\max}]$, pak:

$$f(x) = \frac{d}{dx} F(x),$$

$$F(x) = \int_0^x f(z) dz.$$

- Očekávaná hodnota náhodné proměnné X :

$$\mathbb{E}_{z \sim F}[X(z)] = \int_0^{v_{\max}} X(z) \cdot f(z) dz.$$

34.10 Bayesovský model

Jednoduchá intuice: Bayesovský model předpokládá, že hodnoty účastníků jsou náhodně generovány z distribučních funkcí, které zná pouze prodávající, což umožňuje analyzovat očekávaný příjem.

- Bayesovský model se skládá z:
 - Prostředí s jedním parametrem (x, p) .
 - Hodnoty v_i každého účastníka jsou generovány z distribuční funkce F_i s hustotou f_i na $[0, v_{\max}]$.
 - Předpokládáme, že distribuce F_1, \dots, F_n jsou nezávislé, ale nemusí být stejné.
- Distribuce F_1, \dots, F_n jsou známy pouze návrháři mechanismu.
- Cíl: maximalizovat očekávaný příjem mezi všemi DSIC aukcemi:

$$\mathbb{E}_{v=(v_1, \dots, v_n) \sim (F_1 \times \dots \times F_n)} \left[\sum_{i=1}^n p_i(v) \right].$$

34.11 Příklad: aukce s jedním účastníkem na jeden předmět

Jednoduchá intuice: Tento příklad ukazuje, jak maximalizovat očekávaný příjem při aukci s jedním účastníkem. Výsledek závisí na pravděpodobnostní distribuci hodnoty účastníka.

- Zvažme příklad s jedním účastníkem v aukci na jeden předmět.
- Očekávaný příjem z dané ceny r je:

$$r \cdot (1 - F_1(r)),$$

kde $(1 - F_1(r))$ je pravděpodobnost, že hodnota v je alespoň r .

- Pokud známe distribuční funkci F_1 , můžeme maximalizovat očekávaný příjem výběrem optimální ceny r .
- Například pokud je F_1 rovnoměrné rozložení na $[0, 1]$, tedy $F_1(x) = x$ pro $x \in [0, 1]$, maximální příjem dosáhneme při volbě $r = 1/2$.
- Cena r , která maximalizuje očekávaný příjem, se nazývá **monopolní cena**.

34.12 Příklad: aukce se dvěma účastníky na jeden předmět

Jednoduchá intuice: Tento příklad rozšiřuje koncept na dva účastníky a ukazuje, že existuje více způsobů, jak navrhnout DSIC aukce, přičemž některé maximalizují očekávaný příjem lépe než jiné.

- Zvažme aukci na jeden předmět se dvěma účastníky, jejichž hodnoty v_1 a v_2 jsou nezávisle generovány z rovnoměrného rozdělení na $[0, 1]$.
- Vickreyho aukce je příkladem DSIC aukce, kde očekávaný příjem je $1/3$.
- Existují však i jiné DSIC aukce, které z hlediska očekávaného příjmu dosahují lepších výsledků.
- Dnes se zaměříme na návrh takových optimálních aukcí pomocí Myersonova lematu.

34.13 Maximalizace očekávaného příjmu

Jednoduchá intuice: Myersonova věta poskytuje užitečný nástroj pro charakterizaci DSIC aukcí, které maximalizují očekávaný příjem, pomocí konceptu virtuálních hodnot.

- Abychom charakterizovali DSIC aukce, které maximalizují očekávaný příjem, odvodíme užitečný vzorec pro očekávaný příjem.

- Myersonova věta: Nechť (x, p) je DSIC mechanismus v prostředí s jedním parametrem s Bayesovským modelem s n účastníky a distribučními funkcemi F_1, \dots, F_n s hustotami f_1, \dots, f_n .

$$\mathbb{E}_{v \sim F} \left[\sum_{i=1}^n p_i(v) \right] = \mathbb{E}_{v \sim F} \left[\sum_{i=1}^n \varphi_i(v_i) \cdot x_i(v) \right],$$

kde

$$\varphi_i(v_i) = v_i - \frac{1 - F_i(v_i)}{f_i(v_i)}$$

je virtuální hodnota účastníka i .

34.14 Poznámky k maximalizaci očekávaného příjmu

Jednoduchá intuice: Očekávaný příjem DSIC aukce je roven očekávanému virtuálnímu sociálnímu přebytku. Virtuální hodnoty jsou klíčovým konceptem pro optimalizaci příjmu.

- Jinými slovy, pro každou DSIC aukci je očekávaný příjem roven očekávanému virtuálnímu sociálnímu přebytku $\sum_{i=1}^n \varphi_i(v_i) \cdot x_i(v)$.
- Při maximalizaci očekávaného příjmu tedy maximalizujeme podobnou hodnotu jako dříve.
- Virtuální hodnota $\varphi_i(v_i) = v_i - \frac{1 - F_i(v_i)}{f_i(v_i)}$ účastníka může být záporná a závisí pouze na v_i a F_i , nikoli na parametrech ostatních.
- **Příklad:** Zvažme účastníka i s hodnotou v_i , která je rovnoměrně generována na $[0, 1]$.

$$F_i(z) = z, \quad f_i(z) = 1, \quad \varphi_i(z) = z - \frac{1-z}{1} = 2z - 1 \in [-1, 1].$$

- Hrubá intuice za virtuálními hodnotami: první člen v_i v tomto výrazu lze považovat za maximální příjem dosažitelný od účastníka i a druhý člen za nevyhnutelnou ztrátu příjmu způsobenou neznalostí v_i předem.

34.15 Důkaz maximalizace očekávaného příjmu

Krok 1: Použití Myersonovy věty

Jednoduchá intuice: Použijeme Myersonovu větu k odvození explicitního vyjádření pro platby v DSIC aukci.

- Protože (x, p) je DSIC, předpokládáme $b_i = v_i$ pro každého biddera i .
- Podle Myersonovy věty máme vzorec pro platby:

$$p_i(b; b_{-i}) = \int_0^{b_i} z \cdot \frac{d}{dz} x_i(z; b_{-i}) dz.$$

- Tento vzorec platí pro každou monotónní funkci $x_i(z; b_{-i})$, pokud vhodně interpretujeme derivaci $\frac{d}{dz}x_i(z; b_{-i})$.
- Očekávaná platba pro *biddera i* při fixaci v_{-i} je:

$$\mathbb{E}_{v_i \sim F_i}[p_i(v)] = \int_0^{v_{\max}} p_i(v) f_i(v_i) dv_i.$$

- Dosazením dostaváme:

$$\mathbb{E}_{v_i \sim F_i}[p_i(v)] = \int_0^{v_{\max}} \left(\int_0^{v_i} z \cdot \frac{d}{dz} x_i(z; v_{-i}) dz \right) f_i(v_i) dv_i.$$

Krok 2: Reverze integrace a zjednodušení

Jednoduchá intuice: Použijeme Fubiniho teorém a integraci po částech k dalšímu zjednodušení.

- Reverzí integrace dostaváme:

$$\int_0^{v_{\max}} \left(\int_0^{v_i} z \cdot \frac{d}{dz} x_i(z; v_{-i}) dz \right) f_i(v_i) dv_i = \int_0^{v_{\max}} \left(\int_z^{v_{\max}} f_i(v_i) dv_i \right) z \cdot \frac{d}{dz} x_i(z; v_{-i}) dz.$$

- Použijeme fakt, že $\int_z^{v_{\max}} f_i(v_i) dv_i = 1 - F_i(z)$:

$$\int_0^{v_{\max}} (1 - F_i(z)) \cdot z \cdot \frac{d}{dz} x_i(z; v_{-i}) dz.$$

- Aplikujeme integraci po částech:

$$\int fg' = fg - \int f'g,$$

kde volíme $f(z) = (1 - F_i(z)) \cdot z$ a $g'(z) = \frac{d}{dz} x_i(z; v_{-i})$.

Krok 3: Finální derivace

Jednoduchá intuice: Dokončíme důkaz aplikací definice virtuálních hodnot a linearity očekávání.

- První článek integrace po částech je nulový, protože:

$$[(1 - F_i(z)) \cdot z \cdot x_i(z; v_{-i})]_0^{v_{\max}} = 0.$$

- Zbývající článek lze zapsat jako:

$$\int_0^{v_{\max}} x_i(z; v_{-i}) \cdot (1 - F_i(z) - zf_i(z)) dz.$$

- Použitím definice $\varphi_i(v_i) = v_i - \frac{1-F_i(v_i)}{f_i(v_i)}$ dostáváme:

$$\mathbb{E}_{v_i \sim F_i}[p_i(v)] = \int_0^{v_{\max}} \varphi_i(z) \cdot x_i(z; v_{-i}) f_i(z) dz.$$

- Lineáritou očekávání a součtu:

$$\mathbb{E}_{v \sim F} \left[\sum_{i=1}^n p_i(v) \right] = \mathbb{E}_{v \sim F} \left[\sum_{i=1}^n \varphi_i(v_i) \cdot x_i(v) \right].$$

34.16 Maximalizace očekávaného virtuálního sociálního přebytku

Jednoduchá intuice:

Tato část dokazuje, že maximalizace očekávaného příjmu je ekvivalentní maximalizaci očekávaného virtuálního sociálního přebytku. Ukazuje, že konkrétní typ aukce, Vickreyho aukce s rezervní cenou, je optimální pro maximalizaci očekávaného příjmu.

34.16.1 Maximalizace očekávaného virtuálního sociálního přebytku

- Podle Věty 3.13 je **maximalizace očekávaného příjmu ekvivalentní maximalizaci očekávaného virtuálního sociálního přebytku**.
- Začínáme s aukcemi na jeden předmět a předpokládáme, že existuje pravděpodobnostní distribuce F taková, že $F_1 = F_2 = \dots = F_n = F$ (tedy všechny virtuální hodnoty $\varphi_1, \dots, \varphi_n$ jsou ekvivalentní nějaké funkci φ).
- Dále předpokládáme, že F je **regulární**, což znamená, že $\varphi(v)$ je striktně rostoucí v v .
- Příkladem může být rovnoměrná distribuce na $[0, 1]$, kde $F(v) = v$. Potom je:

$$\varphi(v) = v - \frac{1 - F(v)}{f(v)} = 2v - 1.$$

- Ukážeme, že pro takovéto aukce maximalizuje očekávaný příjem Vickreyho aukce s rezervní cenou.
- Ve **Vickreyho aukci s rezervní cenou r** je předmět prodán nejvyššímu nabízejícímu, pokud všechny nabídky jsou vyšší než r . Jinak není předmět prodán. Vítěz zaplatí druhou nejvyšší nabídku nebo r , podle toho, která hodnota je větší.

34.16.2 Optimálnost Vickreyho aukce s rezervní cenou

[3.14] Nechť F je regulární pravděpodobnostní distribuce s hustotou f a virtuální hodnotou φ . Nechť F_1, \dots, F_n jsou nezávislé pravděpodobnostní distribuce hodnot n účastníků, kde $F = F_1 = \dots = F_n$ (a tedy $\varphi = \varphi_1 = \dots = \varphi_n$). Potom Vickreyho aukce s rezervní cenou $\varphi^{-1}(0)$ maximalizuje očekávaný příjem.

Důkaz:

- Podle Věty 3.13 maximalizujeme očekávaný virtuální sociální přebytek.
- K tomu si můžeme zvolit $x(v) \in X$ pro každý vstup v , avšak nemáme kontrolu nad F a $\varphi(v_i)$.
- V aukci na jeden předmět je omezující podmínka proveditelnosti $\sum_{i=1}^n x_i(v) \leq 1$. Předmět je přidělen účastníkovi i s nejvyšší hodnotou $\varphi(v_i)$. Pokud všichni účastníci mají zápornou virtuální hodnotu, není předmět nikomu přidělen.
- Zbývá ukázat, že tato alokační pravidla x jsou monotónní. Poté, podle Myersonovy věty, je lze rozšířit na DSIC aukci (x, p) .
- Protože F je regulární, funkce φ je striktně rostoucí. Proto je x monotónní, což vede na Vickreyho aukci s rezervní cenou $\varphi^{-1}(0)$.

34.17 Rezervní cena a maximalizace příjmu

Jednoduchá intuice

Tato část poskytuje shrnutí významu rezervní ceny v aukcích a uvádí její praktické důsledky.

Popis

- Tuto teorii vyvinul Roger Myerson.
- Výsledek lze rozšířit i na jedno-parametrická prostředí. Při větší námaze lze dokázat verzi pro distribuce, které nejsou regulární.
- Pokud prodávající věří, že dražitelé mají vysoké hodnoty, měl by podle toho nastavit vysokou rezervní cenu.

34.18 Obecnější optimální aukce

Jednoduchá intuice

Zde rozšiřujeme teorii optimálních aukcí na obecnější podmínky.

Popis

- Existují optimální DSIC aukce, i když podmínky jako identické distribuce F_1, \dots, F_n nejsou splněny.
- Tyto optimální aukce však mohou být velmi složité a obvykle se nepodobají aukcím používaným v praxi.

35 The Bulow–Klemperer Theorem

Teorie: Nechť $F = F_1 = \dots = F_n$ je regulární pravděpodobnostní rozdělení. Potom platí:

$$\mathbb{E}_{v_1, \dots, v_{n+1} \sim F} [\text{Rev}(VA_{n+1})] \geq \mathbb{E}_{v_1, \dots, v_n \sim F} [\text{Rev}(\text{OPT}_{F,n})],$$

kde $\text{Rev}(VA_{n+1})$ značí výnos Vickreyovy aukce VA_{n+1} s $n+1$ účastníky (bez rezervy) a $\text{Rev}(\text{OPT}_{F,n})$ značí výnos optimální aukce $\text{OPT}_{F,n}$ pro F s n účastníky.

Intuice: Přidání dalšího účastníka do aukce Vickrey zvyšuje její očekávaný výnos a dokonce překonává očekávaný výnos optimální aukce s menším počtem účastníků.

Poznámky k teorému

- Pokud $F = F_1 = \dots = F_n$ je regulární pravděpodobnostní rozdělení, pak stále platí:

$$\mathbb{E}_{v_1, \dots, v_{n+1} \sim F} [\text{Rev}(VA_{n+1})] \geq \mathbb{E}_{v_1, \dots, v_n \sim F} [\text{Rev}(\text{OPT}_{F,n})].$$

- $\text{OPT}_{F,n}$ je Vickreyova aukce s rezervní cenou $\varphi^{-1}(0)$, kde φ je virtuální hodnotící funkce F .
- Aukce vlevo (Vickreyova) nezávisí na konkrétním rozdělení F , zatímco aukce vpravo (optimální) ano.
- Bulow–Klemperer teorém implikuje, že ve všech aukcích s jedním předmětem, kde $n \geq 2$, je očekávaný výnos Vickreyovy aukce alespoň $\frac{n-1}{n}$ -násobkem očekávaného výnosu optimální aukce.
- **Neformálně:** Další konkurence je důležitější než precizní nastavení formátu aukce.

Intuice: Tento teorém zdůrazňuje, že přidání účastníků do aukce má větší vliv na výnos než optimalizace podmínek aukce samotné.

Důkaz

- Definujeme pomocnou aukci \mathcal{A} s $n + 1$ účastníky následujícím způsobem:
 - Simulujeme optimální aukci $\text{OPT}_{F,n}$ na účastnících $1, \dots, n$.
 - Pokud předmět nebyl přidělen, dáme jej účastníkovi $n + 1$ zdarma.
- Aukce \mathcal{A} vždy přidělí předmět a platí:

$$\mathbb{E}_{v_1, \dots, v_{n+1} \sim F} [\text{Rev}(\mathcal{A})] = \mathbb{E}_{v_1, \dots, v_n \sim F} [\text{Rev}(\text{OPT}_{F,n})].$$

- Vickreyova aukce maximalizuje očekávaný výnos mezi všemi aukcemi, které vždy přidělí předmět.

Intuice důkazu: Přidáním dodatečného účastníka $n + 1$ je možné vždy dosáhnout vyššího nebo minimálně stejného očekávaného výnosu jako optimální aukce s méně účastníky díky jednoduchému principu přidělení předmětu vítězi s nejvyšší hodnotou.

Knapsack Auction

Definice: V aukci typu "knapsack" s n účastníky $1, \dots, n$ má každý účastník i veřejně známou velikost $w_i \geq 0$ a soukromou hodnotu $v_i \geq 0$. Existuje jediný prodejce s kapacitou $W \geq 0$. Množina přípustných přidělení X zahrnuje $\{0, 1\}$ -vektory (x_1, \dots, x_n) takové, že:

$$\sum_{i=1}^n x_i w_i \leq W,$$

kde $x_i = 1$ značí, že účastník i získal předmět.

Mechanismus: Snažíme se navrhnout optimální mechanismus pro tuto aukci.

- Pro nabídky $b = (b_1, \dots, b_n)$ vybíráme $x(b)$ z X , aby:

$$\sum_{i=1}^n b_i x_i$$

bylo maximalizováno. Když účastníci nabízejí pravdivě, maximalizuje se společenský přebytek.

- Pravidlo přidělení x je monotónní (jednokroková funkce s bodovým zlomem v z), a proto podle Myersonova lemmatu existuje platební pravidlo p , takže (x, p) je DSIC. Pokud $b_i < z$, účastník i nic neplatí, jinak platí $z \cdot (1 - 0) = z$.
- Třetí podmínka však představuje problém, protože x řeší Knapsack problém.

36 Knapsack Problem

Formulace: Je dána kapacita W a n předmětů s hodnotami v_1, \dots, v_n a velikostmi w_1, \dots, w_n . Najděte podmnožinu předmětů s maximální celkovou hodnotou tak, aby celková velikost nepřekročila W .

- Tento problém je NP-těžký.
- Existuje pseudopolynomiální algoritmus využívající dynamické programování a plně polynomiální aproximační schéma.

37 We Are Not Always Awesome

- Pokud platí $P \neq NP$, nemůžeme splnit třetí podmínu (polynomiální čas), protože Knapsack problém je NP-těžký. Volba pravidla přidělení, která maximalizuje společenský přebytek, řeší tento problém. Proto aukce typu knapsack nejsou ideální.
- Uvolnění první podmínky (DSIC vlastnost) nepomáhá, protože se střetávají poslední dvě podmínky. Můžeme uvolnit třetí podmínu, například na pseudopolynomiální čas. To je užitečné, pokud jsou naše instance malé nebo dostatečně strukturované a máme dostatek času a výpočetního výkonu pro implementaci optimálního maximalizování přebytku.
- Dominantní přístup je uvolnit druhou podmínu (optimální přebytek) co nejméně, za předpokladu splnění první (DSIC) a třetí (polynomiální čas).
- Myersonův lemmata implikuje, že následující cíl je ekvivalentní: navrhněte pravidlo přidělení v polynomiálním čase, které je monotónní a co nejvíce se blíží maximalizaci společenského přebytku.

Multi-parameter Mechanism Design

Jednoduché mechanismy: V *single-parameter mechanism design* má každý účastník pouze jeden soukromý parametr – jeho hodnotu v_i .

Obecný model: V modelu *multi-parameter mechanism design* účastníci mají různé soukromé hodnoty pro různé výsledky.

Multi-parameter Mechanism Design Formally

V obecném modelu *multi-parameter mechanism design* máme:

- n strategických účastníků,
- konečnou množinu Ω možných výsledků,

- každý účastník i má soukromou hodnotu $v_i(\omega) \geq 0$ pro každý výsledek $\omega \in \Omega$.

Každý účastník i podává nabídky $b_i(\omega) \geq 0$ pro každý výsledek $\omega \in \Omega$. Cílem je navrhnout mechanismus, který vybere výsledek $\omega \in \Omega$ tak, aby maximalizoval společenský přebytek:

$$\sum_{i=1}^n v_i(\omega).$$

Příklad: Jednopředmětová aukce:

- Nastavíme $\Omega = \{\omega_1, \dots, \omega_n, \omega_\emptyset\}$, kde ω_i znamená, že předmět získá účastník i .
- Hodnoty jsou $v_i(\omega_j) = 0$ pro $j \neq i$ a $v_i(\omega_i) = v_i$.

Multi-parameter Mechanism: Example

Příklad: Uvažujme dva účastníky (1 a 2) v aukci, kde prodejce prodává dva předměty t_1 a t_2 , přičemž každý účastník může získat pouze jeden předmět.

- Množina možných výsledků je:

$$\Omega = \{(0, 0), (0, 1), (0, 2), (1, 0), (2, 0), (1, 2), (2, 1)\},$$

kde (i, j) znamená, že účastník 1 získá předmět t_i a účastník 2 získá t_j .

- Hodnoty účastníků:
 - Účastník 1: $v_1(1, j) = 10$, $v_1(2, j) = 5$.
 - Účastník 2: $v_2(i, 1) = 5$, $v_2(i, 2) = 3$.
- Optimální přidělení: t_1 získá účastník 1, t_2 účastník 2. Společenský přebytek: $10 + 3 = 13$.

Mechanismus VCG zajišťuje, že tento výsledek je DSIC a maximalizuje společenský přebytek.

38 Multi-parameter mechanism design

Intuice: Tento slide popisuje nastavení pro mechanism design, kde je cílem navrhnout mechanismus maximalizující tzv. *společenský přebytek* (social surplus). Účastníci (strategičtí dražitelé) mají různé preference (označované jako *hodnoty*) pro různé výsledky (outcomes), přičemž tyto hodnoty nejsou veřejné.

- Mechanism design s více parametry zahrnuje:

- n strategických dražitelů,
 - konečnou množinu výsledků Ω ,
 - každého dražitele i , který má pro každý výsledek $\omega \in \Omega$ soukromou hodnotu $v_i(\omega) \geq 0$.
- Každý dražitel i odevzdá svou nabídku $b_i(\omega) \geq 0$ pro každý výsledek $\omega \in \Omega$.
 - Cílem je navrhnout mechanismus, který vybere výsledek $\omega \in \Omega$ tak, aby maximalizoval společenský přebytek $\sum_{i=1}^n v_i(\omega)$.
 - Hodnoty dražitelů mohou záviset na různých výsledcích. Například při soutěži o jedinou položku může dražitel preferovat, pokud vyhraje, ale zároveň může mít názor na to, kdo jiný by měl vyhrát.
 - **Příklad (dražba jedné položky):** Nastavíme $\Omega = \{\omega_1, \dots, \omega_n, \omega_\emptyset\}$, kde každý výsledek ω_i odpovídá vítězi i a ω_\emptyset odpovídá situaci, kdy nikdo nevyhraje. Hodnoty jsou definovány jako $v_i(\omega_j) = 0$ pro $j \neq i$ a $v_i(\omega_i) = v_i$ jinak.

39 The Vickrey-Clarke-Groves (VCG) mechanism

Intuice: Tento slide představuje mechanismus Vickrey-Clarke-Groves (VCG), který garantuje, že v prostředí víceparametrového mechanism designu lze vždy navrhnout mechanismus, jenž maximalizuje společenský přebytek a zároveň je *dominantně strategicky kompatibilní* (DSIC).

- V prostředí víceparametrového mechanism designu existuje mechanismus, který je DSIC a maximalizuje společenský přebytek.
- Tento mechanismus je známý jako Vickrey-Clarke-Groves (VCG) mechanismus.
- Tvrzení je formálně zapsáno jako Věta 3.18.

40 VCG Mechanism: Proof Idea

Intuice: Tento slide představuje klíčovou myšlenku důkazu mechanismu VCG. Hlavní myšlenkou je zohlednit ztrátu společenského přebytku, kterou dražitel i způsobuje ostatním $n - 1$ dražitelům tím, že se účastní. Například v dražbě jedné položky vítěz způsobuje ztrátu druhé nejvyšší nabídky ostatním.

- Klíčová myšlenka: Zohlednění ztráty společenského přebytku způsobené dražitelem i na ostatní $n - 1$ dražitelů.

- Definujeme platby tak, aby každý dražitel bral ohled na ostatní.
- Používáme následující pravidlo alokace:

$$x(b) = \arg \max_{\omega \in \Omega} \sum_{i=1}^n b_i(\omega) \quad (1)$$

- Platby se počítají podle následujícího vzorce:

$$p_i(b) = \max_{\omega \in \Omega} \sum_{j \neq i} b_j(\omega) - \sum_{j \neq i} b_j(\omega^*), \quad (2)$$

kde $\omega^* = x(b)$ je výsledek zvolený pravidlem alokace.

41 VCG Mechanism: Proof I

Intuice: Důkaz se provádí ve dvou krocích: Nejprve se předpokládá, že dražitelé odhalují své pravé hodnoty, a poté se určuje pravidlo plateb, aby mechanismus byl DSIC (dominantně strategicky kompatibilní).

- Postupujeme ve dvou krocích:
 1. Předpokládáme, že dražitelé pravdivě odhalují své informace, a určujeme, který výsledek z Ω zvolit.
 2. Definujeme pravidlo plateb $p = (p_1, \dots, p_n)$ tak, aby mechanismus (x, p) byl DSIC.
- Maximalizujeme společenský přebytek, takže pravidlo alokace je:

$$x(b) = \arg \max_{\omega \in \Omega} \sum_{i=1}^n b_i(\omega). \quad (3)$$

- Platby p volíme tak, aby byly v souladu s předpokladem pravdivosti dražitelů.
- Klíčová myšlenka: Zohlednění ztráty společenského přebytku způsobené dražitelem i na ostatní $n - 1$ dražitelů.

42 VCG Mechanism: Proof II

Intuice: Tento slide formálně definuje pravidlo plateb a ukazuje, jak mechanismus maximalizuje společenský přebytek za předpokladu pravdivých nabídek. Platby zajišťují, že každý dražitel maximalizuje svůj užitek, pokud podává pravdivé nabídky.

- Formálně volíme pravidlo plateb jako:

$$p_i(b) = \max_{\omega \in \Omega} \sum_{j \neq i} b_j(\omega) - \sum_{j \neq i} b_j(\omega^*), \quad (4)$$

kde $\omega^* = x(b)$ je výsledek zvolený pravidlem alokace.

- První člen představuje přebytek zbývajících $n-1$ dražitelů, pokud vynecháme dražitele i .
- Druhý člen představuje společenský přebytek při započítání dražitele i .
- Mechanismus (x, p) maximalizuje společenský přebytek při předpokladu pravdivých nabídek.
- Zbývá dokázat, že je DSIC:
 - Každý dražitel i maximalizuje svůj užitek $v_i(x(b)) - p_i(b)$, pokud nastaví $b_i(\omega) = v_i(\omega)$ pro každé $\omega \in \Omega$.
 - Platí $0 \leq p_i(b) \leq b_i(\omega^*)$ (*cvičení*), což zajišťuje, že pravdomluvnost přináší nezáporný užitek.

43 VCG Mechanism: Proof III

Intuice: Tento slide zkoumá, jak dražitel i maximalizuje svůj užitek. Platby jsou definovány tak, že nejlepší strategií je pravdivě odhalit své hodnoty, protože mechanismus vybírá optimální výsledek maximalizující součet nabídek.

- Fixujeme dražitele i a nabídky ostatních dražitelů b_{-i} .
- Pokud $x(b) = \omega^*$, pak podle definice plateb platí:

$$v_i(\omega^*) - p_i(b) = \left(v_i(\omega^*) + \sum_{j \neq i} b_j(\omega^*) \right) - \left(\max_{\omega \in \Omega} \sum_{j \neq i} b_j(\omega) \right). \quad (5)$$

- Druhý člen je nezávislý na b_i , a proto dražitel i musí maximalizovat první člen, aby maximalizoval svůj užitek.
- Dražitel i nemůže přímo ovlivnit ω^* , protože mechanismus (x, p) vybírá ω^* , aby maximalizoval součet nabídek.
- Nejlepší případ pro dražitele i nastává, když mechanismus vybere ω^* , které maximalizuje:

$$\arg \max_{\omega \in \Omega} \left(v_i(\omega) + \sum_{j \neq i} b_j(\omega) \right). \quad (6)$$

- Pokud dražitel i podává pravdivé nabídky, souhlasí to s naším pravidlem alokace x , a tedy pravdivé podávání maximalizuje jeho užitek.

44 Myerson's Lemma

Intuice: Myersonova lemma poskytuje podmínky, za kterých je pravidlo alokace implementovatelné, a definuje odpovídající pravidlo plateb. Tato lemma je klíčová pro mechanism design v prostředí s jedním parametrem.

- Myersonova lemma (Věta 3.8):
 1. Pravidlo alokace je implementovatelné, právě když je monotonné.
 2. Pokud je pravidlo alokace x monotonné, pak existuje jedinečné pravidlo plateb p , které činí mechanismus (x, p) DSIC (předpokládáme, že $b_i = 0$ implikuje $p_i(b) = 0$).
 3. Pravidlo plateb p je dáné explicitním vzorcem:

$$p_i(b; b_{-i}) = \int_0^{b_i} z \cdot \frac{d}{dz} x_i(z; b_{-i}) dz, \quad (7)$$

pro každé $i \in \{1, \dots, n\}$.

- Tento výsledek jsme použili mnohokrát, ale dosud jsme jej nedokázali. Nyní se na to zaměříme.

45 Proof of Myerson's Lemma I

Intuice: Tento slide představuje první část důkazu Myersonovy lemmaty. Důkaz využívá vlastnosti DSIC (dominantně strategicky kompatibilního) mechanismu a ukazuje, že pravidlo alokace musí být monotonné.

- Nechť x je pravidlo alokace a p pravidlo plateb takové, že (x, p) je DSIC.
- Vlastnost DSIC říká, že pro každé z platí:

$$u_i(z; b_{-i}) = v_i \cdot x_i(z; b_{-i}) - p_i(z; b_{-i}) \leq v_i \cdot x_i(v_i; b_{-i}) - p_i(v_i; b_{-i}). \quad (8)$$

- Pro dvě možné nabídky y a z , kde $0 \leq y < z$, může mít dražitel i skutečnou hodnotu z a podat falešnou nabídku y . DSIC podmínka pak dává:

$$u_i(y; b_{-i}) = z \cdot x_i(y; b_{-i}) - p_i(y; b_{-i}) \leq z \cdot x_i(z; b_{-i}) - p_i(z; b_{-i}) = u_i(z; b_{-i}). \quad (9)$$

- Analogicky, pokud $v_i = y$ a $b_i = z$, pak (x, p) splňuje:

$$u_i(z; b_{-i}) = y \cdot x_i(z; b_{-i}) - p_i(z; b_{-i}) \leq y \cdot x_i(y; b_{-i}) - p_i(y; b_{-i}) = u_i(y; b_{-i}). \quad (10)$$

- Spojením těchto nerovností získáme tzv. "payment difference sandwich":

$$z(x_i(y; b_{-i}) - x_i(z; b_{-i})) \leq p_i(y; b_{-i}) - p_i(z; b_{-i}) \leq y(x_i(y; b_{-i}) - x_i(z; b_{-i})). \quad (11)$$

- Protože $0 \leq y < z$, dostáváme $x_i(y; b_{-i}) \leq x_i(z; b_{-i})$. Tedy pokud (x, p) je DSIC, pak x musí být monotonné.

46 Proof of Myerson's Lemma II

Intuice: Druhá část důkazu předpokládá, že x je monotónní, a zkoumá chování této funkce. Předpokládáme, že x je po částech konstantní a analyzujeme její "skoky" mezi intervaly.

- V této části důkazu předpokládáme, že pravidlo alokace x je monotónní.
- Fixujeme i a b_{-i} , takže uvažujeme x_i a p_i jako funkce proměnné z .
- Dále předpokládáme, že funkce x_i je po částech konstantní. Graf x_i tedy sestává z konečného počtu intervalů se "skoky" mezi nimi:

$\text{jump}(x_i, t)$ označuje velikost skoku funkce f v bodě t .

- Pokud fixujeme z v "payment difference sandwich" a necháme y blížit se k z zdola, obě strany nerovnosti se stanou nulovými, pokud v z není žádný skok funkce x_i .
- Pokud $\text{jump}(x_i, z) = h > 0$, pak obě strany mají tendenci k $z \cdot h$.

47 Proof of Myerson's Lemma III

Intuice: Třetí část důkazu ukazuje, že pravidlo plateb p lze odvodit přímo ze skoků monotónní funkce x .

- Pokud má být mechanismus (x, p) DSIC, musí platit následující podmínka pro každé z :

$$\text{jump}(p_i, z) = z \cdot \text{jump}(x_i, z). \quad (12)$$

- Kombinací této podmínky s počátečním stavem $p_i(0; b_{-i}) = 0$ dostáváme vzorec pro platební funkci p pro každého dražitele i a nabídky b_{-i} ostatních dražitelů:

$$p_i(b_i; b_{-i}) = \sum_{j=1}^{\ell} z_j \cdot \text{jump}(x_i(\cdot; b_{-i}), z_j), \quad (13)$$

kde z_1, \dots, z_ℓ jsou body zlomu alokační funkce $x_i(\cdot; b_{-i})$ v intervalu $[0, b_i]$.

- S dalšími faktami z kalkulu lze tento argument zobecnit na libovolné monotónní funkce x_i .

48 Proof of Myerson's Lemma IV

Intuice: Poslední část důkazu ukazuje, že pokud je x monotónní, pak mechanismus (x, p) je skutečně DSIC.

- Zbývá ukázat, že pokud je x monotónní, pak mechanismus (x, p) je skutečně DSIC.

- Tento argument platí obecně, ale prezentujeme jej pouze pro funkce po částech konstantní.
- Připomeňme si, že užitek $u_i(b_i; b_{-i}) = v_i \cdot x_i(b_i; b_{-i}) - p_i(b_i; b_{-i})$.
- Pomocí vyjádření plateb vidíme, že platba $p_i(b_i; b_{-i})$ odpovídá oblasti $[0, b_i] \times [0, x_i(b_i; b_{-i})]$ vlevo od křivky $x_i(\cdot; b_{-i})$.
- Z obrázku plyne, že je optimální, aby dražitel i nabídl $b_i = v_i$.

49 Proof of Myerson's Lemma V: By Picture

Intuice: Obrázky ilustrují tři různé scénáře podávání nabídek dražitele i (pravdivé, přehnané a podhodnocené). Ukazují, že pravdivé podání maximalizuje užitek dražitele díky monotónnímu pravidlu alokace x a platebnímu pravidlu p .

- **Pravdivé podání (řádek a):** Pokud dražitel i podává $b_i = v_i$, jeho užitek je maximální. Modrá oblast odpovídá oblasti $[0, v_i] \times [0, x_i(v_i; b_{-i})]$, která určuje užitek.
- **Přehnané podání (řádek b):** Pokud dražitel podá $b_i > v_i$, červená oblast představuje ztrátu užitku kvůli vyšší platbě, zatímco zelená oblast není přínosná, protože nezvyšuje užitek.
- **Podhodnocené podání (řádek c):** Pokud dražitel podá $b_i < v_i$, modrá oblast se zmenší, což vede k nižšímu užitku. Červená oblast ukazuje ztrátu užitku kvůli nižšímu podílu na alokaci.

50 Summary of Myerson's Lemma Proof

Shrnutí: Myersonova lemma ukazuje, že DSIC mechanismy vyžadují monotónní pravidla alokace x a odpovídající pravidla plateb p . Důkaz se skládá ze čtyř kroků:

1. **Monotónnost:** DSIC podmínky implikují, že x musí být monotónní.
2. **Analýza skoků:** Monotoní x lze analyzovat prostřednictvím skoků, které ovlivňují platby.
3. **Platby:** Funkce plateb p je odvozena pomocí skoků v x a zajišťuje DSIC vlastnosti.
4. **Optimalita:** Pravdivé podání nabídek maximalizuje užitek dražitelů díky monotónnosti x a konstrukci p .

51 Going Beyond DSIC?

Intuice: Tento slide se zabývá otázkou, zda DSIC mechanismy mohou být příliš omezující. Zkoumá se rozdělení vlastností DSIC na dvě části a ukazuje se příklad, kdy mechanismus splňuje pouze jednu z nich.

- Dospud jsme se zaměřovali pouze na DSIC mechanismy, protože jsou snadné na hraní a předpověď.
- Otázka: Ztrácíme něco tím, že se omezujeme pouze na DSIC mechanismy?
- DSIC vlastnost lze rozdělit na dvě části:
 - Každý dražitel má dominantní strategii, bez ohledu na svou soukromou hodnotu.
 - Tato dominantní strategie je přímé odhalení (pravdivé podání).
- **Příklad (dražba, která splňuje pouze první vlastnost):**
 - Uvažujme dražbu jedné položky, kde pro nabídky (b_1, \dots, b_n) prodávající spustí Vickreyovu dražbu na $(2b_1, \dots, 2b_n)$.
 - Každý dražitel má dominantní strategii a tím je splněna první vlastnost.
 - Tato dominantní strategie však není přímé odhalení, ale nabízení poloviny své hodnoty.
- Otázka: Je DSIC příliš omezující?

52 Revelation Principle

Intuice: Tento slide vysvětluje princip odhalení, který říká, že pokud existuje mechanismus s dominantní strategií, můžeme jej převést na mechanismus s přímým odhalením, aniž by ztratil své vlastnosti.

- Odpověď na předchozí otázkou: **Ne!** Ukážeme, že pouze první podmínka je důležitá, zatímco druhá vyplývá zdarma.
- První podmínka není vždy splněna (např. first-price auctions), což ztěžuje předpověď chování dražitelů.
- Princip odhalení říká, že není třeba uvolňovat druhou podmínku.
- **Princip odhalení (Věta 3.19):** Pro každý mechanismus M s více parametry, ve kterém má každý dražitel dominantní strategii bez ohledu na svou soukromou hodnotu, existuje ekvivalentní mechanismus M' s dominantní strategií, která je přímým odhalením.

53 Going Beyond DSIC?

Intuice: Tento slide se zabývá otázkou, zda DSIC mechanismy mohou být příliš omezující. Zkoumá se rozdělení vlastností DSIC na dvě části a ukazuje se příklad, kdy mechanismus splňuje pouze jednu z nich.

- Dospud jsme se zaměřovali pouze na DSIC mechanismy, protože jsou snadné na hraní a předpověď.
- Otázka: Ztrácíme něco tím, že se omezujeme pouze na DSIC mechanismy?
- DSIC vlastnost lze rozdělit na dvě části:
 - Každý dražitel má dominantní strategii, bez ohledu na svou soukromou hodnotu.
 - Tato dominantní strategie je přímé odhalení (pravdivé podání).
- **Příklad (dražba, která splňuje pouze první vlastnost):**
 - Uvažujme dražbu jedné položky, kde pro nabídky (b_1, \dots, b_n) prodávající spustí Vickreyovu dražbu na $(2b_1, \dots, 2b_n)$.
 - Každý dražitel má dominantní strategii a tím je splněna první vlastnost.
 - Tato dominantní strategie však není přímé odhalení, ale nabízení poloviny své hodnoty.
- Otázka: Je DSIC příliš omezující?

54 Revelation Principle

Intuice: Tento slide vysvětluje princip odhalení, který říká, že pokud existuje mechanismus s dominantní strategií, můžeme jej převést na mechanismus s přímým odhalením, aniž by ztratil své vlastnosti.

- Odpověď na předchozí otázku: **Ne!** Ukážeme, že pouze první podmínka je důležitá, zatímco druhá vyplývá zdarma.
- První podmínka není vždy splněna (např. first-price auctions), což ztěžuje předpověď chování dražitelů.
- Princip odhalení říká, že není třeba uvolňovat druhou podmínku.
- **Princip odhalení (Věta 3.19):** Pro každý mechanismus M s více parametry, ve kterém má každý dražitel dominantní strategii bez ohledu na svou soukromou hodnotu, existuje ekvivalentní mechanismus M' s dominantní strategií, která je přímým odhalením.

55 Proof of the Revelation Principle

Intuice: Tento slide popisuje důkaz principu odhalení pomocí konstrukce ekvivalentního mechanismu M' , který využívá přímé odhalení. Ukazuje, že přímé odhalení je dominantní strategií v M' .

- Důkaz postupuje pomocí simulačního argumentu.
- Pro každého dražitele i a jeho ohodnocení $v_i(\omega)_{\omega \in \Omega}$ nechť $s_i(v_i(\omega)_{\omega \in \Omega})$ je dominantní strategie dražitele i v mechanismu M .
- Konstrukce mechanismu M' :
 - Mechanismus M' přijímá zapečetěné nabídky $b_1(\omega)_{\omega \in \Omega}, \dots, b_n(\omega)_{\omega \in \Omega}$ od dražitelů.
 - Poté M' podá nabídky $s_1(b_1(\omega)_{\omega \in \Omega}), \dots, s_n(b_n(\omega)_{\omega \in \Omega})$ do M a M' vrátí stejný výsledek jako M .
- Přímé odhalení je dominantní strategií v M' , protože pokud má dražitel i ohodnocení $v_i(\omega)_{\omega \in \Omega}$, podání jiné nabídky než $v_i(\omega)_{\omega \in \Omega}$ povede pouze k použití jiné strategie než $s_i(v_i(\omega)_{\omega \in \Omega})$ v M' , což může pouze snížit užitek dražitele i .