

Міністерство освіти і науки України
Національний університет «Києво-Могилянська академія»
Факультет інформатики
Кафедра математики

Курсова робота
на тему: «Багатокрокові антагоністичні стохастичні ігри»

Виконала: студентка 3-го року навчання
спеціальності 113 "Прикладна математика"

Метелюк Софія Володимирівна

Керівник курсової роботи:
к. ф.-м. н. Чорней Р.К.

Київ - 2021

Тема: Багатокрокові антагоністичні стохастичні ігри

Календарний план виконання роботи:

№ п/п	Назва етапу курсової роботи	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу.	15.10.2020	
2.	Огляд літератури за темою роботи.	20.11.2020	
3.	Дослідження результатів отриманих в літературі	25.11.2020	
4.	Аналіз отриманих результатів з керівником	30.12.2020	
5.	Оформлення курсової роботи	10.05.2021	
6.	Створення презентації до курсової роботи та програмної реалізації	17.05.2021	
12.	Захист курсової роботи	20.05.2021	

Студент _____

Керівник _____

“ _____ ” _____

ЗМІСТ

на тему: «Багатокрокові антагоністичні стохастичні ігри»	0
ЗМІСТ	3
Вступ	4
Анотація	5
1.Теоретична інформація	6
1.1. Формалізація та визначення гри	6
1.2. Опис гри для двох гравців	7
1.3. Два критерія якості керування	8
2. Ітераційний метод Ховарда для рішення Марківських процесів із доходами	11
2.1 Ітераційний цикл	11
2.1.1 Визначення вагів	11
2.1.2 Покращення результату	12
2.1.3 Умова зупинки	12
2.2 Приклад	13
2.3. Програмна реалізація	19
2.3.1 Опис роботи програми.....	20
ВИСНОВКИ	21
СПИСОК ЛІТЕРАТУРИ	22
Додаток А	23

Вступ

Антагоністичні багатокрокові стохастичні ігри можуть використовуватися для моделювання таких ситуацій як конкурентна боротьба в бізнесі. Мета отримати оптимальну стратегію для гравців у грі за допомогою ітераційного методу знаходження оптимальної стратегії для Марківського процесу з доходами та реалізувати програму, що виконує ітераційний цикл.

Анотація

В цій роботі міститься загальні відомості про багатокрокові антогоністичні стохастичні ігри, розглядається приклад гри двох гравців, а також розв'язується за допомогою ітераційного методу знаходження оптимальної стратегії для Марківського процесу з доходами. Зроблена коротка реалізація основного циклу ітераційного методу.

1. Теоретична інформація

1.1. Формалізація та визначення гри

Стратегія це опис послідовності дій.

Ігри із нульовою сумою часто називають антагоністичними іграми, адже виграш одного означає програш іншого(інших).

В цій роботі мова йде про некооперативні ігри, тобто гравці не мають можливості домовлятися та діяти спільно. Відсутня ситуація створення коаліцій. Фактор змови виключений. Виграш одного відбувається за програшу іншого.

Вибір дії здійснюється одночасно.

$X_1, X_2, \dots$ – стани першого гравця

$Y_1, Y_2, \dots$ – стани другого гравця

Набір умовних ймовірностей

$$P(X_{t+1}|X_t, Y_t, S^{(1)})$$

$$P(Y_{t+1}|X_t, Y_t, S^{(2)})$$

S -це стратегія, згідно з якими вибирається рішення

Процес гри:

$$(X, Y) \rightarrow (S^{(1)}, S^{(2)}) (* \text{ обираємо рішення }) \rightarrow (X_{t+1}, Y_{t+1}) \rightarrow \dots$$

Функція витрат/нагород

$$U(X_t, Y_t, S^{(1)}, S^{(2)})$$

Але ми вважаємо, що немає витрат на прийняття рішення, тому $S^{(1)} = S^{(2)} = 0$, того

$$U(X_t, Y_t)$$

Математичне Сподівання -

$$E^{S^{(1)}, S^{(2)}} \sum_{t=1}^{\infty} U(X_t, Y_t) = \varphi(S^{(1)}, S^{(2)})$$

Функціонал може бути розбіжною

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{t=1}^n U(X_t, Y_t)$$

Функція виплат

$$|U(X_t, Y_t)| < L < \infty$$

Сідлова точка гри

$$S^{(1)*}, S^{(2)*}$$

Необхідно знайти умови існування сідлової точки і ціни гри за таких умов

$$\inf_{S^{(2)}} \sup_{S^{(1)}} \varphi(S^{(1)}, S^{(2)}) = \sup_{S^{(1)}} \inf_{S^{(2)}} \varphi(S^{(1)}, S^{(2)})$$

1.2. Опис гри для двох гравців

Матеріал цього розділу має реферативний характер. Використовуються теоретичні положення з роботи [1].

В момент часу $n = 0, 1, \dots$ два гравці P_1 та P_2 обирають рішення -керування. Рішення відповідно будемо позначати $a_n \in A$ та $b_n \in B$. Рішення приймаються на основі повної інформації про стан системи $x_n \in X$ і всіх попередніх станів і рішень.

Після цього гравець P_2 платить гравцю P_1 суму $U(x_n, a_n, b_n)$, а система переходить в інший стан.

$$P\{x_{n+1} \in C/x_0, a_0, b_0, \dots, x_n = x, a_n = a, b_n = b\} = P(C/x, a, b)$$

Функція $U(x_n, a_n, b_n)$ та перехідна ймовірність $P(\cdot/x, a, b)$

$$|U(x_n, a_n, b_n)| < L < \infty, x \in X, a \in A, b \in B$$

Сукупність X, A, B, P, U визначає стохастичну гру.

Стратегія π гравця P_1 визначається як послідовність $\{\pi_0, \pi_1, \pi_2, \dots\}$, де $\pi_n(\cdot/x_0, a_0, b_n, \dots, x_{n-1}, a_{n-1}, b_{n-1}, x_n)$ – ймовірнісна міра на $(A, \mathcal{U})$, вимірювання по сукупності змінних. Стратегія π гравця P_1 називається напівмарківською, якщо $\pi_n = \pi_n(\cdot/x_0, x_n), n = 0, 1, 2, \dots$,

і стаціонарною, якщо існує борелівська функція $f: X \rightarrow P_A$ така, що $\pi_n = f, n = 0, 1, 2, \dots$

Де P_A -множина всіх ймовірнісних мір на $(A, \mathcal{U})$ з топологією слабкою збіжності.

В подальшому стаціонарні стратегії і відповідні їм борелівські функції ототожнюються. Стратегії гравця P_2 визначаються аналогічно.

1.3. Два критерія якості керування

1. ψ – критерій

2. φ – критерій

$$\varphi(x, S^{(1)}, S^{(2)}) = \lim_{n \rightarrow \infty} \inf \frac{1}{n+1} E_{S^{(2)}}^{S^{(1)}} \sum_{k=0}^n U(x_k, a_k, b_k)$$

де $E_{S^{(2)}}^{S^{(1)}}$ - це математичне сподівання, що відповідає процесу, керованому стратегіями $S^{(1)}, S^{(2)}$.

Стратегія $S^{(1)}$ * називається φ – оптимальною для гравця А, якщо

$$\varphi(x, S^{(1)} *, S^{(2)'}) \geq \inf_{S^{(2)}} \sup_{S^{(1)}} \varphi(x, S^{(1)}, S^{(2)})$$

для всіх $S^{(2)'} \text{ і } x \in X$.

Стратегія $S^{(2)*}$ називається φ – оптимальною для гравця В, якщо

$$\varphi(x, S^{(1)'}, S^{(2)*}) \leq \sup_{S^{(1)}} \inf_{S^{(2)}} \varphi(x, S^{(1)}, S^{(2)})$$

для всіх $S^{(1)'} \text{ і } x \in X$.

Стохастична гра має ціну, якщо

$$\inf_{S^{(2)}} \sup_{S^{(1)}} \varphi(S^{(1)}, S^{(2)}) = \sup_{S^{(1)}} \inf_{S^{(2)}} \varphi(S^{(1)}, S^{(2)})$$

Визначимо редукційовану модель (R-модель)

$$U(x, \mu, \lambda) = \iint U(x, a, b) \mu(da) \lambda(db)$$

$$P(\cdot/x, \mu, \lambda) = \iint P(\cdot/x, a, b) \mu(da) \lambda(db)$$

Нехай g - це фіксована стаціонарна стратегія гравця В.

R-моделлю - називається керована марківська модель з дискретним часом, в якому фазовим простором є X , простором рішень P_A , функція доходу U та перехідною ймовірністю P визначені наступні відношення

$$U(x, \mu) = U(x, \mu, g(x))$$

$$P'(C/x, \mu) = P(C/x, \mu, g(x))$$

1.4. Стратегії

Бувають рандомізовані і нерандомізовані:

Стратегія нерандомізована, якщо

$$S^{(1)} \sim F^{(1)}(X_t, Y_t)$$

Стратегія рандомізована, якщо

$$S^{(1)} \sim F^{(1)}(\cdot | X_t, Y_t)$$

Лема 1. Якщо π - напівмарківська (стаціонарна) стратегія гравця P_1 в стохастичній грі, то π є нерандомізованою напівмарківською (стаціонарною) стратегією в R -моделі і $\varphi'(x, \pi) = \varphi(x, \pi, g), x \in X$.

І навпаки, якщо π нерандомізована напівмарківська (стаціонарна) стратегія в R -моделі, то π є напівмарківською (стаціонарною) стратегією гравця P_1 в стохастичній грі і $(x, \pi, g) = \varphi'(x, \pi), x \in X$

Лема 2. Якщо π — стратегія гравця P_1 в стохастичній грі, то існує напівмарківська стратегія π^* игрока P_1 така, що $\varphi(x, \pi^*, g) = \varphi(x, \pi, g)$

Теорема 1 ([1])

Нехай існує стаціонарні стратегії f гравця A і g гравця B , константа c і обмежена борелівська функція $v(x)$ на X , такі що:

$$c + v(x) = \sup_{\mu \in P_A} \left\{ U(x, \mu, g(x)) + \iint v(y) P(dy/x, \mu, g(x)) \right\}$$

$$c + v(x) = \inf_{\lambda \in P_B} \left\{ U(x, f(x), \lambda) + \iint v(y) P(dy/x, f(x), \lambda) \right\}, \quad x \in X$$

Тоді стохастична гра має φ — ціну, **яка тотожно рівна c** , і стратегії f, g є φ — оптимальні для гравців A та B відповідно.

$$\text{Тобто } \inf_{S^{(2)}} \sup_{S^{(1)}} \varphi(S^{(1)}, S^{(2)}) = \sup_{S^{(1)}} \inf_{S^{(2)}} \varphi(S^{(1)}, S^{(2)}) = c$$

2. Ітераційний метод Ховарда для рішення Марківських процесів із доходами

Матеріал цього розділу має реферативний характер. Використовуються теоретичні положення з роботи [2].

2.1 Ітераційний цикл

Позначення:

$v_i(n)$ – повний очікуваний дохід після n переходів, якщо першим був стан i .

p_{ij} – умовна ймовірність, що система, що зараз в стані i , а після наступного переходу опиниться в стані j .

q_i^k – очікуваний дохід за один перехід при виході із стану

$d_i(n)$ – номер стратегії, обраної в стані i , яка буде використовуватися на n -му кроці. Назвемо $d_i(n)$ рішенням в стані i на n -му кроці.

Ітераційний цикл складається із двох блоків: визначення вагів та покращення результату. Проте не важливо, з якого блоку починати роботу.

2.1.1 Визначення вагів

Використовуючи p_{ij} та q_i для даного рішення знайти прибуток q і відносні ваги v_i із системи рівнянь

$$g + v_i = q_i + \sum_{j=1}^N p_{ij} v_j, i = 1, 2, \dots, N$$

Поклавши v_N рівним нулю.

2.1.2 Покращення результату

Для кожного стану i , використовуючи відносні ваги із попереднього рішення, знайти стратегію k' , яка максимізує критерій $q_i^k + \sum_{j=1}^N p_{ij}^k v_j$

Потім прийняти цю стратегію за нове рішення в i -м стані, q_i^k змінити на $q_i^{k'}$, а p_{ij}^k на $p_{ij}^{k'}$.

$$q_i^k = \sum_{j=1}^N p_{ij}^k r_{ij}^k$$

Поведінка гравця є визначеною, якщо для всіх i та n задано $d_i(n)$, тобто задано *рішення*. Оптимальним є така поведінка, яка максимізує повний очікуваний дохід для всіх i та n .

Щоб проаналізувати задачу знаходження оптимальної поведінки, визначимо знову $v_i(n)$ як повний очікуваний за n кроків дохід при оптимальній поведінці, якщо система відправляється зі стану i .

Таким чином, для будь-якого n маємо

$$v_i(n+1) = \max_k \sum_{j=1}^N p_{ij}^k [r_{ij}^k + v_j(n)], n = 0, 1, 2, \dots$$

2.1.3 Умова зупинки

Оптимальне рішення (g максимізовано) буде знайдено, коли рішення двох послідовних ітерацій збігатимуться.

Щоб уникнути можливого повторення процедури покращення рішення при однаково хороших стратегіях в деякому стані, достатньо вимагати, щоб старе рішення d_i залишалося незмінним, якщо величина критерія для нього така ж, як і для всякої іншої стратегії знову визначеного рішення.

Властивості ітераційного циклу:

- 1) Визначення оптимального рішення в процесі послідовних рішень зводиться до вирішення системи лінійних рівнянь з подальшим порівнянням.
- 2) Ітераційний цикл буде закінчено при отриманні рішення, яке забезпечить найбільший можливий дохід у цій задачі.

2.2 Приклад

Антогоністична стохастична гра двох гравців А та В.

Два стани s_1 та s_2 .

Обидва гравці мають дві дії в стані s_1 .

Для гравця А:

- 1) Обрати рядок 1;
- 2) Обрати рядок 2;

Для гравця В:

- 1) Обрати колонку 1;
- 2) Обрати колонку 2;

В стані s_2 , гравець А має три дії, а гравець В має дві дії.

Для гравця А:

- 1) Обрати рядок 1;
- 2) Обрати рядок 2;
- 3) Обрати рядок 3;

Для гравця В:

1) Обрати колонку 1;

2) Обрати колонку 2;

Гравці обирають дії, гра згідно із перехідними ймовірностями переходить у інший стан і тоді гравці виплачують\отримують нагороду.

Функція виплат при переході й перехідні ймовірності записані нижче.

Виплати записуємо, як від гравця А, тобто якщо із плюсом, то отримує він, якщо із мінусом, то інший гравець.

В дужках перехідні ймовірності, а над ними винагороду, що платить у випадку переходу за цією ймовірністю у відповідний стан.

$$s_1 \begin{bmatrix} 1 & -1 & -2 & 2 \\ \left(\frac{1}{4}, \frac{3}{4}\right) & (0,1) & & \\ -1 & 1 & 2 & -2 \\ \left(\frac{1}{4}, \frac{3}{4}\right) & (0,1) & & \end{bmatrix}$$

$$s_2 \begin{bmatrix} -5 & 5 & 5 & -5 \\ \left(\frac{1}{5}, \frac{4}{5}\right) & \left(\frac{1}{5}, \frac{4}{5}\right) & & \\ 1 & -1 & -1 & 1 \\ (1,0) & (1,0) & & \\ 2 & -2 & -2 & 2 \\ (0,1) & (0,1) & & \end{bmatrix}$$

Приклад, якщо в стані s_1 гравці обирають рядок 1, колонку 1, то

і із ймовірністю $\frac{1}{4}$ гра залишається в стані s_1 і гравець А отримує 1 від гравця В або із ймовірністю $\frac{3}{4}$ переходить в стан s_2 і тоді вже гравець А платить 1 гравцю В, тобто у нього -1.

Таблиця 1 – Запис гри у таблиці

Стани i	Стратегія k, m	Перехідна ймовірність p_{ij}^k $j = 1 \quad j = 2$	Дохід r_{ij}^k $j = 1 \quad j = 2$	Безпосередньо очікуваний дохід $q_i^k = \sum_{j=1}^N p_{ij}^k r_{ij}^k$
$i = 1$	$1 \quad k = 1, m = 1$ $2 \quad k = 1, m = 2$ $3 \quad k = 2, m = 1$ $4 \quad k = 2, m = 2$	$\begin{bmatrix} 1/4 & 3/4 \\ 0 & 1 \\ 1/4 & 3/4 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & -1 \\ -2 & 2 \\ -1 & 1 \\ 2 & -2 \end{bmatrix}$	$-1/2$ 2 $1/2$ -2
$i = 2$	$k = 1, m = 1$ $k = 1, m = 2$ $k = 2, m = 1$ $k = 2, m = 2$ $k = 3, m = 1$ $k = 3, m = 2$	$\begin{bmatrix} 1/5 & 4/5 \\ 1/5 & 4/5 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -5 & 5 \\ 5 & -5 \\ 1 & -1 \\ -1 & 1 \\ 2 & -2 \\ -2 & 2 \end{bmatrix}$	3 -3 1 -1 -2 2

Починаємо розв'язок із покладання $v_2 = 0$.

$$d = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$P = \begin{bmatrix} 1/4 & 3/4 \\ 1/5 & 4/5 \end{bmatrix}$$

$$q = \begin{bmatrix} -1/2 \\ 3 \end{bmatrix}$$

Тепер переходимо до процедури визначення вагів, вирішуючи наступне рівняння

$$g + v_i = g_i + \sum_{j=1}^N p_{ij}^k v_j^k, \quad i = 1, 2, \dots, N$$

Що у нашому випадку буде записано так

$$g + v_1 = -\frac{1}{2} + \frac{1}{4}v_1 + \frac{3}{4}v_2$$

$$g + v_2 = 3 + \frac{1}{5}v_1 + \frac{4}{5}v_2$$

Якщо $v_2 = 0$, то $g = \frac{43}{19}$, $v_1 = \frac{70}{19}$

Тепер повертаємося до процедури покращення рішення, обчислимо величину

$$q_i^k + \sum_{j=1}^N p_{ij}^k v_j^k, \text{ для всіх } i \text{ та } k$$

Таблиця 2- Другий запис гри з оновленими даними

Стани i	Стратегія k, m	Перехідна ймовірність p_{ij}^k $j = 1 \quad j = 2$	Безпосередньо очікуваний дохід $q_i^k = \sum_{j=1}^N p_{ij}^k r_{ij}^k$	$q_i^k + \sum_{j=1}^N p_{ij}^k v_j^k$
$i = 1$	$k = 1, m = 1$	$\begin{bmatrix} 1/4 & 3/4 \\ 0 & 1 \\ 1/4 & 3/4 \\ 0 & 1 \end{bmatrix}$	$-1/2$	1,09
	$k = 1, m = 2$		2	2
	$k = 2, m = 1$		$1/2$	2,09 *
	$k = 2, m = 2$		-2	-2

$i = 2$	$k = 1, m = 1$	$\begin{bmatrix} 1/5 & 4/5 \\ 1/5 & 4/5 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$	3	4,27
	$k = 1, m = 2$		-3	-1,73
	$k = 2, m = 1$		1	7,36 *
	$k = 2, m = 2$		-1	5,36
	$k = 3, m = 1$		-2	-2
	$k = 3, m = 2$		2	2

Символом * позначено максимальні значення

Тобто при $i = 1$, значення максимальне при $k = 2, m = 1$

Для $i = 2$ при $k = 2, m = 1$

$$d = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

Якщо $d = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$, то

$$P = \begin{bmatrix} 1/4 & 3/4 \\ 1 & 0 \end{bmatrix}$$

$$q = \begin{bmatrix} 1/2 \\ 1 \end{bmatrix}$$

Знову переходимо до процедури визначення вагів

$$g + v_1 = \frac{1}{2} + \frac{1}{4}v_1 + \frac{3}{4}v_2$$

$$g + v_2 = 1 + 1v_1 + 0v_2$$

Якщо $v_2 = 0$, то $g = \frac{5}{7}$, $v_1 = -\frac{2}{7}$

Тепер знову переходимо до процедури покращення рішення, обчислимо величину

$$q_i^k + \sum_{j=1}^N p_{ij}^k v_j^k, \text{ для всіх } i \text{ та } k$$

Таблиця 2- Третій запис гри з оновленими даними

Стани i	Стратегія k, m	Перехідна ймовірність p_{ij}^k $j = 1 \quad j = 2$	Безпосередньо очікуваний дохід $q_i^k = \sum_{j=1}^N p_{ij}^k r_{ij}^k$	$q_i^k + \sum_{j=1}^N p_{ij}^k v_j^k$
$i = 1$	$k = 1, m = 1$ $k = 1, m = 2$ $k = 2, m = 1$ $k = 2, m = 2$	$\begin{bmatrix} 1/4 & 3/4 \\ 0 & 1 \\ 1/4 & 3/4 \\ 0 & 1 \end{bmatrix}$	$-1/2$ 2 $1/2$ -2	$-0,57$ $2 *$ $0,43$ -2
$i = 2$	$k = 1, m = 1$ $k = 1, m = 2$ $k = 2, m = 1$ $k = 2, m = 2$ $k = 3, m = 1$ $k = 3, m = 2$	$\begin{bmatrix} 1/5 & 4/5 \\ 1/5 & 4/5 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$	3 -3 1 -1 -2 2	$2,94 *$ $-3,06$ $0,71$ $-1,29$ -2 2

Тобто при $i = 1$, значення максимальне при $k = 1, m = 2$

Для $i = 2$ при $k = 1, m = 1$

Якщо $d = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

$$P = \begin{bmatrix} 0 & 1 \\ 1/5 & 4/5 \end{bmatrix}$$

$$q = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

Знову переходимо до процедури визначення вагів.

$$g + v_1 = 2 + 0v_1 + 1v_2$$

$$g + v_2 = 3 + \frac{1}{5}v_1 + \frac{4}{5}v_2$$

Якщо $v_2 = 0$, то $g = \frac{17}{6}$, $v_1 = -\frac{5}{6}$

Таблиця 2- Четвертий запис гри з оновленими даними

Стани i	Стратегія k, m	Перехідна ймовірність p_{ij}^k $j = 1 \quad j = 2$	Безпосередньо очікуваний дохід $q_i^k = \sum_{j=1}^N p_{ij}^k r_{ij}^k$	$q_i^k + \sum_{j=1}^N p_{ij}^k v_j^k$
$i = 1$	$k = 1, m = 1$ $k = 1, m = 2$ $k = 2, m = 1$ $k = 2, m = 2$	$\begin{bmatrix} 1/4 & 3/4 \\ 0 & 1 \\ 1/4 & 3/4 \\ 0 & 1 \end{bmatrix}$	$-1/2$ 2 $1/2$ -2	$-0,71$ $2 *$ $0,29$ -2
$i = 2$	$k = 1, m = 1$ $k = 1, m = 2$ $k = 2, m = 1$ $k = 2, m = 2$ $k = 3, m = 1$ $k = 3, m = 2$	$\begin{bmatrix} 1/5 & 4/5 \\ 1/5 & 4/5 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$	3 -3 1 -1 -2 2	$2,83 *$ $-3,17$ $0,17$ $-1,83$ -2 2

Отже, $d = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ є оптимальною стратегією для гравця А.

2.3. Програмна реалізація

2.3.1 Опис роботи програми

У роботі використовується пакет numpy.

Спочатку задаються матриці :

$p0$ – це масив, що передає матрицю переходів для стану 1

$p1$ -це масив, що передає матрицю переходів для стану 2

$r0$ - це масив, що передає матрицю доходів в стані 1

$r1$ - це масив, що передає матрицю доходів в стані 2

Потім обчислюється за формулою безпосередньо очікуваний дохід

$$q_i^k = \sum_{j=1}^N p_{ij}^k r_{ij}^k$$

Отримується масив `q_array1`, де зберігаються дані q_i^k

Тоді формується дві матриці A та b і за допомогою функції `np.linalg.solve`, що вирішує систему лінійних рівнянь і отримується значення v_1 .

Тепер переходимо до процесу покращення рішення.

Обчислюється значення за формулою

$$q_i^k + \sum_{j=1}^N p_{ij}^k v_j^k$$

І отримані значення записуються в масив `improves`. Тоді за допомогою методу `np.argmax` знаходиться номер стратегії, яка варто обрати у відповідному стані.

Проте варто враховувати, що у мові програмування Python нумерація масиву розпочинається із 0, тож якщо наше отримане значення n , то це насправді стратегія $n + 1$.

ВИСНОВКИ

У процесі роботи над курсовою було розглянуто термінологію та основні положення необхідні для роботи із багатокроковими антагоністичними стохастичними іграми, Марківськими процесами із доходами.

Зроблено реалізацію основного циклу ітераційного методу. У майбутніх дослідженнях можна розширити гру на більшу кількість гравців, а також обчислювати всі оптимальні стратегії гравця, якщо їх є декілька.

СПИСОК ЛИТЕРАТУРЫ

- [1] Губенко Л. Г. О многошаговых стохастических играх / Л. Г. Губенко. // Теория вероятностей и математическая статистика. – 1973. – №8. – С. 35–49.
- [2] Ховард Р. А. Динамическое программирование и Марковские процессы / Р. А. Ховард. – Москва: Советское радио, 1964. – 195 с.

Додаток А

(обов'язковий)

```

p0 = np.array([[0.25,0.75],
               [0.00,1.00],
               [0.25,0.75],
               [0.00,1.00]],float)

p1 = np.array([[0.20,0.80],
               [0.20,0.80],
               [1.00,0.00],
               [1.00,0.00],
               [0.00,1.00],
               [0.00,1.00]], float)

r0 =np.array([[1.00,-1.00],
              [-2.00,2.00],
              [-1.00,1.00],
              [2.00,-2.00]], float)

r1 =np.array([[5.00,-5.00],
              [-5.00,5.00],
              [1.00,-1.00],
              [-1.00,1.00],
              [2.00,-2.00],
              [-2.00,2.00]], float)

d_global= np.array([ ], float)
q_array1= np.zeros(4)
np.append(q_array1,1)

for i in range(4):
    for j in range(1):
        q_array1[i]=p0[i][j]*r0[i][j]+p0[i][j+1]*r0[i][j+1]

print(q_array1)

d=np.array([1,1])
P1=np.array([[0.25,0.75],
             [0.20,0.80]],float)
Q1=np.array([-0.5,3.00],float)

A = np.array([[1,1-P1[0][0]], [1,-P1[1][0]]])
b=np.array([-0.50, 3])
n=b.shape[0]

```

```

v_1=np.linalg.solve(A, b)
print(v_1)

imprvres = np.zeros(4)
for idx, val in enumerate(q_array1):
    imprvres[idx] = val + p0[idx, 0] * v_1[1]

np.argmax(imprvres)

q_array2= np.zeros(6)
for i in range(6):
    for j in range(1):
        q_array2[i]=p1[i][j]*r1[i][j]+p1[i][j+1]*r1[i][j+1]

print(q_array2)

imprvres2 = np.zeros(6)
for idx, val in enumerate(q_array2):
    imprvres2[idx] = val + p1[idx, 0] * v_1[1]

np.argmax(imprvres2)

```