Київський національний університет

імені Тараса Шевченка

Факультет комп’ютерних наук та кібернетики

Кафедра обчислювальної математики

Кваліфікаційна робота

на здобуття ступеня бакалавра

за спеціальністю 113 Прикладна математика

на тему:

Розподілені алгоритми пошуку рівноваги Неша

Виконала студентка 4-го курсу
Кушнір Катерина Миколаївна ________________

Науковий керівник:
доктор фіз.-мат. наук, професор
Семенов Володимир Вікторович ________________

Київ – 2021

Зміст

Вступ

В даній роботі ми розглянемо декілька алгоритмів по пошуку рівноваги Неша. Це одне із ключових понять в теорії ігор.

Теорія ігор – це розділ математики, в якому вивчаються математичні моделі прийняття рішень, в умовах конфлікту, тобто в умовах зіткнення сторін, кожна з яких намагається впливати на розвиток конфлікту в своїх власних інтересах. Теорію математичних моделей прийняття оптимальних рішень прийнято називати дослідженням операцій, тому теорію ігор слід розглядати як прикладну математичну теорію – складну частину досліджень операцій.[1]

Ігри можна класифікувати за різними ознаками. По-перше, безкоаліційні ігри, в яких кожна коаліція (безліч гравців, що діють спільно), є лише одним гравцем. Так названа кооперативна теорія безкоаліційних ігор допускає тимчасові об'єднання гравців у коаліції в процесі ігри з подальшим розділенням отриманого вигравання або прийняття спільних рішень. По-друге, коаліційні ігри, в яких приймають рішення гравці згідно з правилами гри об'єднані в фіксовані коаліції. Члени однієї коаліції можуть вільно обмінюватися інформацією і приймати повністю узгоджені рішення. За виграшом, ігри можна розділити на антагоністичні та ігри з ненульовою сумою. За характером отримання інформації - на ігри в нормальній формі (гравці отримують всю призначену їм інформацію до початку гри) і динамічні ігри(Інформація надходить гравцям в процесі розвитку гри). За кількістю стратегій - на кінцеві і нескінченні гри.

Рівновага Неша – це концепція рішення, набір стратегій у грі для двох і більше гравців, в якому ні один із гравців не може збільшити виграш, змінивши свою стратегію, якщо інші гравці стратегій не змінюють.[2]

Дана концепція вперше була використана Антуаном Огюстом Курно. Він показав, як можна знайти те, що ми звемо рівновагою Неша, в грі Курно. Джо Неш першим довів існування такої рівноваги для всіх кінцевих ігор, з будь-якою кількістю гравців. Це було зроблено в його дисертації по некооперативним іграм у 1950 році. До Неша дане твердження було доведено тільки для ігор з 2 учасниками з нульовою сумою Джоном фон Нейманом та Оскаром Моргенштерном у 1947.

Між проблемою пошуку рівноваги Неша (NE) розподіленої гри та розподілена задача оптимізації існує тісний зв’язок. У розподіленій задачі оптимізації з N агентами, які обмінюються даними через зв'язний граф, бажано мінімізувати глобальну мету:

$$\begin{cases} \underset{x}{\text{minimize}}\, f(x) \coloneqq \sum_{i=1}^{N} f_i(x) \\ \qquad\qquad x \in \Omega \end{cases} \quad (1)$$

У цьому завданні агенти спільно вирішують (1) над загальною оптимізаційною змінною x. Іншими словами, всі агенти служать загальним інтересам таким чином, щоб зменшити глобальні втрати. Однак є багато реальних додатки, які передбачають егоїзм гравців (агентів) наприклад, контроль перевантаження для тимчасових бездротових мереж і максимізація відносин оптичного сигналу/шуму (OSNR) в оптичних мережах. У цих додатках гравці бажають оптимізувати свою продуктивність, навіть якщо глобальна мета не може бути мінімізована, тому. У зв'язку з цим ми зацікавлені у вивченні рівноваги Неша цієї гри.

З огляду на різницю між розподіленої оптимізацією і розподіленим пошуком рівноваги Неша (NE), ми будемо використовувати метод оптимізації, званий метод змінного напрямку множників (*the alternating direction method of multipliers* (ADMM)) для знайти точку рівноваги в багатокористувацької грі. ADMM використовує два різних підходи, які використовуються при вирішенні задачі оптимізації: 1)Подвійна декомпозиція і 2)Розширені Лагранжеві методи.

Подвійна декомпозиція - це окремий випадок подвійного сходження. метод розв'язання оптимізаційної задачі, коли цільова функція роздільна щодо змінної x, тобто $f(x) \coloneqq \sum_{i=1}^{N} f_i(x_i)$, де $x = [x_1, \dots, x_N]^T$. Це розкладання призводить до N паралельних задач подвійного сходження, кожна з яких вирішена для . Цей паралелізм робить сходження швидшим.

Розширений Лагранжа метод більш надійний. Цей метод передбачає додавання штрафу до нормального Лагранжіану. Ми будемо прагнути використовувати переваги ADMM в контексті пошуку NE гри. Ось деяуі труднощі, які нам необхідно подолати:

1)Гру Неша можна розглядати як набір завдань паралельної оптимізації, кожна з яких пов'язана з мінімізацією власної функції витрат гравця щодо його змінної. Однак кожна оптимізаційна проблема залежить від рішення інших паралельних задач. Це призводить до наявності N лагранжіанів, при цьому кожен залежить від змінних інших гравців.

2) Кожен гравець i оновлює тільки свою змінну x_i, однак він вимагає також оцінки всіх інших змінних $(x_j)_{j\in\{1,\dots,N\},j\neq i}$, і оновлює їх по порядку вирішення його оптимізаційної задачі. Це вимагає додатковий крок в алгоритмі на основі комунікацій між гравцями.

3) Кожна задача оптимізації не в підходящому форматі суми поділюваних функцій, щоб дозволити пряме застосування ADMM.

Теоретична частина

Постановка задачі. Алгоритм ADMM

Розглянемо $V = \{1, \dots, N\}$ як набір з N гравців в мережевій розрахованій на багато користувачів грі. Гра позначається G і визначається наступним чином:

• $\Omega_i \subset R$: множина дій гравця i, $\forall i \in V$,

• $\Omega = \prod_{i \in V} \Omega_i \subset R^N$: набір дій всіх гравців,

• J_i: $\Omega \to R$: функція витрат гравця i, $\forall i \in V$.

Гра $G(V, \Omega_i, J_i)$ визначена над безліччю гравців V, безліч дій гравця $i \in V, \Omega_i$ функція вартості гравця $i \in V, J_i$.

Дії гравців позначаються наступним чином:

•$x = (x_i, x_{-i}) \in \Omega$: всі дії гравців,

•$x_i \in \Omega_i$: Дії гравця i, $\forall i \in V$,

•$x_{-i} \in \Omega_{-i} \coloneqq \prod_{j \in V \setminus \{i\}} \Omega_j$: дії всіх гравців, крім гравця i.

Гра ведеться так, що для даного $x_{-i} \in \Omega_{-i}$, кожен гравець $i \in V$ прагне мінімізувати свою функцію вартості, щоб знайти оптимальну дію,

$$\begin{cases} \underset{x_i}{\text{minimize}} \, J_i(x_i, x_{-i}) \\ \quad x_i \in \Omega_i \end{cases} \quad (2)$$

Кожним завданням оптимізації займається конкретний гравець I одночасно з іншими гравцями.

NE гри визначається наступним чином:

Визначення 1. Розглянемо гру $G(V, \Omega_i, J_i)$, кожен гравець i мінімізує функцію вартості $J_i: \Omega \to R$. Вектор $x^* = (x_i^*, x_{-i}^*) \in \Omega$ називається рівновагою Неша цієї гри, якщо

$$J_i(x_i^*, x_{-i}^*) \le J_i(x_i, x_{-i}^*) \quad \forall x_i \in \Omega_i, \forall i \in V. \ (3)$$

NE лежить на перетині всіх рішень множини (2). Проблема в тому, що кожна задача оптимізації в (2) залежить від рішення інших одночасних задач. І оскільки ця гра розподілена, жоден гравець не знає про дії(рішення) інших гравців(задачі).[3]

Ми припускаємо, що кожен гравець i оцінює дії інших гравців. Далі ми визначаємо кілька позначень для оцінок гравців.

• $x^i = (x_i^i, x_{-i}^i) \in \Omega$: оцінка гравцем i всіх дій гравців;

• $x_i^i \in \Omega_i$: Оцінка гравця i своєї дії, $\forall i \in V$;

• $x_{-i}^i \in \Omega_{-i} := \prod_{j \in V \setminus \{i\}} \Omega_j$: Оцінка гравцем i всіх дій інших гравців, крім його дій;

• $\underline{x} = [x^{1^T}, \dots, x^{N^T}]^T \in \Omega^N$: розширений вектор оцінки дій всіх гравців.

Звернемо увагу, що оцінка своєї дії гравцем i дійсно є його дією, тобто $x_i^i = x_i$ для $i \in V$. Також, що всі дії гравців $x = (x_i, x_{-i})$ можуть бути взаємозаміно представлені як $x = [x_i^i]_{i \in V}$.

Ми припускаємо, що функція вартості J_i і множина дій Ω - єдина інформація, доступна гравцеві i. Таким чином, гравцям необхідно обмінятися деякою інформацією, щоб оновити свої оцінки. Визначений граф $G_C(V, E)$, де $E \subseteq V \times V$ позначає набір каналів зв'язку між гравцями. $(i, j) \in E$ тоді і тільки тоді, коли гравці i і j обмінюються інформацією. Далі маємо кілька визначень для G_C:

• $N_i := \{J \in V | (i, j) \in E\}$: множина сусідів гравця i в G_C;

• $A := [a_{ij}]_{i,j \in V}$: матриця суміжності G_C, де a_{ij}=1, якщо $(i, j) \in E$ та a_{ij}=0 в іншому випадку,

• $D := \mathrm{diag}\{|N_1|, \dots, |N_N|\}$: степенева матриця G_C.

Використовується таке припущення.

Припущення 1. G_C - зв'язний граф.

Ми прагнемо пов'язати гру (2) з наступною задачею, у якій рішення може бути засноване на методі змінного напрямку множників.

$$\begin{cases} \underset{x \in C_1, z \in C_2}{\text{minimize}} \; G_1(x) + G_2(z) \\ Ax = z \end{cases}$$

Для цього формулюємо гру (2) так, щоб цільова функція віддільна за рахунок використання оцінок дій для кожного гравця $i \in V$ при x^i (Оцінки також

інтерпретується як локальні копії x). Зокрема, від (2) вважаємо, що для даного $x_{-i}^i \in \Omega_{-i}$, Кожен гравець $i \in V$ мінімізує свою функцію витрат щодо його власної дії з обмеженням-рівністю, тобто для всіх $i \in V$,

$$\begin{cases} \underset{x_i^i, x_i^j|_{j \in N_i} \in \Omega_i}{\text{minimize}} J_i(x_i^i, x_{-i}^i) + \sum_{j \in N_i} g_j(x_i^j, x_{-i}^j) \\ x_i^i = x_i^j \quad \forall j \in N_i \end{cases}$$

Де $g_j(\cdot) = 0$ для $j \in N_i$. Звернемо увагу, що для поновлення всіх елементів в x_{-i}^i потрібно посилити обмеження в векторну форму розміром $N \times 1$ $x^i = x^j$, $j \in N_i$. Більш того, замінимо обмеження на $x^l = x^s \; \forall l \in V, \forall s \in N_l$, що включає $x^i = x^j$, $j \in N_i$. Збільшення обмежень таким чином не впливає на рішення задачі. Тоді для даного $x_{-i}^i \in \Omega_{-i}$ і для всіх $i \in V$ ми отримуємо,

$$\begin{cases} \underset{x_i^j \in \Omega_i}{\text{minimize}} J_i(x_i^i, x_{-i}^i) \\ x^l = x^s \; \forall l \in V, \forall s \in N_l \end{cases} \quad (4)$$

Обмеження рівності разом з припущенням 1 забезпечує що всі локальні копії x ідентичні, тобто $x^1 = x^2 = \cdots = x^N$. Отже, (4) відновлює (2).

За **припущенням 1** сукупність задач (4) еквівалентно до наступного набору задач оптимізації: для заданого $x_{-i}^i \in \Omega_{-i}$ і для всіх $i \in V$

$$\begin{cases} \underset{x_i^i \in R}{\text{minimize}} J_i(x_i^i, x_{-i}^i) + \mathcal{I}_{\Omega_i}(x_i^i) \\ x^l = t^{ls} \; \forall l \in V, \forall s \in N_l \\ x^s = t^{ls} \; \forall l \in V, \forall s \in N_l \end{cases} \quad (5)$$

Де $\mathcal{I}_{\Omega_i}(x_i^i) \coloneqq \begin{cases} 0, & \text{якщо } x_i^i \in \Omega_i \\ \infty, & \text{в іншому випадку} \end{cases}$ - індикаторна функція обмеження здійсненності $x_i^i \in \Omega_i$ та t^{ls} - це проміжна змінна для поділу обмежень рівності.

Відзначимо, що набір задач (5) можна розглядати як то ж, що і набір задач (2), але з урахуванням N оцінок(локальні копії) дій гравців для кожного гравця $i \in V$.

Характеристика рівноваги Неша для гри (2) може бути отримана шляхом знаходження умов ККТ(Умови Каруша — Куна — Такера) на множині задач (5). Нехай $\{u^{ls}, v^{ls}\}_{l\in V, s\in N_l}$, де $u^{ls}, v^{ls} \in R^N$ - множники Лагранжа, пов'язані з двома обмеженнями в (5) відповідно. Відповідна функція Лагранжа для гравця i, $i \in V$ виглядає наступним чином:

$$L_i\left(x^i, \{t^{ls}\}_{l\in V, s\in N_l}, \{u^{ls}\}_{l\in V, s\in N_l}, \{v^{ls}\}_{l\in V, s\in N_l}\right) \coloneqq J_i\left(x_i^i, x_{-i}^i\right) + \mathcal{I}_{\Omega_i}\left(x_i^i\right) + \sum_{l\in V}\sum_{s\in N_l} u^{ls^T}\left(x^l - t^{ls}\right) + \sum_{l\in V}\sum_{s\in N_l} v^{ls^T}\left(x^s - t^{ls}\right), \quad (6)$$

Нехай $(x^{i^*})_{i\in V}$ і $\{u^{ls}, v^{ls}\}_{l\in V, s\in N_l}$ - пара оптимальних прямого і двоїстого рішення рівняння (5). Умови ККТ резюмується в такий спосіб:

$$\nabla_i J_i\left(x^{i^*}\right) + \partial_i \mathcal{I}_{\Omega_i}\left(x_i^{i^*}\right) + \sum_{j\in N_i} u_i^{ij^*} + v_i^{ij^*} = 0 \ \forall i \in V \quad (7)$$

$$x^{i^*} = x^{j^*} \quad \forall i \in V, \ \forall j \in N_i \quad (8)$$

$$u_i^{ij^*} + v_i^{ij^*} = 0_N \quad \forall i \in V, \forall j \in N_i \quad (9)$$

де $\nabla_i J_i(\cdot)$ - градієнт J_i щодо x_i і $\partial_i \mathcal{I}_{\Omega_i}(\cdot)$ - субградієнт $\mathcal{I}_{\Omega_i}$в x_i . Зауважимо, що індекс v^{ij^*}в (7) обернений v^{ij^*}в (9). Згідно (8) і **припущенню 1**, $x^{1^*} = \cdots = x^{N^*} \coloneqq x^*$. Таким чином, $x^* \coloneqq (x_i^*, x_{-i}^*)$ є рішенням з (5) тоді і тільки тоді, коли,

$$\begin{cases} \nabla_i J_i\left(x^{i^*}\right) + \partial_i \mathcal{I}_{\Omega_i}\left(x_i^{i^*}\right) + \sum_{j\in N_i} u_i^{ij^*} + v_i^{ij^*} = 0 \ \forall i \in V \\ u_i^{ij^*} + v_i^{ij^*} = 0_N \quad \forall i \in V, \forall j \in N_i \end{cases} \quad (10)$$

Сформулюємо кілька припущень про існування і унікальність рівноваги Неша.

Припущення 2. Для будь-якого $i \in V$ множина дій Ω_i це непорожня, компактна і опукла підмножина R. $J_i\left(x_i^i, x_{-i}^i\right)$ є безперервно диференційованою функцією по x_i, спільно неперервна по x і опукла по x_i для будь-якого x_{-i}.

Опуклість Ω_i випливає з того, що$\mathcal{I}_{\Omega_i}$- опукла функція.

Звідси випливає, що існує хоча б один обмежений субградієнт $\partial \mathcal{I}_{\Omega_i}$.

Припущення 3. Нехай $F: \Omega^N \to R^N, F(\underline{x}) \coloneqq [\nabla_i J_i(x^i)]_{i \in V}$ – вектор псевдоградієнта, де $\underline{x} = [{x^1}^T, \dots, {x^N}^T]^T \in \Omega^N$. F коерцитивна $\forall \underline{x} \in \Omega^N$ і $\forall \underline{y} \in \Omega^N$, тобто

$$\left(F(\underline{x}) - F\left(\underline{y}\right)\right)^T (x - y) \geq \sigma_F \left|\left|F(\underline{x}) - F\left(\underline{y}\right)\right|\right|^2, (11)$$

де $\sigma_F > 0$.

Розподілений алгоритм ADMM

Стисло пояснимо механізм роботи алгоритму наступним чином: кожен гравець оцінює дії всіх гравців і локально обмінюється інформацією зі своїми сусідами по G_C. Потім він бере середнє значення інформації своїх сусідів і використовує її для поновлення своїх оцінок. Алгоритм складається з наступних етапів:

1) **Крок ініціалізації:** кожен гравець $i \in V$ має початкова оцінка для всіх гравців, $x^i(0) \in \Omega$. Початкові значення з вас $u^{ij}(0)$ і $v^{ij}(0)$ дорівнюють нулю для всіх $i \in V, j \in N_i$.

2) **Етап комунікації:** на ітерації $T(k)$ кожен гравець $i \in V$ обмінюється своєю оцінкою дій інших гравців зі своїми сусідами j, $\forall j \in N_i$. Потім він бере в середньому отриману інформацію з його оцінкою і оновлює його оцініте такий спосіб:

$$x_{-i}^{i}(k) = \frac{1}{2}\left(x_{-i}^{i}(k-1) + \frac{1}{|N_i|}\sum_{j \in N_i} x_{-i}^{j}(k-1) \right) - \\ - \frac{1}{2c|N_i|}\sum_{j \in N_i}\left(u_{-i}^{ij}(k) + v_{-i}^{ij}(k)\right), \quad (12)$$

де $\frac{1}{|N_i|}\sum_{j \in N_i} x_{-i}^{j}(k-1)$ – отримана інформація,

$\frac{1}{2c|N_i|}\sum_{j \in N_i}\left(u_{-i}^{ij}(k) + v_{-i}^{ij}(k)\right)$ - штрафний коефіцієнт,

де $c > 0$ - скалярний коефіцієнт, а $\forall i \in V, j \in N_i$

$$u^{ij}(k) = u^{ij}(k-1) + \frac{c}{2}\left(x^i(k-1) - x^j(k-1)\right), \quad (13)$$

$$v^{ij}(k) = v^{ij}(k-1) + \frac{c}{2}\left(x^j(k-1) - x^i(k-1)\right). \quad (14)$$

Рівняння (13), (14) являють собою правила поновлення подвійних множників Лагранжа. Відзначимо, що в (12) штрафний коефіцієнт $\frac{1}{2c|N_i|}\sum_{j \in N_i}\left(u_{-i}^{ij}(k) + v_{-i}^{ij}(k)\right)$ віднімається, що пов'язано з різницею оцінок сусідніх гравців (рівняння (13), (14)).

Зауваження. На відміну від розподілених алгоритмів оптимізації, де мінімізація щодо x, тут кожен гравець мінімізує свою функція вартості щодо x_i^i. Щоб оновити x_i^i, Кожному гравцеві потрібно оцінка інших гравців x_{-i}^iна кожній ітерації.

Таким чином, крок зв'язку неминучий для поновлення x_{-i}^i для наступної ітерації.

3) **Крок оновлення:** У цей момент всі гравці оновлюють свої дії за допомогою ADMM-подібний підхід розвивається таким чином. Для кожного гравця i, $\forall i \in V$, нехай розширена функція Лагранжа асоціює до задачі (5) мати такий вигляд

$$
\begin{aligned}
&{L_i}^a\left(x^i, \{t^{ls}\}_{l\in V, s\in N_l}, \{u^{ls}\}_{l\in V, s\in N_l}, \{v^{ls}\}_{l\in V, s\in N_l}\right) \\
&\qquad \coloneqq J_i\left(x_i^i, x_{-i}^i\right) + \mathcal{I}_{\Omega_i}\left(x_i^i\right) + \sum_{l\in V}\sum_{s\in N_l} {u^{ls}}^T (x^l - t^{ls}) \\
&\qquad + \sum_{l\in V}\sum_{s\in N_l} {v^{ls}}^T (x^s - t^{ls}) \\
&\qquad + \frac{c}{2}\sum_{l\in V}\sum_{s\in N_l}\left(\left\|x^l - t^{ls}\right\|^2 + \left\|x^s - t^{ls}\right\|^2\right). \quad (15)
\end{aligned}
$$

де $c > 0$ - скалярний коефіцієнт, який також використовується в рівняннях (12), (13) та (14). Розглянемо алгоритм ADMM, зв’язаний з задачею (5) на основі (15):

$$x_i^i(k) = \underset{x_i^i \in R}{\arg\min} L_i{}^a \left(\left(x_i^i, x_{-i}^i(k - 1)\right), \{t^{ls}(k-1)\}_{l\in V, s\in N_l}, \{u^{ls}(k)\}_{l\in V, s\in N_l}, \{v^{ls}(k)\}_{l\in V, s\in N_l}\right)$$
$$= \underset{x_i^i \in R}{\arg\min} \left\{ J_i\left(x_i^i, x_{-i}^i(k-1)\right) + \mathcal{I}_{\Omega_i}\left(x_i^i\right) + \sum_{j\in N_i} \left(u^{ij}(k) + v^{ji}(k)\right)^T \left(x_i^i, x_{-i}^i(k-1)\right) + c\sum_{j\in N_i} \left\| x_i^i, x_{-i}^i(k-1) - t^{ij}(k-1)\right\|^2 \right\}\Bigg). \ \forall i \in V, \qquad (16)$$

Правило оновлення допоміжної змінної t^{ij} $\forall i \in V, j \in N_i$ заснований на (15).

$$t^{ij}(k) = \underset{t^{ij}}{\arg\min}\ L_i{}^a \left(\left(x_i^i(k), x_{-i}^i(k - 1)\right), \{t^{ls}\}_{l\in V, s\in N_l}, \{u^{ls}(k)\}_{l\in V, s\in N_l}, \{v^{ls}(k)\}_{l\in V, s\in N_l}\right) =$$
$$\underset{t^{ij}}{\arg\min} \left\{ -\left(u^{ij}(k) + v^{ij}(k)\right)^T t^{ij} + \frac{c}{2}\left(\left\| \left(x_i^i(k), x_{-i}^i(k-1)\right) - t^{ij} \right\|^2 + \left\| \left(x_i^j(k), x_{-i}^j(k-1)\right) - t^{ij} \right\|^2\right) \right\} = \frac{1}{2c}\left(u^{ij}(k) + v^{ij}(k)\right) +$$
$$\frac{1}{2}\left(\left(x_i^i(k), x_{-i}^i(k-1)\right) + x_i^j(k), x_{-i}^j(k-1)\right)). (17)$$

Початкові умови $u^{ij}(0) = v^{ij}(0) = 0_N$ $\forall i \in V, \forall\, j \in N_i$ разом з (13) і (14) припускаємо, що $u^{ij}(k) + v^{ij}(k) = 0_N$ $\forall i \in V, \forall\, j \in N_i$ $k > 0$.
Тоді

$$t^{ij}(k) = \frac{\left(x_i^i(k), x_{-i}^i(k-1)\right) + \left(x_i^j(k), x_{-i}^j(k-1)\right)}{2}. \ (18)$$

Підставивши (18) у (16), отримаємо оновлення локальної оцінки для всіх $i \in V$ наступним чином

$$x_i^i(k) = \underset{x_i^i \in R}{\arg\min} \{J_i\left(x_i^i, x_{-i}^i(k-1)\right) + \mathcal{I}_{\Omega_i}(x_i^i) + \sum_{j \in N_i}\left(u^{ij}(k) + v^{ji}(k)\right)^T \left(x_i^i, x_{-i}^i(k-1)\right) + c\sum_{j \in N_i}\left\|\left(x_i^i, x_{-i}^i(k-1)\right) - \frac{\left(x_i^i(k-1), x_{-i}^i(k-2)\right) + \left(x_i^j(k-1), x_{-i}^j(k-2)\right)}{2}\right\|^2\} \quad (19)$$

Спростимо (19), використовуючи проксимальне приближення першого порядку для $J_i\left(x_i^i, x_{-i}^i(k-1)\right)$ для $x^i(k-1)$. Таким чином використовуючи ADMM отримуємо:

$$x_i^i(k) = \underset{x_i^i \in R}{\arg\min} \left\{\nabla_i J_i\left(x^i(k-1)\right)^T \left(x_i^i - x_{-i}^i(k-1)\right) + \frac{\beta_i}{2}\left\|\left(x_i^i - x_{-i}^i(k-1)\right)\right\|^2 + \mathcal{I}_{\Omega_i}(x_i^i) + \sum_{j \in N_i}(u_i^{ij}(k) + v_i^{ji}(k)(x_i^i) + c\sum_{j \in N_i}\left\|(x_i^i) - \frac{\left(x_i^i(k-1)\right) + \left(x_i^j(k-1)\right)}{2}\right\|^2\right\}$$

$$\forall i \in V \quad (20)$$

Де $\beta_i > 0$ – штрафний коефіцієнт проксимального першого порядку апроксимація функції затрат кожного гравця i.

На цьому етапі гравці готові почати нову ітерацію з кроку 2. Таким чином, алгоритм складається з (12), (13), (14) і (20), які є правилом поновлення для оцінки гравців, крім власних дій, правила оновлення для множників Лагранжа і правило поновлення для дії гравця відповідно.

Доведення збіжності алгоритму AMDD

Теорема 1. Нехай $\beta_{min} \coloneqq \min_{i \in V} \beta_i > 0^3$ – мінімальний штрафний коефіцієнт апроксимації в неточному ADMM алгоритмі, який задовольняє

$$\sigma_F > \frac{1}{2(\beta_{min} + c\lambda_{min}(D+A))}, \quad (21)$$

де σ_F - позитивна константа для коерцитивної властивості матриці F, а D і A - матриці ступеня і суміжності G_C відповідно. У припущеннях 1-3 послідовність $\{x^i(k)\}\ \forall i \in V$, породжена алгоритмом (12), (13), (14) і (20) сходиться до x^* рівноваги Неша гри (2).[4]

Доведення

Умова оптимальності (20) дає

$$\nabla_i J_i(x^i(k-1)) + \beta_i\left(x_i^i(k) - x_i^i(k-1)\right) + \partial_i \mathcal{I}_{\Omega_i}\left(x_i^i(k)\right) + \sum_{j \in N_i}(u_i^{ij}(k) + v_i^{ji}(k)) + 2c\sum_{j \in N_i}(x_i^i(k) - \frac{x_i^i(k-1) + x_i^j(k-1)}{2}) = 0. \quad (22)$$

Сумістимо (22) з (10), яке представляє собою рівняння пов'язаних з рішеннями сукупності задача (5) (Рівновага Неша в грі (2)). Тоді отримуємо

$$\nabla_i J_i(x^i(k-1)) - \nabla_i J_i(x^*) + \beta_i\left(x_i^i(k) - x_i^i(k-1)\right) + \partial_i \mathcal{I}_{\Omega_i}\left(x_i^i(k)\right) - \partial_i \mathcal{I}_{\Omega_i}(x^*) + \sum_{j \in N_i}(u_i^{ij}(k) + v_i^{ji}(k)) + 2c\sum_{j \in N_i}(x_i^i(k) - \frac{x_i^i(k-1) + x_i^j(k-1)}{2}) = 0$$

(23)

Множимо обидві частини на $(x_i^i(k) - x_i^*)$, а потім додамо і віднімемо $x_i^i(k-1)$ наступним чином:

$(\nabla_i J_i(x^i(k-1)) - \nabla_i J_i(x^*))^T(x_i^i(k-1) - x_i^*) + (\nabla_i J_i(x^i(k-1)) - \nabla_i J_i(x^*))^T(x_i^i(k) - x_i^i(k-1)) + \beta_i\left(x_i^i(k) - x_i^i(k-1)\right)^T(x_i^i(k) - x_i^*) + (\partial \mathcal{I}_{\Omega_i}(x_i^i(k)) - \partial \mathcal{I}_{\Omega_i}(x_i^*))^T(x_i^i(k) - x_i^*) + \sum_{j \in N_i}((u_i^{ij}(k) + v_i^{ji}(k) - u_i^{ij^*} -$

$v_i^{ji^*})^T(x_i^i(k) - x_i^*) + \ 2c\sum_{j\in N_i}\left(x_i^i(k) - \frac{x_i^i(k-1)+x_i^j(k-1)}{2}\right)^T (\ x_i^i(k) - x_i^*) = 0$

(24)

Крім поновлення власних дій, гравцям необхідно оновити свої оцінки також. Далі пояснюємо, як ввести правило оновлення x_{-i}^i в (24).

Відзначимо, що з (12) можна отримати,

$$\sum_{j\in N_i}(u_{-i}^{ij}(k) + v_{i-}^{ji}(k)) + \ 2c\sum_{j\in N_i}\left(x_{-i}^i(k) - \frac{x_{-i}^i(k-1)+x_{-i}^j(k-1)}{2}\right) = 0_{N-1} \quad (25)$$

Помножимо (25) на ($x_{-i}^i(k) - x_{-i}^*$) та отримаємо

$$\sum_{j\in N_i}((u_{-i}^{ij}(k) + v_{-i}^{ji}(k))^T(x_{-i}^i(k) - x_{-i}^*) + \ 2c\sum_{j\in N_i}\left(x_{-i}^i(k) - \frac{x_{-i}^i(k-1)+x_{-i}^j(k-1)}{2}\right)^T (\ x_{-i}^i(k) - x_{-i}^*) = 0 \quad (26)$$

Додаючи (26) до (24) і використовуючи (13), (14), отримуємо $i \in V$,

$$(\nabla_i J_i(x^i(k-1)) - \nabla_i J_i(x^*))^T(\ x_i^i(k-1) - x_i^*) + \ (\nabla_i J_i(x^i(k-1)) - \nabla_i J_i(x^*))^T(x_i^i(k) - x_i^i(k-1)) + \beta_i\left(x_i^i(k) - x_i^i(k-1)\right)^T(\ x_i^i(k) - x_i^*) + (\partial \mathcal{I}_{\Omega_i}(\ x_i^i(k)) - \partial \mathcal{I}_{\Omega_i}(x_i^*))^T(\ x_i^i(k) - x_i^*) + \sum_{j\in N_i}\left((u_i^{ij}(k+1) + v_i^{ji}(k+1) - u_i^{ij^*} - v_i^{ji^*}\right)^T\left(x_i^i(k) - x_i^*\right) + \sum_{j\in N_i}\left((u_{-i}^{ij}(k+1) + v_{-i}^{ji}(k+1)\right)^T\left(x_{-i}^i(k) - x_{-i}^*\right) + \ 2c\sum_{j\in N_i}\left(\frac{x^i(k)+x^j(k)}{2} - \frac{x^i(k-1)+x^j(k-1)}{2}\right)^T\left(x^i(k) - x^*\right) = 0. \quad (27)$$

Другий і третій члени обмежені наступним чином:

$$(\nabla_i J_i(x^i(k-1)) - \nabla_i J_i(x^*))^T\left(x_i^i(k) - x_i^i(k-1)\right) \geq \frac{-1}{2\rho}\left\|\nabla_i J_i(x^i(k-1)) - \nabla_i J_i(x^*)\right\|^2 - \frac{\rho}{2}\left\|x_i^i(k) - x_i^i(k-1)\right\|^2, \quad (28)$$

для будь-якого $\rho > 0\ \forall i \in V$. За опуклості $\mathcal{I}_{\Omega_i}$ (**Припущення 2**) для четвертого члена маємо

$$(\partial \mathcal{I}_{\Omega_i}\left(x_i^i(k)\right) - \partial \mathcal{I}_{\Omega_i}(x_i^*))^T(x_i^i(k) - x_i^*) \ \geq 0. \quad (29)$$

Використовуючи (28) і (29) в (27) і підсумувавши по $i \in V$, отримуємо

$$\left(F\left(\underline{x}(k-1)\right) - F(\underline{x^*})\right)^T (x(k-1) - x^*) - \frac{1}{2\rho}\left\|F\left(\underline{x}(k-1)\right) - F(\underline{x^*})\right\|^2 - \frac{1}{2}\left\|\underline{x}(k) - \underline{x}(k-1)\right\|_{M_1}^2 + \left(\underline{x}(k) - \underline{x}(k-1)\right)^T \operatorname{diag}\left(\left(\beta_i e_i e_i^T\right)_{i\in V}\right)\left(\underline{x}(k) - \underline{x^*}\right) + \sum_{i\in V}\sum_{j\in N_i}\left(u_i^{ij}(k+1) + v_i^{ji}(k+1) - u_i^{ij^*} - v_i^{ji^*}\right)^T \left(x_i^i(k) - x_i^*\right) + \sum_{i\in V}\sum_{j\in N_i}\left(u_{-i}^{ij}(k+1) + v_{-i}^{ji}(k+1)\right)^T \left(x_{-i}^i(k) - x_{-i}^*\right) + c\left(\underline{x}(k) - \underline{x}(k-1)\right)^T ((D+A)\otimes I_N)(\underline{x}(k) - \underline{x^*}) \le 0, \quad (30)$$

Де $M_1 := \operatorname{diag}\left(\left(\rho e_i e_i^T\right)_{i\in V}\right)$ і $\underline{x^*} = [x^{1^{*^T}}, \dots, x^{N^{*^T}}]^T$.

Оцінимо перший член, використовуючи **припущення 3**

$$\left(F\left(\underline{x}(k-1)\right) - F(\underline{x^*})\right)^T (x(k-1) - x^*) \ge \sigma_F \left\|F\left(\underline{x}(k-1)\right) - F(\underline{x^*})\right\|^2. \quad (31)$$

Ми також спрощуємо п'ятий і шостий члени в (30). Оскільки G_C - неорієнтований граф, для будь-якого $\{a_{ij}\}, \sum_{i\in V}\sum_{j\in N_i} a_{ij} = \sum_{i\in V}\sum_{j\in N_i} a_{ji}$. Тоді

$$\sum_{i\in V}\sum_{j\in N_i}\left((u_i^{ij}(k+1) + v_i^{ji}(k+1) - u_i^{ij^*} - v_i^{ji^*}\right)^T \left(x_i^i(k) - x_i^*\right) + \sum_{i\in V}\sum_{j\in N_i}\left((u_{-i}^{ij}(k+1) + v_{-i}^{ji}(k+1)\right)^T \left(x_{-i}^i(k) - x_{-i}^*\right) = \sum_{i\in V}\sum_{j\in N_i}\left((u_i^{ij}(k+1) - u_i^{ij^*}\right)^T \left(x_i^i(k) - x_i^*\right) + \sum_{i\in V}\sum_{j\in N_i}\left(v_i^{ji}(k+1) - -v_i^{ji^*}\right)^T \left(x_i^j(k) - x_i^*\right) + \sum_{i\in V}\sum_{j\in N_i}\left(u_{-i}^{ij}(k+1)\right)^T \left(x_{-i}^i(k) - x_{-i}^*\right) + \sum_{i\in V}\sum_{j\in N_i}\left(v_i^{ji}(k+1)\right)^T \left(x_{-i}^j(k) - x_{-i}^*\right) \quad (32)$$

Відзначимо, що згідно з (13) і (14), а також початкових умов для множників Лагранжа ми отримуємо,

$$u^{ij}(k) + v^{ij}(k) = 0_N \ \forall i \in V, j \in N_i, k > 0. \quad (33)$$

Підставляючи (33) у (32) та використовуючи (8), ми отримуємо,

$\sum_{i\in V}\sum_{j\in N_i}\left(u_i^{ij}(k+1)-u_i^{ij^*}\right)^T\left(x_i^i(k)-x_i^j(k)\right)+\sum_{i\in V}\sum_{j\in N_i}\left(u_{-i}^{ij}(k+1)\right)^T\left(x_{-i}^i(k)-x_{-i}^j(k)\right)=\sum_{i\in V}\sum_{j\in N_i}(u^{ij}(k+1)-u_i^{ij^*}e_i)^T\left(x^i(k)-x^j(k)\right)=\frac{2}{c}\sum_{i\in V}\sum_{j\in N_i}(u^{ij}(k+1)-u_i^{ij^*}e_i)^T\left(u^{ij}(k+1)-u^{ij}(k)\right)\coloneqq\frac{2}{c}\left(\underline{u}(k+1)-\underline{u^*}\right)^T(\underline{u}(k+1)-\underline{u}(k))$ (34)

де $\underline{u}=(u^i)_{i\in V}\in R^{N\sum_{i\in V}|N_i|}$ і $u^i=(u^{ij})_{j\in N_i}\in R^{N|N_i|}$.

Також $\underline{u^*}=(u^{i^*})_{i\in V}\in R^{N\sum_{i\in V}|N_i|}$ та $u^{i^*}=(u_i^{ij^*})_{j\in N_i}\otimes e_i\in R^{N|N_i|}$.

Використовуючи (31) та (34), для $\rho=\frac{1}{2\sigma_F}$ (30) маємо,

$-\frac{1}{2}\left\|\underline{x}(k)-\underline{x}(k-1)\right\|_{M_1}^2+\left(\underline{x}(k)-\underline{x}(k-1)\right)^T M_2\left(\underline{x}(k)-\underline{x^*}\right)+\frac{2}{c}\left(\underline{u}(k)-\underline{u^*}\right)^T\left(\underline{u}(k+1)-\underline{u}(k)\right)\leq 0$ (35)

де $M_2\coloneqq\operatorname{diag}\left(\left(\beta_i e_i e_i^T\right)_{i\in V}\right)+c((D+A)\otimes I_N)$. Відзначимо, що $\operatorname{diag}\left(\left(\beta_i e_i e_i^T\right)_{i\in V}\right)\succcurlyeq 0$. Також зауважимо, що

$c\left((D+A)\otimes I_N\right)=\mathrm{c}\left((2D-L)\otimes I_N\right)=c\left(\left(D^{\frac{1}{2}}\left(2I-D^{-\frac{1}{2}}LD^{-\frac{1}{2}}\right)D^{\frac{1}{2}}\right)\otimes I_N\right)=c(\left(D^{\frac{1}{2}}(2I-L_N)D^{\frac{1}{2}}\right)\otimes I_N)$, (36)

де $L\coloneqq D-A$, $D^{\frac{1}{2}}$, $D^{-\frac{1}{2}}$ і $L_N\coloneqq D^{-\frac{1}{2}}LD^{-\frac{1}{2}}$ є лапласіаном G_C, квадратним коренем і зворотним квадратним коренем з D і нормалізований лапласіан з G_C, відповідно. Оскільки D . Тоді (36) дає $c\left((D+A)\otimes I_N\right)\succcurlyeq 0$. Звідки маємо $M_2\succcurlyeq 0$.

Отриману рівність використовуємо в (35) для кожного $\{a(k)\}$ та $M\succcurlyeq 0$:

$\left(a(k)-a(k-1)\right)^T M(a(k)-a^*)=\frac{1}{2}\|a(k)-a^*\|_M^2+\frac{1}{2}\|a(k)-a(k-1)\|_M^2-\frac{1}{2}\|a(k-1)-a^*\|_M^2.$ (37)

Тоді (35) перетворюється

$\frac{1}{2}\left\|\underline{x}(k)-\underline{x^*}\right\|_{M_2}^2 + \frac{1}{c}\left\|\underline{u}(k+1)-\underline{u^*}\right\|^2 \le \frac{1}{2}\left\|\underline{x}(k-1)-\underline{x^*}\right\|_{M_2}^2 + \frac{1}{c}\left\|\underline{u}(k)-\underline{u^*}\right\|^2 - \frac{1}{2}\left\|\underline{x}(k)-\underline{x}(k-1)\right\|_{M_2-M_1}^2 - \frac{1}{c}\left\|\underline{u}(k+1)-\underline{u}(k)\right\|^2.$ (38)

За умовою (21), тоді з (38) отримуємо 2 результати:

1) $\frac{1}{2}\left\|\underline{x}(k)-\underline{x^*}\right\|_{M_2}^2 + \frac{1}{c}\left\|\underline{u}(k+1)-\underline{u^*}\right\|^2 \to \theta$, для будь-яких $\theta \ge 0$

2) $\begin{cases} \underline{x}(k)-\underline{x}(k-1) \to 0_{N^2} \\ \underline{u}(k+1)-\underline{u}(k) \to 0_{N\sum_{i\in V}|N_i|} \end{cases}$

З результату 1 слідує, що послідовності $\{x^i(k)\}$ і $\{u^{ij}(k)\}$(Аналогічно $\{v^{ij}(k)\}$) обмежені і мають граничні точки, що позначається $\tilde{x}^i$і $\tilde{u}^{ij}$ ($\tilde{v}^{ij}$) відповідно. Тоді отримуємо

$$\theta = \frac{1}{2}\left\|\underline{\tilde{x}}-\underline{x^*}\right\|_{M_2}^2 + \frac{1}{c}\left\|\underline{\tilde{u}}-\underline{u^*}\right\|^2 \qquad (39)$$

Результат 2 також дає $\tilde{x}^i = \tilde{x}^j := \tilde{x}$ для всіх $\forall i \in V, j \in N_i$ оскільки з (13) ми маємо

$\frac{c}{2}\left(x^i(k)-x^j(k)\right) = u^{ij}(k+1)-u^{ij}(k) \to 0_N \Rightarrow \tilde{x}^i = \tilde{x}^j \; \forall i \in V, j \in N_i$ (40)

Крім того, згідно з (33) отримуємо

$$\tilde{u}^{ij} + \tilde{v}^{ij} = 0_N \; \forall i \in V, j \in N_i \qquad (41)$$

Із результату 2 також слідує, що згідно з (22) і (40):

$$\nabla_i J_i(\tilde{x}) + \partial \mathcal{I}_{\Omega_i}(\tilde{x}_i) + \textstyle\sum_{j\in N_i}({\tilde{u}^{ij}}_i + {\tilde{u}^{ji}}_i) = 0\,. \qquad (42)$$

Порівнюючи (41), (42) з (10) отримуємо $\forall i \in V, j \in N_i$

$$\tilde{x}^i = x^* \;\; (\underline{\tilde{x}} = \underline{x^*}), \quad (43)$$

$$\tilde{u}^{ij} = {u^{ij}}^* \;\; \left(\underline{\tilde{u}} = \underline{u^*}\right). \; (44)$$

Використовуючи (43) і (44) в (39), отримуємо $\theta = 0$. Таким чином, з результату 1 можна зробити висновок, що,

$$\frac{1}{2}\left\|\underline{x}(k)-\underline{x^*}\right\|_{M_2}^2 + \frac{1}{c}\left\|\underline{\tilde{u}}-\underline{u^*}\right\|^2 \to 0$$

Отримали те, що і потрібно було довести.

Payoff-Based Algorithm (Алгоритм заснований на виграші)

Постановка задачі

Розглянемо гру $G(V, \{\Omega_i\}, \{J_i\})$ з N гравців, множині дій гравців $\Omega_i \subset R^d, i \in V$, і функції вартості $J_i: \Omega \to R$, де$\Omega = \Omega_1 \times \ldots \times \Omega_N$– множина спільних дій. Ми можемо зробити такі припущення щодо гри G.

Припущення 4: Гра, що ми розглядаємо є опукла. А саме, для всіх і ∈ V множина Ω_i опукла і компактна, функція вартості $J_i(x^i, x^{-i})$ визначена на R^{Nd}, безперервно диференційовна по x і опукла по x^i фіксованого x^{-i}.

Припущення 5: Відображення $M: R^{Nd} \to R^{Nd}$, вказане, як відображення гри, визначається таким чином

$$M(x) = [\nabla_{x^i} J_i(x^i, x^{-i})]_{i=1}^N = [M_{1,1}, \ldots, M_{1,d}, \ldots, M_{N,1}, \ldots, M_{N,d}]^T,$$

$$M_{i,k}(x) = \frac{\partial J_i(x)}{\partial x_k^i} \quad x \in \Omega, i \in V$$

є Ліпшицевим відображенням на R^{Nd} та псевдомонотонною на Ω.

Визначення2: Відображення $M: R^d \to R^d$ називається псевдомонотонним на $X \subseteq R^d$, якщо $(M(y), x - y) \geq 0$ передбачає $(M(x), x - y) \geq 0$ для будь-яких $x, y \in X$.

Визначення 3: Відображення $M: R^d \to R^d$ називається відображенням Ліпшиця, якщо знайдеться така константа L на $X \subseteq R^d$, якщо $\|M(x) - M(y)\| \leq L(\|x - y\|$ для будь-яких $x, y \in X$.

Припущення 6: Функція $J_i(x), i \in V$ зростає не швидше, ніж поліном $\|x\| \to \infty$.

Опис алгоритму

У цьому розділі ми сформулюємо алгоритм, заснований на виграші для розподіленого пошуку рівноваги Неша x^* в грі $G(V, \Omega_i, J_i)$, яка задовольнить припущеннями 3-4. [5]

Доступ до інформації на поточний стан $x^i(t) = [x_1^i, \dots, x_d^i]^T \in R^d$ на ітерації *t* і поточне значення витрат $\widehat{J_i}(t)$ в спільному стані $x(t)$, $\widehat{J_i}(t) = J_i(x^i(t), \dots, x^N(t))$, кожен агент «перемішує» свій наступний стан $x^i(t+1)$, а саме вибирає своє наступне стан $x^i(t+1)$ випадково згідно багатовимірному нормальному розподілу $N(\mu^i(t) = \left[\mu_1^i(t), \dots, \mu_d^i(t)\right]^T, \sigma(t))$з щільністю:

$$p_i\left(x_1^i, \dots, x_d^i; \mu^i(t+1), \sigma(t+1)\right) = \frac{1}{(\sqrt{2\pi}\sigma(t+1))^d} \exp\{-\sum_{k=1}^{d} \frac{\left(x_k^i - \mu_k^i(t+1)\right)^2}{2\sigma^2(t+1)}\}.$$

Наш вибір Гаусового розподілу заснований на ідеї CALA (*continuous action-set learning automaton*, автоматичний екзаменатор з безперервним процесом). Середній параметр $\mu^i(t)$ для розподілу стану оновлюється наступним чином:

$$\mu^i(t+1) = \mathrm{Proj}_{\Omega_i}[\mu^i(t) - \gamma(t+1)\sigma^2(t+1)\widehat{J_i}(t)\frac{x^i(t) - \mu^i(t)}{\sigma^2(t)}]$$

Вище Proj_C позначає оператор проекції на множині С. Початкове значення $\mu(0)$ можна визначити довільно. Підкреслюємо різницю між станами і діями. Зокрема, стани є проміжними цінностями $x(t) = [x^i(t), \dots, x^N(t)]$ оновлюється під час виплати на основі розглянутий алгоритм. Вони не повинні належати до множини спільних дій Ω.

Практична частина

Алгоритм ADMM

1: initialization: $x^{\boldsymbol{i}}(\mathbf{0}) \in \Omega,\ w^i(0) = 0_N\ \forall i \in V$

2: for *k*=1,2… **do**

3: **for** кожен гравець $i \in V$ **do**

4: гравці обмінюються оцінками з сусідом

5: $w^i(k) = w^i(k-1) + c\sum_{j\in N_i}(x^{\boldsymbol{i}}(k-1) - x^{\boldsymbol{j}}(k-1))$

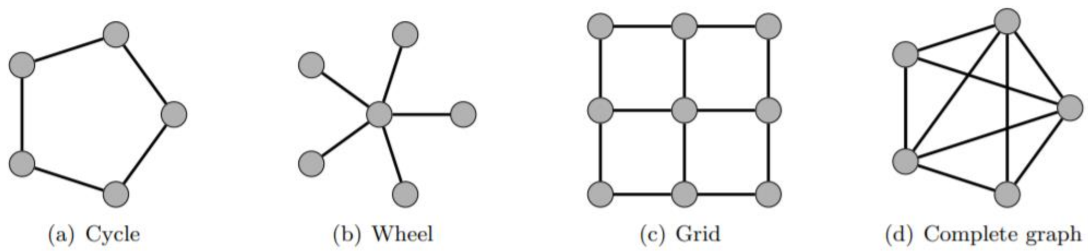
6: $x^i_{-i}(k) = \frac{\sum_{j\in N_i} x^j_{-i}(k-1)}{|N_i|} - \frac{w^i_{-i}(k-1)}{2c|N_i|}$

7: $x^i_i(k) = T_{\Omega_i}[\frac{\beta_i + c|N_i|}{\alpha_i} x^i_i(k-1) - \alpha_i^{-1}(w^i_i(k) + \nabla_i J_i\left(x^i(k-1)\right) - c\sum_{j\in N_i} x^j_i(k-1))]$

8: **end for**

9:end for

Для різних графів даний алгоритм буде працювати з різною ефективністю. Розглянемо такі графи:

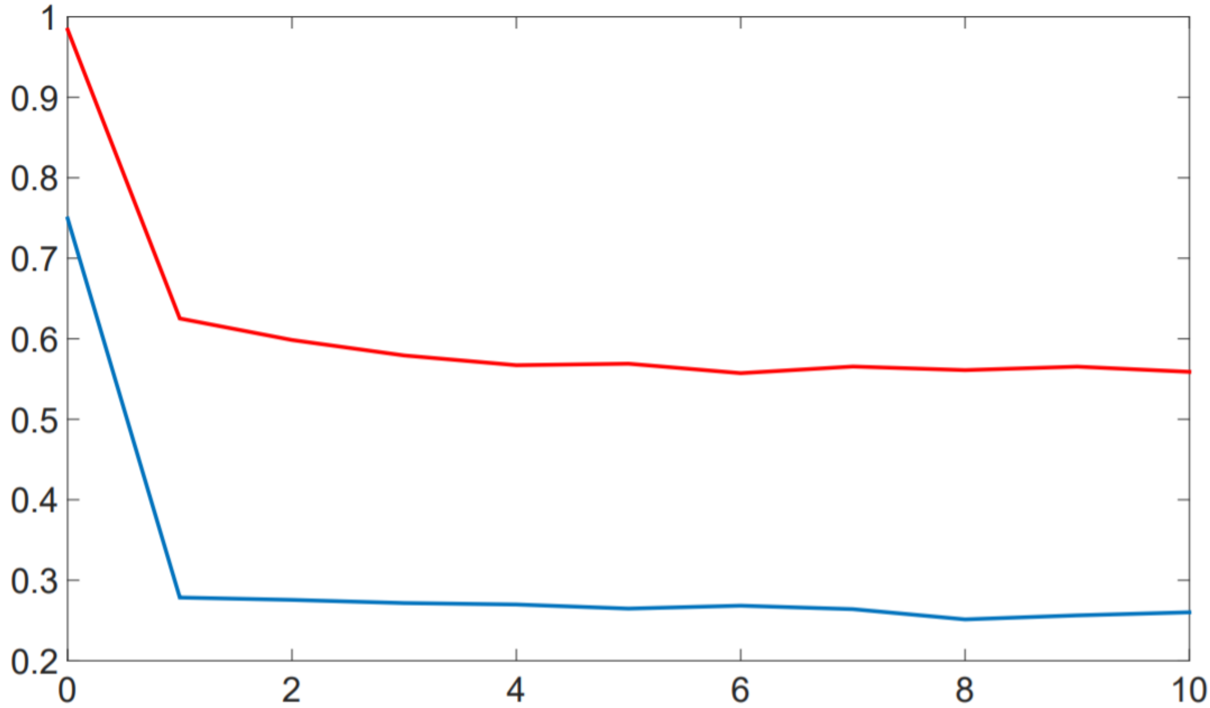


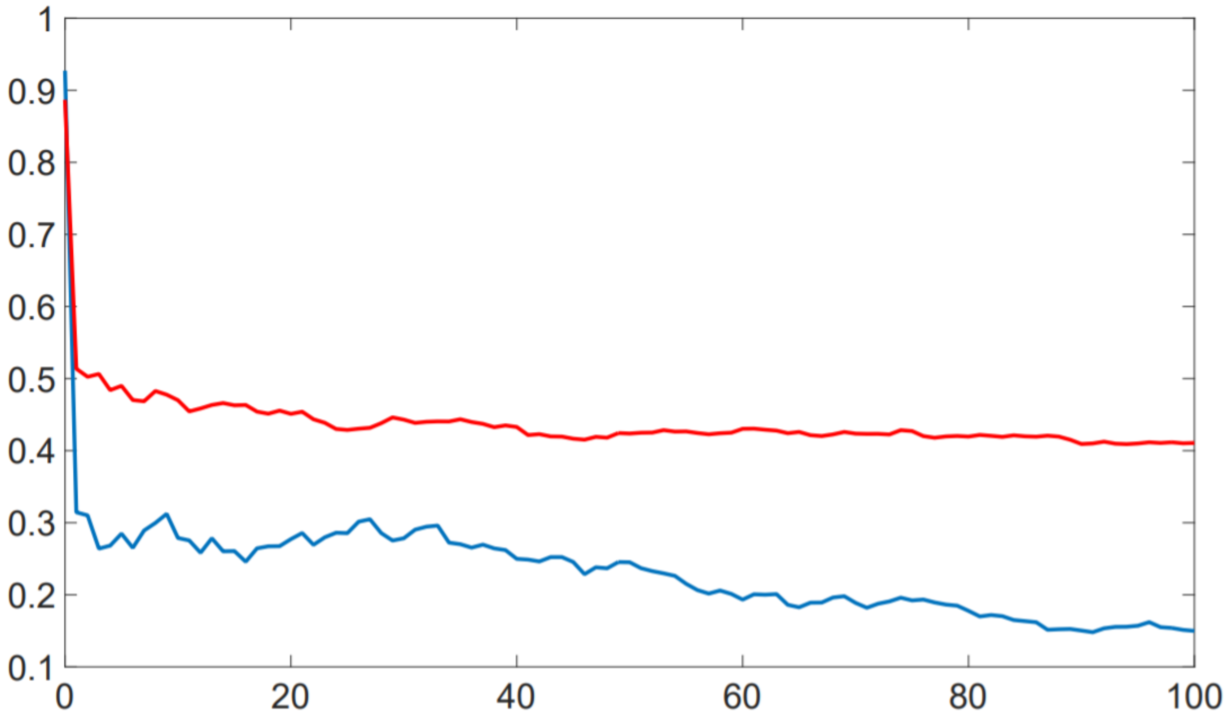
Отриманий результат

Граф	Кількість ітерацій	Час роботи алгоритму
a)	49985	14698 с
b)	8435	479 с
c)	38924	9644 с
d)	8204	504 с

Payoff-Based Algorithm (Алгоритм заснований на виграші)

З даних малюнків можемо бачити, що збіжність даного алгоритму до нуля досить повільна. Показані результати для N=10(блакитна лінія) та N=100(червона лінія).





Порівняння отриманих результатів

Можемо бачити, що робота алгоритмів досить сильно залежить від кількості гравців, виду графа, що показує їх взаємодію. Але, навіть незважаючи на це, бачимо, що алгоритм ADMM працює більш ефективно.

Висновки

Розподілений алгоритм по пошуку рівноваги Неша, розроблений і названий ADMM для досягнення більшої швидкості і надійності. Ми показали, що неточний алгоритм ADMM збігається. В кінцевому підсумку швидкість збіжності алгоритму можна порівняти зі швидкістю збіжності існуючих алгоритмів, заснованих на плітках по пошуку рівноваги Неша.

Також ми переглянули новий алгоритм по пошуку рівноваги Неша, що заснований на виграшах. Він також є збіжним, але по отриманим результат можемо бачити, що він не настільки ефективний.

Дані алгоритми можна використовувати з різними цілями, так як Теорія Ігор досить потрібна наука в житті. Її досягнення використовуються в різних науках, таких як в біології, соціології, політології, психології, економіці.

Теорія Ігор може бути корисна також і в повсякденному житті, а не тільки в науці. У кожного в житті бувають ситуації, в яких потрібно прийняти важливе рішення. Наша інтуїція часто може помилятися, тому вміння аналізувати інформацію та логічно мислити допоможе в житті. В цьому нам може допомогти теорія ігор та концепція рівноваги Неша.

Використані джерела

1. Мак-Кинси Дж., “Введение в теорию игр”, 2012

2. Петросян Л. А., Зенкевич Н. А., Шевкопляс Е. В. “Теория игр”, 2012, 432 с.

3. Jayash Koshal, Angelia Nedic, Uday V. Shanbhag, “Distributed Algorithms for Aggregative Games on Graphs”, DOI: 10.1287/opre.2016.1501, 2016

4. Farzad Salehisadaghiani, Lacra Pavel, “Distributed Nash Equilibrium Seeking via the Alternating Direction Method of Multiplier”, arXiv1707.01965, 2016

5. Maryam Kamgarpour, Tatiana Tatarenko, “Payoff-Based Approach to Learning Nash Equilibria in Convex Games”, arXiv:1611.10156v1, 2016