Шановні члени державної комісії, шановні присутні! До вашої уваги представляється до захисту дипломна робота на тему: «Методи розв'язування матричних ігор та їх програмна реалізація». Дипломна робота виконана на на мові програмування JavaScript.

(слайд 1)

Організації звичайно мають цілі, які суперечать цілям інших організацій-конкурентів. Тому робота менеджерів часто полягає у виборі рішення з урахуванням дій конкурентів. Для вирішення таких проблем призначені методи теорії ігор.

(слайд 2)

Теорія ігор - це розділ прикладної математики, який вивчає моделі і методи прийняття оптимальних рішень в умовах конфлікту.

Під конфліктом розуміється така ситуація, в якій зіштовхуються інтереси двох або більше сторін, що переслідують різні (найчастіше суперечливі) цілі. При цьому кожне рішення має прийматися в розрахунку на розумного суперника, який намагається зашкодити іншому учаснику гри досягти успіху.

(слайд 3)

З метою дослідження конфліктної ситуації будують її формалізовану спрощену модель. Для побудови такої моделі необхідно чітко описати конфлікт, тобто:

1. уточнити кількість учасників (учасники або сторони конфлікту називаються гравцями);
2. вказати на всі можливі способи (правила) дій гравців, які називаються стратегіями гравців;
3. розрахувати, якими будуть результати гри, якщо кожний гравець вибере певну стратегію (тобто з'ясувати виграші або програші гравців).

(слайд 4)

Основну задачу теорії ігор можна сформулювати так: визначити, яку стратегію має застосувати розумний гравець у конфлікті з розумним суперником, щоб гарантувати кожному з них виграш, при чому відхилення будь-якого з гравців від оптимальної стратегії може тільки зменшити його виграш.

Математична теорія ігор здатна не тільки вказати оптимальний шлях до вирішення деяких проблем, а й прогнозувати їх результат. Матричні ігри серйозно вивчаються фахівцями, так як вони досить прості і до них можуть бути зведені ігри загального виду. Тому теорія матричних ігор добре розвинена, існують різні методи пошуку розв'язку ігор.

В роботі розглядаються парні антагоністичні ігри, тобто такі, в яких приймають участь тільки два гравці, і виграш одного гравця рівний програшу іншого. Вважається, що кожен гравець має скінченну кількість чистих стратегій.

(слайд 5)

Розігрування матричної гри зводиться до вибору гравцем 1 i-го рядка матриці виграшів, а гравцем 2 - j-го стовпця. Після цього гравець 1 отримує виграш рівний a_{ij}, а гравець 2 - ($-a_{ij}$). При правильній грі гравець 1 може завжди гарантувати собі виграш, який назвемо нижнім значенням ціни гри. Позначимо його: $\underline{v} = \max_i \min_j a_{ij}$ (на слайді). У свою чергу, гравець 2 може гарантувати собі програш, який назвемо верхнім значенням ціни гри. Позначимо його: $\bar{v} = \min_j \max_i a_{ij}$ (на слайді). Чисті стратегії i* і j*, що відповідають $\underline{v}$ і $\bar{v}$ називаються максимінною і мінімаксною стратегіями.

(слайд 6)

Ситуація (i*, j*) називається ситуацією рівноваги, якщо для *i*Î1, 2, ..., m, *j*Î1, 2, ..., n виконується нерівність:.$a_{ij^*} \le a_{i^*j^*} \le a_{i^*j}$ Якщо ситуація рівноваги для гри існує, то така гра називається грою з сідловою точкою, а елемент $a_{i^*j^*}$ матриці А, що їй відповідає називається сідловою точкою цієї матриці. Ситуація рівноваги це така ситуація, від якої жодному з гравців не вигідно відхилятися.

(слайд 7)

Мішаною стратегією гравця називається повний набір ймовірностей застосування його чистих стратегій.

Так, якщо мішана стратегія гравця записана як (1/5, 0, 4/5), то це значить, що з п'яти розігрувать один раз він має вибрати першу стратегію, і чотири — третю. Такий підхід можна реалізувати, наприкоад, за методом пропорційної рулетки: розбити рулетку на 5 однакових секторів, 1 зафарбувати чорним кольором, 4 - білим. Якщо стрілка вкаже на білий сектор, то вибрати рішення 3, якщо на чорний - рішення 1.

(слайд 8)

Теорема (фон Неймана). У мішаному розширенні матричної гри завжди існує ситуація рівноваги.

Стратегії, яким відповідають однакові значення платіжної матриці (тобто матриця містить однакові рядки(стовпці)), називаються дублюючими. Якщо всі елементи і-го рядка (стовпця) платіжної матриці перевищують значення елементів j-го рядка (стовпця), то кажуть, що і-та стратегія гравця А (гравця В) є домінуючою над j-ю, а j-та називається домінованою і-ю.

Основні методи розв'зування матричних ігор

Алгоритм пошуку розв'язку матричної антагоністичної гри, заданої платіжною матрицею можна звести до алгоритму симплекс-методу розв'язання пари взаємодвоїстих задач лінійного програмування. Розглянемо метод, за допомогою якого можна привести скінченну матричну антагоністичну гру до двох взаємодвоїстих задач лінійного програмування.

(слайд 9)

Часто в практичних задачач немає необхідності знаходити точний розв'язок матричної гри. Досить знайти наближений розв'язок, який дає середній виграш, близький до ціни гри і наближені оптимальні стратегії гравців.

Орієнтовне значення ціни гри може дати вже простий аналіз матриці виграшів та визначення нижньої і верхньої ціни гри. Якщо вони близькі, то пошуками точного розв'язку займатися не обов'язково, так як досить вибрати чисті мінімаксну та максимінну стратегії. Якщо ж вони не близькі, можна отримати прийнятний для практики розв'язок за допомогою чисельних методів розв'язання ігор, за допомогою одного з чисельних методів, наприклад, за допомогою методу Брауна-Робінсона.

(слайд 10)

Монотонний ітеративний алгоритм розв'язання матричних ігор

Цей алгоритм реалізується тільки для одного гравця на відміну від методу Брауна-Робінсона, який працює для двох гравців. Алгоритм дозволяє знаходити точно і наближено оптимальну стратегію гравця 1 і значення ціни гри n. За допомогою цього алгоритму можна отримати задану точність розв’язку, причому число кроків, необхідних для досягнення результатів, слабо залежить від розмірності матриці виграшів.

Особливість цього алгоритму у здатності генерувати строго монотонно зростаючу послідовність оцінок ціни гри, що не властиво алгоритму Брауна-Робінсона.

(слайд 11)

У 1994 році Нобелівська премія з економіки була присуджена Джону Нешу «за аналіз рівноваги в теорії некоаліційних ігор». До виходу в світ дисертації Джона Неша дослідження проводились основному в області ігор з нульовою сумою, де сумарний виграш одних учасників дорівнює сумарному програшу інших. Неш досліджував гри з ненульовою сумою, в яких можливі ситуації, вигідні відразу для всіх учасників. Так, наприклад, у разі суперечки керівництва заводу і профспілки про підвищення заробітної плати вона може закінчитися страйком, що не вигідно відразу обом учасникам суперечки, тоді як від певної угоди всі учасники конфлікту можуть в цілому залишитися у виграші. Неш виділив спеціальний клас ситуацій в грі (тобто фіксованих стратегій всіх учасників гри) - такі ситуації, від яких не вигідно відхилятися поодинці нікому з гравців. Пізніше такі ситуації були названі ситуаціями рівноваги по Нешу.

В чистих стратегіях в грі може не існувати ситуації рівноваги. Орієнтація на оптимальні мішані стратегії дозволяє першому гравцеві максимізувати найменший очікуваний виграш, а другому мінімізувати найбільший очікуваний програш. Однак такий вибір може бути виправданий лише для багаторазово повторюваних ігор. В окремій грі кожен з гравців навіть при реалізації «оптимальної» мішаної стратегії може дуже сильно програти. В окремих іграх мішані стратегії не тільки не обіцяють максимального виграшу, але навіть не захищають від максимально можливого програшу, тобто кожен гравець може прийняти найгірше з усіх можливих рішень!

Вибір максимінної і мінімаксної стратегій добре підходить для обережних гравців. Він дозволяє не програти занадто багато, не виграти занадто мало. Але деякі гравці можуть бути більш схильні до ризику, і орієнтуватися на ті стратегії, які можуть забезпечити їм максимально можливий виграш, деякі намагаються максимізувати середній очікуваний виграш, тобто вибір мінімаксу або максиміну не є єдиноможливим хорошим в деякому сенсі рішенням.

Нарешті, орієнтація на мішані стратегії передбачає відсутність будь-якої інформації про психологію суперника, припускаючи лише, що він діє раціонально.

На практиці ми можемо мати про противника будь-яку додаткову інформацію і на її основі робити які-небудь припущення щодо його поведінки. При цьому наші припущення практично завжди носять нечіткий характер. Приміром, наші припущення можуть мати такий вигляд: «якщо противник самовпевнений, то, скоріше за все, він вибере стратегію максимізації максимального доходу».

(слайд 12)

Дипломна робота повністю реалізована на JavaScript. Це дає змогу розширити кількість пристроїв, які зможуть виконувати програму без потреби перекомпіляції. Поділяється на клієнтську та серверну частини.

В якості бази для проектування інтерфейсу було використано фреймворк EnyoJS. Це досить молодий проект. Використовувався в розробці інтерфейсу WebOS.

(слайд 13)

Кожна функція знаходиться в окремому файлі і підключається в основному модулі.

Якщо розміри матриці невеликі, то можна обрахувати результат прямо в браузері і не використовувати інтернет-з'єднання для цього. Методи роботи з нечіткими множинами реалізовані тільки на клієнтській частині.

(слайд 14)

В випадку, кому матриця має значні розміри має сенс ввести її в текстовий файл і відіслати на сервер (матричну гру, введену в браузері теж можна порахувати на сервері). На сервері реалізовано три основних методи розв'язку матричних ігор: графічний метод, метод Брауна-Робінсона та симплекс-метод. Дані кодуються в формат JSON та передаються на сервер за протоколом WebSocket.

При введені матричної гри в файл, немає змоги вирішити її в браузері.

Серверна частина працює під платформою nodejs. Я обрав саме цю платформу, тому що це дало змогу писати на одній мові на клієнті та на сервері. Також як сервер може використовуватися будь-який дистрибутив ОС Linux, Windows, FreeBSD та деякі інші операційні системи. До стандартного набору модулів nodejs потрібно було довстановити node-o3-canvas та socket.io.

При обрахунках як на сервері так і в браузері — формат виводу не змінюється.

Задачі, що зводяться до матричних ігор зустрічаються в багатьох аспектах життя, особливо в економіці. Багато вчених проводили дослідження з оптимізації розв'язку матричних ігор. Особливо віділяються методи приведення до задачі лінійного програмування та ітеративні методи, такі як метод Брауна-Робінсона та монотонний ітеративний алгоритм. Прості ігри виду 2*n та m*2 можна розв'язувати графічно.

(слайд 15)

Розглянемо, як приклад розв'язання такої матричної гри (на слайді).

Після введення завдання в поле вводу матриці обирається метод і місце проведення обрахунків.

(слайд 16)

Результат рішення симплекс-методом на слайді.

Потім можна обрати метод Брауна-Робінсона

(слайд 17)

Результат рішення методом Брауна-Робінсона на слайді.

Для демонстрації графічного методу введемо матрицю розмірності 2 на 4

(слайд 18)

Спочатку преревіримо роботу програми двома наведеними вище методами

(слайд 19)

Симплекс-метод

(слайд 20)

Метод Брауна-Робінсона

(слайд 21)

При виборі графічного методу маємо графік та значення мішаних стратегій гравців та ціни гри.

(слайд 22)

Особливістю моєї програми є функція знаходження мішаної стратегії на основі деяких відомостей про супротивника.

Спочатку вводиться матриця виграшів/програшів.

Потім вводиться база правил, якими (за нашим припущенням) керується супротивник.

(слайд 23)

Так це виглядає в браузері.

Всі правила за замовчуванням розділяються умовою "або". Але можна ввести умову "і", в тому впадку, якщо на вибір певного методу впливає одночасно 2 правила.

(слайд 24)

Результат підрахунків. На нього впливає наша сумарна впевненість в виборі упротивником того чи іншого методу вибору оптимальної стратегії.

(слайд 25)

В даній дипломній роботі було реалізовано основні алгоритми розв’язування матричних ігор. А саме симплекс-метод, метод Брауна-Робінсона та графічний метод.

(слайд 26)

Також було реалізовано алгоритм, що при обчислені враховує прогнозування поведінки супротивника на основі попередній відомостей про нього. Цей алгоритм будує змішану стратегію, в якій враховано степінь впевненості в виборі супротивником кожної своєї стратегії.

(слайд 27)

Програма має можливість проводити складні обчислення на стороні сервера, що значно зменшує навантаження на пристрій. Це особливо актуально у випадку планшету чи мобільного телефону. Було протестовано на планшеті HP TpuchPad з операційною системою WebOS та браузером на основі WebKit та телефоні HTC Dezire Z з операційною системою Android та браузером на основі WebKit.

На цьому мій виступ підійшов до завершення. Дякую за увагу!