Міністерство освіти і науки України
Національний тейхнічний університет України «Київський політехнічний інститут»
Інститут прикладного та системного аналізу
Кафедра математичного моделювання системного аналізу

Дипломна робота на тему:

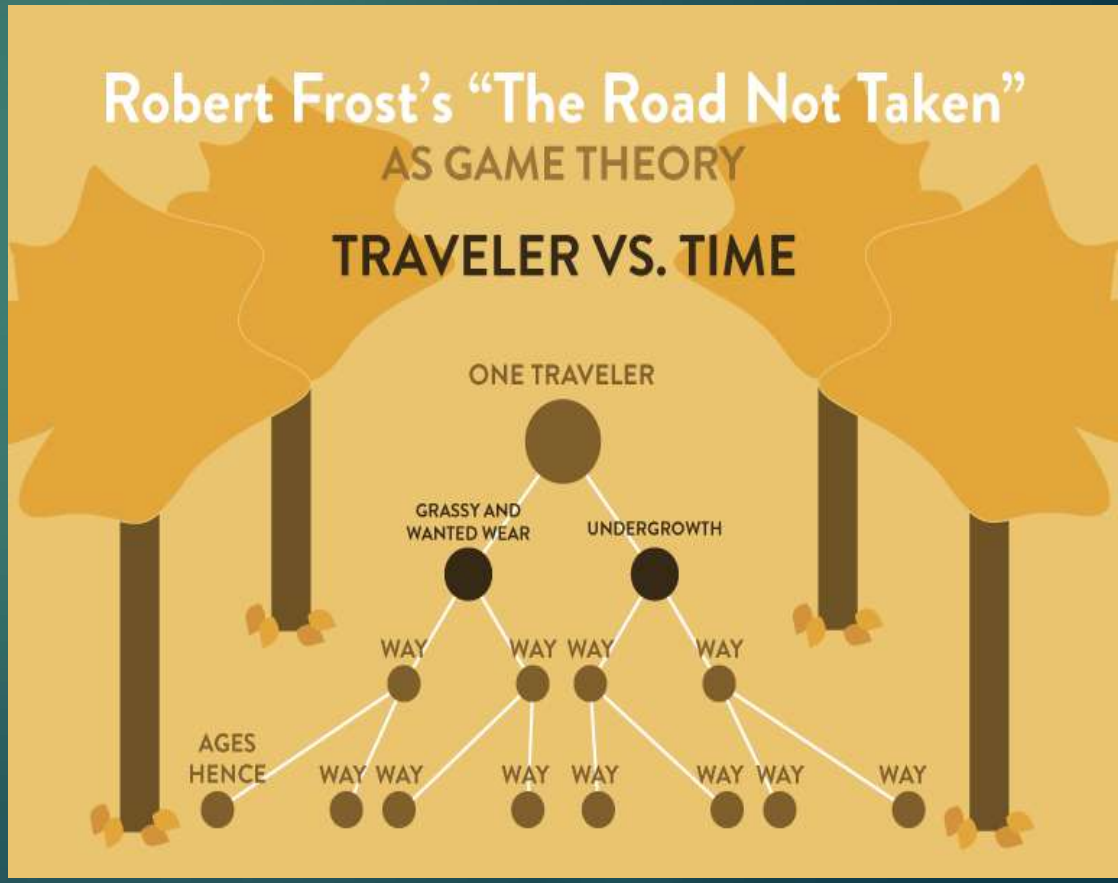
Інтелектуальні системи прийняття рішень для ігор з неповною інформацією

Виконав студент групи КА-33
Скляр А.В.
Науковий керівник:
к.т.н Дідковська М.В.

Актуальність роботи

Пошук оптимального рішення в ситуаціях коли людина не володіє повною інформацією про об'єкт дослідження, його поведінку. Це може бути застосовано в практично усіх сферах життя

- Військова справа
- Комп'ютерні науки
- Економіка
- Політика
- Бізнес



Постановка задачі

- Дослідити сучасні підходи до прийняття рішень в іграх з неповною інформацією
- Дослідити існуючі алгоритми для роботи з іграми з неповною інформацією на прикладі покеру
- Запропонувати і реалізувати програмний продукт, що буде демонструвати роботу обраного алгоритму
- Провести експериментальне дослідження розробленої системи

Чому саме покер?

- Надзвичайно висока складність гри ($9.17 * 10^{17}$ станів гри та $3.19 * 10^{14}$ інформаційних станів).
- Через неповноту інформації, покер потребує володіння не тільки логікою, а й моделями поведінки (зокрема, вмінням блефувати), які до сьогоднішнього дня вважаються занадто складними для аналізу штучним інтелектом.
- Покер дозволяє створити потужний інструмент для прийняття рішень в непередбачуваних ситуаціях і надає можливість перевірити ефективність цього інструменту.

Існуючі рішення

- Libratus (створений Carnegie Mellon University)
  - Базується на модифікованому алгоритмі CFR+, на сьогоднішній день вважається найкращим агентом з гри в покер, що було підтверджено на щорічному турнірі ACPC.
  - Унікальність полягає в методі пошуку рішень – замість нейронних мереж, які зазвичай використовуються у зв'язці з CFR, шукає рішення використовуючи техніку Reinforcement Learning.
- DeepStack (створений University of Alberta Poker Research Group)
  - Базується на поєднанні алгоритму CFR з глибинними нейронними мережами, що дозволяє виконувати пошук вшир, а не в глибину, тим самим обмежуючи кількість можливих розв'язків на кожному кроці.
  - Вважається першим агентом, який переміг в команди діючих чемпіонів з покеру.

Існуючі алгоритми

- Жаль (regret)
  - Жаль – це концепція навчання в онлайн режимі, яка є базою для цілого сімейства потужних алгоритмів навчання.
  - Середня загальна жаль гравця i в час T:

$$R_i^T = \frac{1}{T} \max_{\sigma_i^* \in \Sigma_i} \sum_{t=1}^{T} (u_{i(\sigma_i^*, \sigma_{-i}^t)} - u_i(\sigma^t))$$

  - Середня стратегія для гравця i з часу 1 до T $\overline{\sigma_i^t}$ для кожного інформаційного набору $I \in \mathcal{L}_i$, для кожного $a \in A(I)$:

$$\overline{\sigma_i^t}(I)(a) = \frac{\left(\sum_{t=1}^{T} \pi_i^{\sigma^t}(I)\sigma^t(I)(a)\right)}{\sum_{t=1}^{T} \pi_i^{\sigma^t}(I)}$$

- Теорема. У грі з нульовою сумою в часі T, якщо середній жаль обох гравців менше деякого ϵ, то $\overline{\sigma_i^t}$ – 2ϵ рівновага.

Існуючі алгоритми

- Алгоритм мінімізації жалю протидії (Counterfactual Regret Minimization)

  - Корисність протидії $u_i(\sigma, I)$ визначається наступним чином:

$$u_{i(\sigma,I)} = \frac{\sum_{h \in I, h' \in Z} \pi^{\sigma}_{-i}(h)\pi_i^{\sigma(h,h')u}(h')}{\pi^{\sigma}_{-i}(I)},$$

де $\pi^{\sigma}(h, h')$ - ймовірність переходу від історії h до історії h '.

  - Безпосередній жаль протидії розраховується за формулою:

$$R^{T}_{i,imm}(I) = \frac{1}{T} \max_{a \in A(I)} \sum_{t=1}^{T} \pi^{\sigma^t}_{-i}(I)(u_i(\sigma^T|_{I \to a}, I) - u_i(\sigma^t, I))$$

$$R^{T,+}_{i,imm}(I) = \max(R^{T}_{i,imm}(I), 0)$$

Існуючі алгоритми

- Алгоритм мінімізації жалю протидії (Counterfactual Regret Minimization)
- Теорема. $R_i^T \leq \sum_{I \in L_i} R_{i,imm}^{T,+}(I)$
  - Тоді можна порахувати стратегію на кроці T+1 наступним чином:

$$\sigma_i^{T+1}(I)(a) = \begin{cases} \dfrac{R_i^{T,+}(I,a)}{\sum_{a \in A(I)} R_i^{T,+}(I,a)}, \text{якщо} \sum_{a \in A(I)} R_i^{T,+}(I,a) > 0 \\ \dfrac{1}{|A(I)|}, \text{в іншому випадку} \end{cases}$$

- Теорема. Якщо гравець і вибирає дії згідно з останнім рівнянням, то

$$R_{i,imm}^T(I) \leq \Delta_{u,i}\sqrt{|A_i|}/\sqrt{T}$$

і, відповідно,

$$R_i^T \leq \frac{\Delta_{u,i}|L_i|\sqrt{|A_i|}}{\sqrt{T}}, \text{где } |A_i| = \max_{h:P(h)=i} |A(h)|$$

Модифікований алгоритм Pure CFR

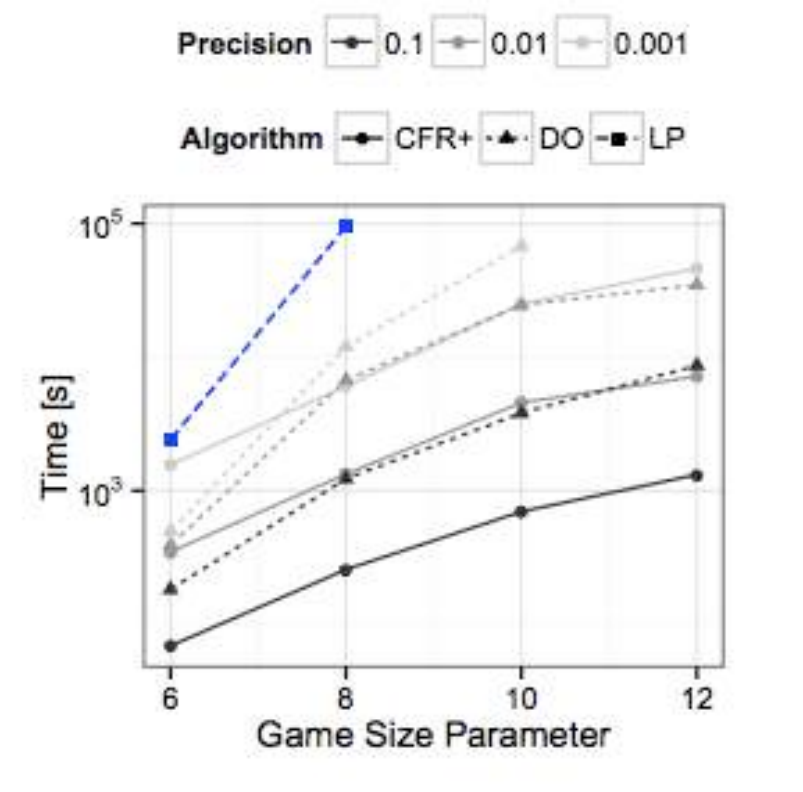
- Головна відмінність Pure CFR полягає в тому, що, на відміну від CFR, Pure CFR на кожному кроці використовує чисту стратегію.

- Більш формально, з огляду на наш поточний профіль стратегії $\sigma = \sigma^t$, ми вибираємо $\hat{s}$ з σ, незалежно визначаючи одну дію для кожного інформаційного набору I.

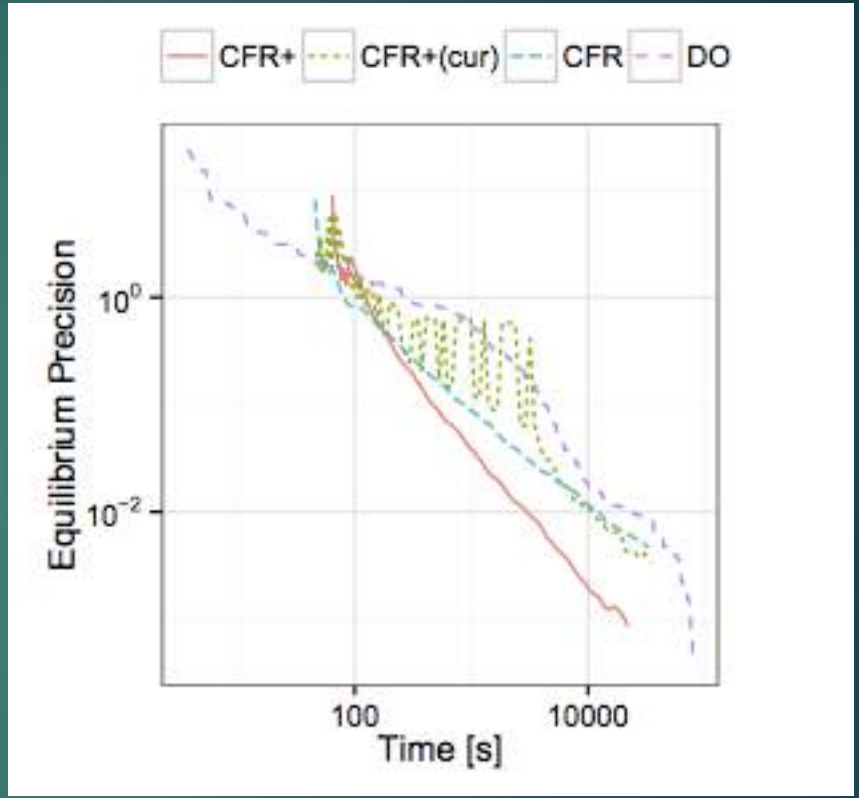
- Оцінене значення жалю супротиву в інформаційному наборі I і для стратегії σ для чистого CFR визначається як:

$$\hat{v}_i(I, \sigma) = \sum_{z \in Z_I} u_i(z) \pi^{\hat{s}}_{-i}(z[I]) \pi^{\hat{s}}(z[I], z)$$

Результати порівняння алгоритмів



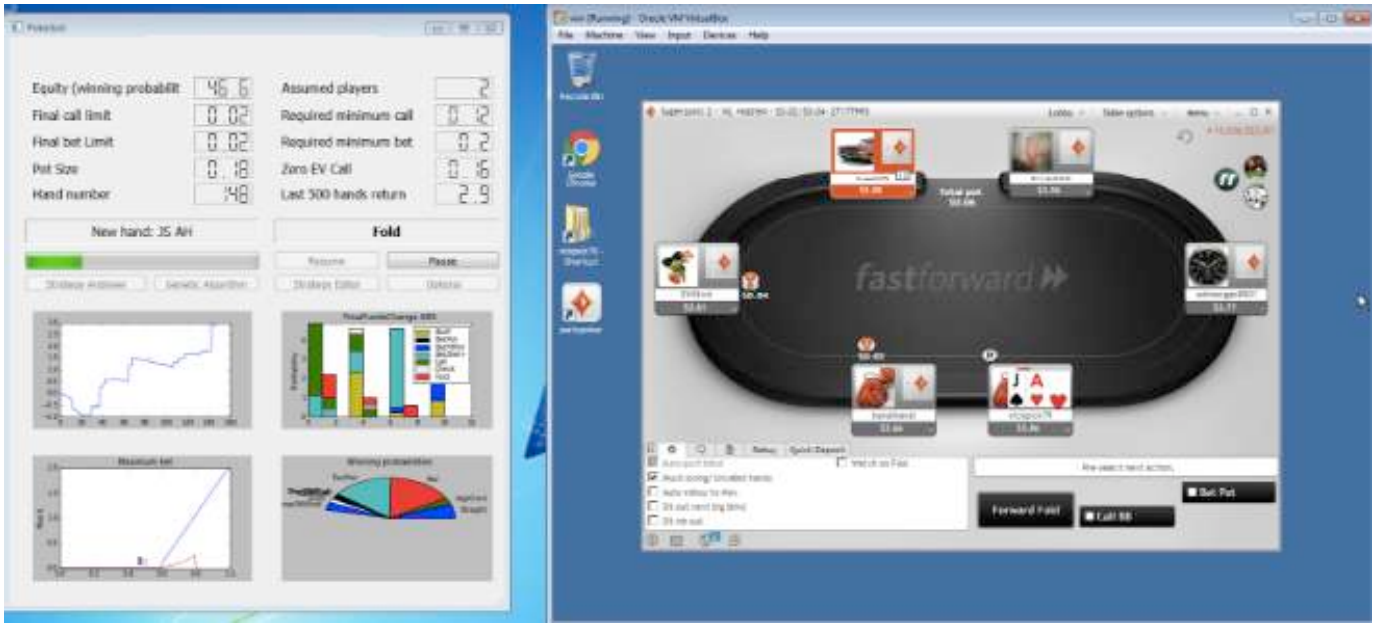
Час тренування алгоритму Pure CFR (CFR+) порівняно з класичними CFR



Точність знайденої рівноваги в порівнянні із класичними CFR алгоритмами

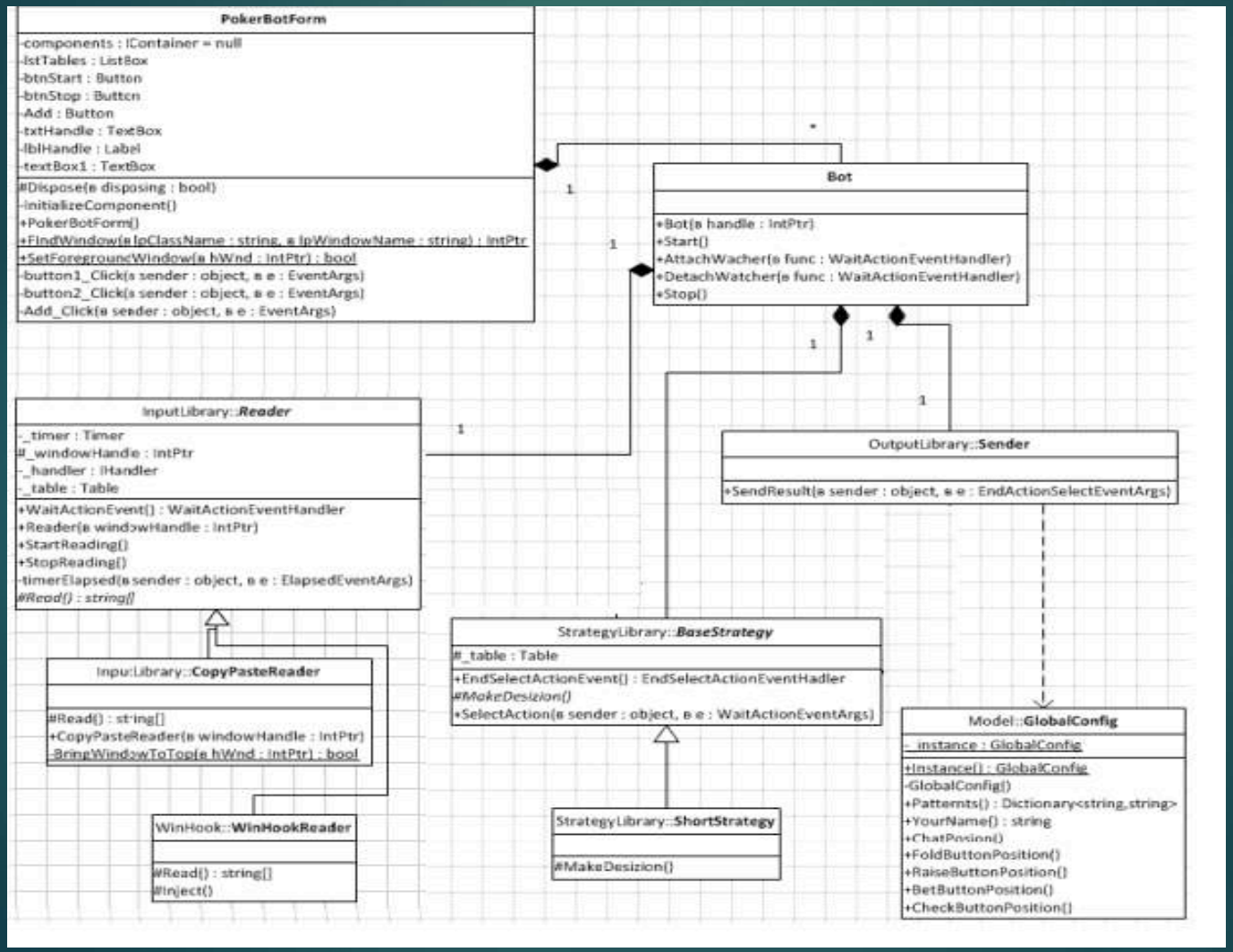
Аналіз результатів

- Алгоритм Pure CFR демонструє значно кращі результати, аніж класичний CFR.
- Окрім того, Pure CFR використовує менше обчислювальних ресурсів аніж класичний CFR.
- Pure CFR потребує більше ітерацій для навчання, ніж CFR і аналогічні алгоритми, але при цьому час навчання є меншим через нижчу складність обчислень.


Equity (winning probabilit
Final call limit
Final bet Limit
Pot Size
Hand number
Assumed players
Required minimum call
Required minimum bet
Zero EV Call
Last 500 hands return
New hand: JS AH
Fold
fastforward

DEMO TIME

Архітектура програмного продукту



Висновки

- Новизна:
  - ✓ Проведено порівняльний аналіз як існуючих, так і модифікованих алгоритмів
  - ✓ Були визначені основні переваги та недоліки існуючих підходів
  - ✓ Запропоновано архітектуру програмного продукту
- Цінність:
  - ✓ Реалізована програма, яка демонструє роботу обраного алгоритму, може бути використана пересічним користувачем, оскільки не потребує додаткових налаштувань і працює з ефективністю 86%

Подальші шляхи розвитку

- Застосувати принципи, які використовує DeepStack до Pure CFR.
- Реалізувати розпізнавання символів для можливості гри в онлайн покерних системах.
- Пристосувати систему до вирішення ігор з неповною інформацією, відмінних від покеру, та ситуацій, які можуть бути змодельовані за допомогою таких ігор.
- Реалізувати можливість використання існуючих даних для подальшого тренування агента.
- Проаналізувати питання інтеграції з апаратною частиною.

Дякую за увагу!