

РОЗРОБКА ТА ОПТИМІЗАЦІЯ БОЙОВОГО ІІІ В КОЛЕКЦІЙНІЙ КАРТКОВІЙ ГРІ

Потурайко Н.І.

e-mail: nazar.poturaiko@nure.ua

Науковий керівник – ст. викладач Новіков Ю.С.

Харківський національний університет радіоелектроніки, каф. ПІ
м. Харків, Україна

This paper examines the problem of developing combat artificial intelligence for a collectible card game, considering stochasticity and limited resources. The main approaches to building AI have been analyzed, and their advantages and disadvantages in the context of CCG have been discussed. The validity of using Expectimax as the basic algorithm has been substantiated. Suggested an optimization system based on Star2 for branch cutting, Transposition Tables for state caching, and Zobrist Hashing for quick position identification.

Колекційні карткові ігри характеризуються високим рівнем прихованої інформації, кількості ігрових станів та стохастичності. Важливою частиною у ККІ є бої які є обов'язкові для прогресії, штучний інтелект у яких має давати достатній виклик та підігрівати цікавість до дослідження нових комбінацій та знаходження нових карток. Недоліком класичних алгоритмів є неповне забезпечення приймання рішень у реальному часі та високі витрати ресурсів. Метою є аналіз підходів та вироблення власного з його комплексною оптимізацією.

Детальний аналіз класичних підходів виявив їх обмеженість для використання у стохастичних іграх до яких і відноситься ККІ. Евристичний підхід хоча і забезпечує просту реалізацію та високу швидкодію, але позбавлений будь-якої тактичної глибини не маючи змогу прогнозувати наступний хід суперника. Алгоритм Minimax знаходить складні тактики в декілька ходів, проте не може працювати в умовах стохастичності. Monte Carlo Tree Search (MCTS) забезпечує високу точність ходів, неймовірно складні тактики, але витрачає надмірну кількість ресурсів та складно підлаштовується під рівні складності. Найкращим варіантом було визначено Expectimax, що є модернізацією Minimax з вузлами випадковості перекриваючи головний недолік попередника.

Оцінка поточного ігрового стану S в базовому алгоритмі Expectimax обчислюється як математичне сподівання можливих результатів випадкових подій – витягування картки, критична атака тощо. Оцінка вузла виглядає так:

$$V(S) = \sum_{i=1}^n P(s_i) \cdot V(s_i)$$

де $P(s_i)$ – ймовірність настання події s_i , а $V(s_i)$ – оцінка корисності стану після цієї події.

З цього випливає головний недолік експоненціально зростаючий розмір дерева рішень $O((b \times c)^d)$, де b – кількість можливих ходів, c – кількість випадкових подій, d -- глибина пошуку. Для подолання цієї фундаментальної проблеми було запропоновано новий для ККІ комплекс рішень.

Першим кроком буде застосування техніки відсікання гілок Alpha-Beta адаптованої під Expectimax з назвами Star1 та Star2. Основна різниця для нашого контексту між 1 та 2 полягає у тому, що перший алгоритм працює при нерівномірних ймовірностях, а другий при виключно рівномірних. Оскільки програмна реалізація, де буде використаний алгоритм, має рівномірні шанси то Star2 буде значно краще. Це більш агресивна версія першого, яка швидко перевіряє декілька можливих з усіх варіантів, та якщо результат поганий відсікає гілку повністю в результаті зекономивши ресурси та час.

Наступна оптимізація полягає у тому, що однакова позиція може розіграватись декількома способами витрачаючи зайві ресурси. Для усунення цього буде використано Transposition Tables. Логіка роботи полягає у зберіганні обчисленої позиції разом з глибиною пошуку. Коли алгоритм знайде таку саму позицію, то замість витрачання ресурсів на повторне обчислення подальших ходів, просто поверне уже знайдену, скоротивши час обчислення та зменшуючи кількість вузлів. Для ще більш ефективної роботи Transposition Table необхідно використати швидкий та надійний спосіб ідентифікації станів - Zobrist Hashing. Він генерує хеш для кожної сутності гри бітове число, що потім формується побітовою операцією XOR ($\oplus$):

$$H = v_1 \oplus v_2 \oplus \dots \oplus v_n$$

При зміні стану у майбутньому хеш оновлюється інкрементально, повторно застосовуючи операції до змінених елементів. Це дозволяє отримати високу швидкість доступу до таблиць транспозицій та уникнути повного перерахунку станів.

Для оцінки ефективності оптимізації комплексної моделі алгоритму був розроблений ізольований математичний модуль у якому симулюється бій, випадковий вибір суперника та карток та типові правила гри. Вхідні дані звичайні, глибина пошуку 8. Результати отримані у ході обчислень були занесені у таблицю 1.

Таблиця 1 – Результати симуляції ККІ з використання алгоритму та різної кількості оптимізації

Конфігурація алгоритму	Кількість обчислених вузлів	Час на обчислення (мс)
Expectimax	98247661	57267
Expectimax + Star2	12050146	9472
Expectimax + Star2 + Zobrist Hashing + Transposition Tables	127161	232

З результатів обчислення у Таблиці 1 видно, що використання неоптимізованого Expectimax є неприйнятним для ігор у реальному часі витративши майже одну хвилину на хід. Star2 хоча і дав задовільний результат, але використання усіх оптимізацій дало кратно кращий результат підтверджуючи теоретичні розрахунки.

В результаті дослідження було вирішено типічну проблему високих обчислювальних витрати бойового ІІ в стохастичних ККІ. Доведено, що класичні алгоритми у повній мірі не здатні забезпечити прийняття оптимальних рішень без витрати великої кількості часу при цьому зберігаючи високу точність ходів. Запропоноване нове рішення у контексті стохастичних ККІ, включаючи в собі комплексну оптимізацію оптимізацію алгоритма Expectimax. Отримана модель ефективно показала себе у симуляції зі сторони ресурсів підтверджуючи теоретичні розрахунки.

Список використаних джерел:

1. Мельник А. О., Керницький А. Б. Дослідження методів штучного інтелекту в іграх з неповною інформацією та елементами випадковості. Системи керування, навігації та зв'язку. Збірник наукових праць. 2021. Вип. 3. С. 102–107.
2. Гаврилов О. В., Козловський В. В. Оптимізація алгоритмів пошуку в ігрових деревах для стохастичних ігор. Інформаційні технології та комп'ютерна інженерія. 2022. № 1. С. 45–52.
3. Suryana N., Setyanto A. Efficiency of Expectimax Algorithm with Alpha-Beta Pruning in Stochastic Game Environments. International Journal of Computer Science and Information Security. 2023. Vol. 21, No. 4. P. 112–119.
4. Бондаренко М. С., Сидоров О. В. Використання хешування Зобріста та таблиць транспозицій для прискорення інтелектуальних агентів у настільних стратегіях. Комп'ютерні науки та інформаційні технології. 2020. Т. 5, № 2. С. 88–94.
5. Логвін І. Ю., Новіков Ю. С. Дослідження та використання підходів створення ігрового ІІ для реалізації поведінки монстрів в ігрових додатках. Радіoeлектроніка та молодь у ХХІ столітті : матеріали ХХVІІ Міжнар. молодіж. форуму (Харків, 2023 р.). Харків : ХНУРЕ, 2023. [С. 1–2].