

## ПОРІВНЯННЯ АЛГОРИТМІВ МІНІМАКС, МОНТЕ-КАРЛО І АЛЬФА-БЕТА ВІДСІЧЕННЯ НА ПРИКЛАДІ ГРИ ХРЕСТИКИ-НУЛИКИ

*Донецький національний університет імені Василя Стуса, м. Вінниця*

Ігри це вид діяльності спрямований на відтворення інших видів діяльності з метою навчання або розваг. Ігри характеризуються повторюваністю та напрацюванням навичок, рефлексів. Вивченням ігор з математичним підходом займається теорія ігор [1]. Теорія ігор – це наука, що вивчає рішення гравців, їх взаємодію (як співпрацю, так і їх суперництво) залежно від прийнятих іншими гравцями рішень. Об'єктом вивчення теорії ігор є аналіз прийняття рішень, причому не важливо чи є ці дії виграшними для гравця чи програшними. Теорія ігор активно використовується в програмуванні для передбачення можливого розвитку певних подій, розрахунку оптимального рішення, для створення опонента у стратегічних іграх. Теорія ігор формулює правила для формування певних рішень. Для розрахунку оптимальних ходів були створені алгоритми, що розглядають і аналізують можливі ходи гравців. Розглянемо та порівняємо такі алгоритми на прикладі гри хрестики-нулики.

Хрестики-нулики – це популярна логічна гра [2]. В гру можуть грати два гравці, які по черзі малюють символи X або O в одну з комірок поля розміром 3 на 3. Гравець, який першим поставив 3 однакових знаки в горизонтальному, вертикальному чи діагональному ряду, виграє партію. Для знаходження найкращого кроку в грі зазвичай використовуються дерева рішень і алгоритми пошуку у деревах. Можна використати декілька різних алгоритмів - мінімакс, мінімакс з альфа-бета відсіченням та дерево пошуку Монте-Карло.

В основі цих алгоритмів лежать дерева і алгоритми пошуку у деревах. Дерева використовуються для зберігання, аналізу і впорядкування інформації. Алгоритми пошуку у деревах допомагають швидко отримати доступ до такої інформації. Дерева і пошук у деревах використовуються у навчанні штучного інтелекту шляхом формування дерев рішень і у подальшому пошуку оптимальних рішень штучним інтелектом. Пошук у деревах використовується для знаходження найкращого можливого ходу в стратегічній грі.

Згідно з останніми дослідженнями, вчені зосередились на оптимізації, покращенні оптимальності і точності рішень існуючих алгоритмів. Кожен з алгоритмів пошуку у деревах має свої недоліки і переваги, таким чином

об'єднавши декілька способів пошуку у деревах у один алгоритм, можна досягти кращих результатів, аніж при використанні цих алгоритмів окремо. Наприклад, у статті “Monte-Carlo Tree Search and Minimax Hybrids with Heuristic Evaluation Functions” розглядається використання гібридних алгоритмів на прикладі гри “Connect Four” [3]. Автори розглядають звичайні алгоритми пошуку мінімакс і дерево пошуку Монте-Карло. Згодом, автори порівнюють їх з гібридним алгоритмом пошуку, що поєднує ці алгоритми. В результаті експериментів, виявилось, що гібридний алгоритм значно більш ефективний і швидкий аніж традиційні алгоритми пошуку.

Метою дослідження стане вивчення алгоритмів пошуку у деревах, їх аналіз і порівняння. Необхідно представити алгоритми пошуку, пояснити яким чином вони працюють, порівняти їх між собою і підсумувати інформацію.

Мінімакс – правило, що використовується для формування максимально виграшного для гравця рішення. Алгоритм мінімакс – алгоритм, що використовується для отримання максимальної користі від певного ходу і навпаки мінімізації можливих збитків. Алгоритм рекурсивно від кінцевого ходу шукає найкращий хід. Визначення кращого і гіршого ходу задається за певною умовою, наприклад, виграшний хід принесе гравцю X 10 очків, в той же час гравець O втратить десять очків через поразку. У разі нічії, ніхто нічого не отримує і не втрачає. Перевагою алгоритму є точність і оптимальність рішень. Оскільки мінімакс розглядає всі можливі перебіги подій, у комплексних задачах з великою кількістю варіантів перебігу подій, використання алгоритму у таких випадках не є раціональним рішенням, через низьку швидкість. Алгоритм мінімакс має складність  $O(n^k)$ , де  $n$  - всі можливі ходи,  $k$  - глибина дерева. Був розроблений алгоритм альфа-бета відсічення, який мав на меті нівелювати цей недолік.

Алгоритм мінімаксу з альфа-бета відсіченням — алгоритм пошуку що діє аналогічно мінімаксу, але припиняє аналіз ходу, якщо цей хід менш вигідний, аніж попередньо досліджений. Складність алгоритму альфа-бета відсічення -  $O(n^{(k/2)})$ , в найгіршому випадку складність алгоритму співпадає зі складністю звичайного мінімакса. Алгоритм мінімаксу з альфа-бета відсіченням швидший і потребує менше пам'яті, аніж звичайний мінімакс. Але, на відміну від звичайного мінімаксу, алгоритм мінімакс з альфа-бета відсіченням може давати некоректний результат, коли важливий порядок ходів у грі. Також алгоритм може пропустити оптимальний варіант, що знаходиться на менш глибокому рівні дерева, якщо гілки рівно оцінені за оптимальністю. Алгоритм мінімаксу з альфа-бета відсіченням швидший аніж дерево пошуку Монте-Карло у невеликих задач, але зі зростанням складності, алгоритм стає менш релевантним ніж алгоритм Монте-Карло.

Монте-Карло — підхід до пошуку оптимального рішення шляхом симуляції даних і їх аналізу [5]. Дерево пошуку Монте-Карло – алгоритм, що використовується для знаходження швидкого оптимального рішення. Алгоритм багаторазово запускає гру, випадково вибирає ходи, проходить до кінця і зберігає підсумок. Дерево пошуку Монте-Карло аналізує ходи і обирає

хід з найкращим результатом. На основі досліджень, алгоритм генерує оптимальне рішення, причому чим більша кількість ітерацій, тим більша оптимальність і точність алгоритму. Алгоритм не має чіткої складності, складність алгоритму залежить від кількості симуляцій, які необхідно виконати. Перевагою дерева пошуку Монте-Карло є велика швидкість роботи, яка досягається за рахунок розгляду лише найбільш перспективних розгалужень. Іншою перевагою алгоритма є можливість роботи алгоритма із задачами з невизначеністю і випадковими подіями. Недоліком алгоритму дерева пошуку Монте-Карло є те, що на початку існують вигідні ходи, за якими алгоритм вірогідно будуватиме свою стратегію, але у перспективі такі кроки виявляються помилковими й призведуть до поразки. Іншим недоліком є те, що для досягнення дійсно оптимальних і якісних рішень, необхідно провести велику кількість симуляцій, що займе багато часу і ресурсів. Якщо можливі ходи обмежені, алгоритми мінімаксу і мінімаксу з альфа-бета відсічення можуть впоратись із задачею швидше, ніж дерево пошуку Монте-Карло.

Підсумовуючи дослідження, кожен з алгоритмів пошуку у деревах має свої переваги і недоліки і буде корисним у певній ситуації. Алгоритм мінімаксу у результаті дасть оптимальний результат, однак у більш складних задачах буде повільним і використовуватиме велику кількість пам'яті. Алгоритм мінімаксу з альфа-бета відсіченням швидше впорається з задачею, але може дати некоректну відповідь у певних ситуаціях і буде повільнішим у великих задачах ніж алгоритм Монте-Карло. Дерево пошуку Монте-Карло у більшості випадків знаходить раціональне рішення, але потребує попередню симуляцію, причому для більшої точності необхідно провести більшу кількість симуляцій, що займе багато часу.

### Список літератури

1. *Теорія* igor  
<https://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BE%D1%80%D1%96%D1%8F%D1%96%D0%B3%D0%BE%D1%80> Дата доступу 11.05.2023
2. *Хрестики-нулики*  
<https://uk.wikipedia.org/wiki/%D0%A5%D1%80%D0%B5%D1%81%D1%82%D0%B8%D0%BA%D0%B8-%D0%BD%D1%83%D0%BB%D0%B8%D0%BA%D0%B8> Дата доступу 11.05.2023
3. *Monte-Carlo Tree Search and Minimax Hybrids with Heuristic Evaluation Functions*  
[https://link.springer.com/chapter/10.1007/978-3-319-14923-3\\_4](https://link.springer.com/chapter/10.1007/978-3-319-14923-3_4) Дата доступу 11.05.2023
4. *Game Theory — The Minimax Algorithm Explained* <https://towardsdatascience.com/how-a-chess-playing-computer-thinks-about-its-next-move-8f028bd0e7b1> Дата доступу 11.05.2023
5. *Monte Carlo Simulation: History, How it Works, and 4 Key Steps*  
<https://www.investopedia.com/terms/m/montecarlosimulation.asp> Дата доступу 11.05.2023