

Методы и алгоритмы, применяемые в ИИ для позиционных, логических игр

Студент: Кочуркин И.А., ИУ7-104

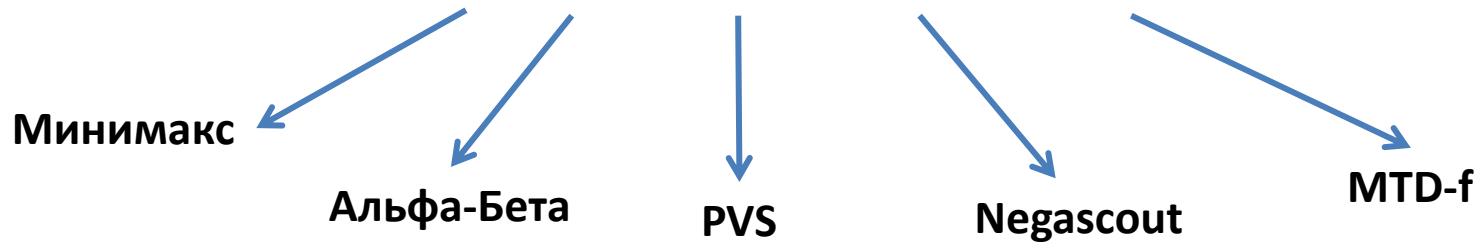
Руководитель: Филиппов М. В.

Сравнительный анализ

	Шахматы	Го	Точки
Цель	Поставить мат сопернику	Захват территории	Захват точек противника
Комбинаторная сложность (NP)	Да	Да	Да
Сложность правил	Средняя	Низкая	Низкая
Размер поля	8*8	9*9 — 19*19	39*32
Фактор ветвления	10^{47}	10^{360}	10^{1248}
Возможное количество партий	$\sim 10^{120}$	$\sim 10^{10^{48}} - 10^{10^{171}}$	
Среднее количество ходов в партии	80	150	
Сложность представления	Средняя	Высокая	Высокая
Аддитивная природа игры	Нет	Да	Да
Сложность формализуемости правил	Средняя	Тяжелая	Тяжелая
Уровень динамичности	Высокий	Средний	Отсутствует
Ко-борьба	-	Да	Нет
Уровень исследований ИИ	Высокий	Высокий	Низкий

Цель: Разработка и исследование эффективных методов и алгоритмов ИИ в позиционной игре «Точки».

Деревья поиска



Достоинства:

- Простота реализации.
- Предсказуемость результата.

Недостатки:

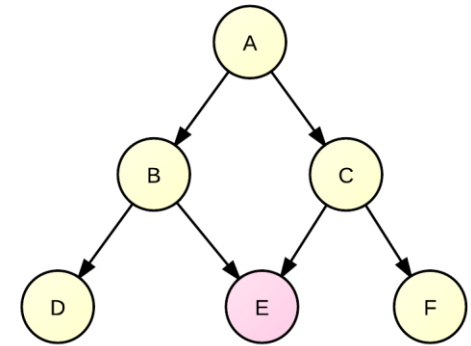
- Сложность создания эффективной оценочной функции.
- Сложность создания эффективного генератора ходов.
- Большой фактор ветвления.
- Сложность реализации эффективного параллельного алгоритма.

Кеширование и хеширование

- Zobrist-хеширование

$$Z_n = Z_n \text{ xor HashTable}[x][y][\text{color}]$$

- Запись информации (типа узла, глубина, лучший ход) при состоянии поля Z_n в таблицу перестановок для последующего использования в игровом дереве.



Достоинства:

- Минимизация коллизий в игровом дереве.
- Использование не только для хеширования состояний доски (поля), но и для более высокоуровневых примитивов.
- Можно использовать результаты при последующих расчетах.

Недостатки:

- Тяжело обнаруживаемые ошибки при коллизиях ключа и индекса.
- Сложность реализации параллельного алгоритма поиска (т.к. хеш-таблица общая, то нужно использовать взаимоблокировки или атомарные операции).

Методы Монте-Карло



Простой метод:

- Моделируется N случайных партий
- $Next = \max_{positions} \frac{Win}{Fail}$

Достоинства:

- Простота реализации.
- Простая и эффективная реализация параллельного алгоритма.
- Стратегический поиск.
- Положительный опыт использования метода UCT в играх против профессионалов (Го).

Недостатки:

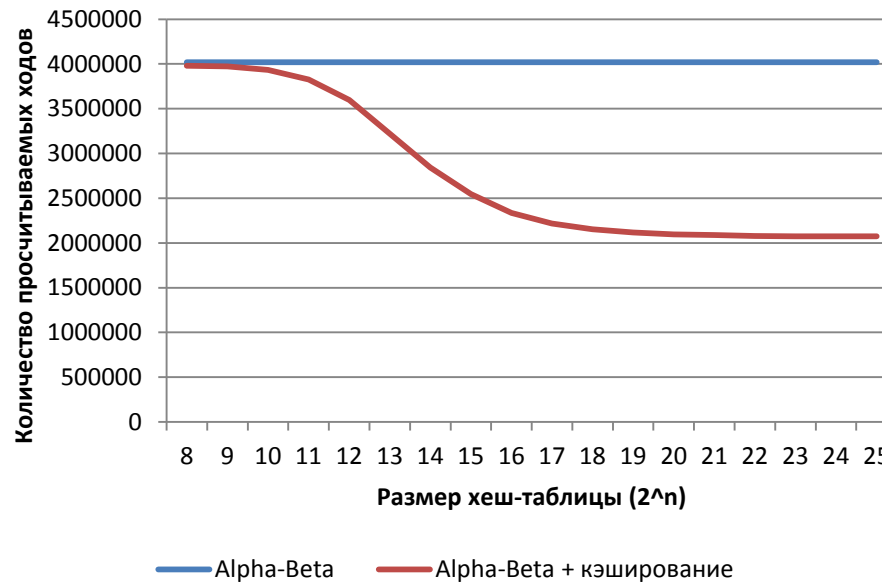
- Требуются большие вычислительные ресурсы для приемлемой игры.
- Большая энтропия => Высокая эффективность для полей маленького размера.

Метод UCT (2006):

- Моделируется N случайных партий
- $Uct = \begin{cases} \frac{Win}{child.Visits} + UctK \cdot \sqrt{\frac{\ln(node.Visits)}{child.Visits}}, & child.Visits > 0 \\ rand(K, M), & child.Visits = 0 \end{cases}$
- $Next = \max_{positions} Uct$ на каждом шаге

Текущие исследования и выводы

- Эффективность хеш-таблицы в зависимости от ее размера. Оптимальный размер: $\sim 2^{20}$ элементов.



- Метод UCT работает лучше, чем Alpha-Beta. Ходы выбирались на расстоянии менее двух позиций до всех поставленных точек. При этом основной проблемой UCT является случаи каскадных атак и защит.

Дальнейший план

- Методы минимакса:
 - Статичная природа игры в «Точках» -> просчет точек только одного цвета и нахождение предварительных alpha и beta для последующего использования их в алгоритмах минимакса (Alpha-Beta, Negascout).
 - Моделирование ходов «2 атаки 1 защита»: два хода точек одного цвета, приводящих к окружению + один ход точек другого цвета, защищающий группу. Это довольно часто-встречающаяся ситуация.
- Экспертные системы:
 - Разработать базу шаблонов для точек (формат, атрибуты).
 - Реализация распознавания паттерном с помощью КА.
- Методы Монте-Карло:
 - Модификация метода УСТ: просчет не всей партии, а локального участка, например соединение групп.
 - «Умный» выбор ходов (с помощью шаблонов).
 - Исследовать метод в различных вариациях с различными параметрами.
- Общее:
 - Реализовать и исследовать описанные алгоритмы в комбинации друг с другом.

Выводы:

1. Большой фактор ветвления у «Го» и «Точек» => глубокий Альфа-бета поиск неприменим.
2. Аддитивная природа игры у «Го» и «Точек» => количество игровых моментов увеличивается (в отличие от шахмат).
3. Относительная статичность камней (точек) => легко для человека, но тяжело для ЭВМ.
4. Игры «Шахматы» и «Го» хорошо исследованы. «Точки» – нет (из-за малой популярности).