UTTT – Анализ результатов

Для тестирования алгоритма можно использовать параметр глубины обхода по дереву решений.

Исходя из этого введем ботов:

* Легкий бот – 2 хода
* Средний бот – 4 хода
* Сложный бот – 5 ходов
* Экспериментальный бот – 7 ходов

Программа работает за константное количество памяти, так как память в программе выделяется всегда в максимально требуемом количестве

Итого: 1,2 МБ = 1228,8 КБ

Проверим время работы алгоритма на пустом поле (так как на нем больше всего возможных вариантов ходов).

|  |  |
| --- | --- |
| **Глубина обхода по дереву** | **Время работы программы** |
| 7 | 3814 мс |
| 6 | 1221 мс |
| 5 | 396 мс |
| 4 | 111 мс |
| 3 | 15 мс |
| 2 | 7 мс |
| 1 | 5 мс |

Было проведено исследование в виде игры ботов друг с другом.

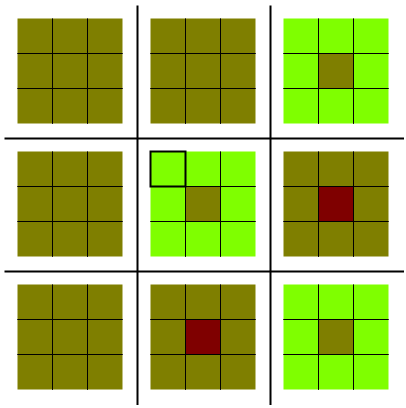
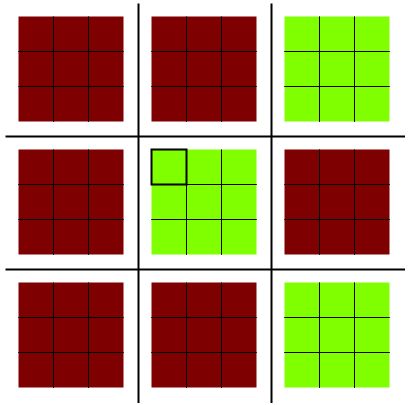
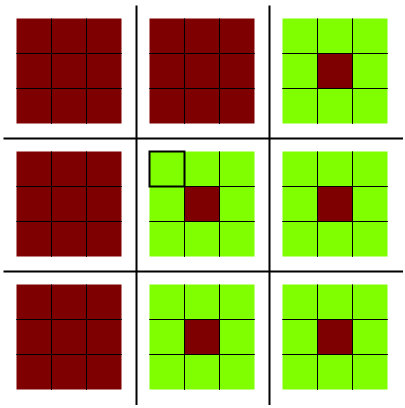
|  |  |  |  |
| --- | --- | --- | --- |
|  | **Легкий** | **Средний** | **Сложный** |
| Легкий | Ничья | Средний | Сложный |
| Средний | Средний | Ничья | Сложный |
| Сложный | Сложный | Сложный | Ничья |

В данной таблице отображены результаты при раскладе, что более сильный бот ходит первый.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Легкий** | **Средний** | **Сложный** |
| Легкий | Ничья | Ничья | Сложный |
| Средний | Ничья | Ничья | Ничья |
| Сложный | Сложный | Ничья | Ничья |

В данной таблице отображены результаты при раскладе, что более слабый бот ходит первый.

Можно увидеть, что тот, кто начинает первый имеет небольшое преимущество в виде бонуса: возможности закончить игру первым. Например, средний бот, играя со сложным, может выйти в ничью, поскольку у него есть возможность закончить игру первым. Однако можно заметить, что если бы у среднего бота не было такой возможности, то он бы проиграл на следующий ход.

Также можно наглядно увидеть различия “рассуждений” ботов

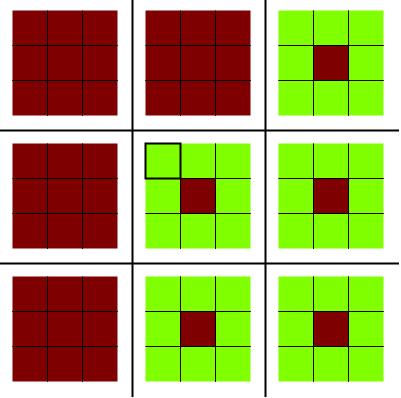
Результаты для легкого, среднего и сложного ботов соответственно.

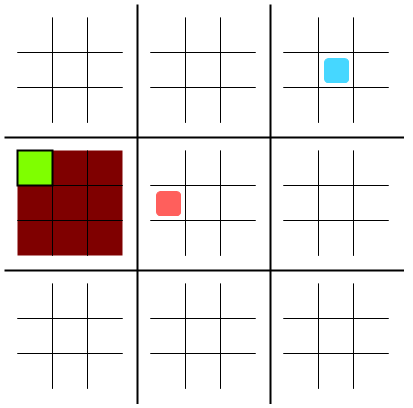
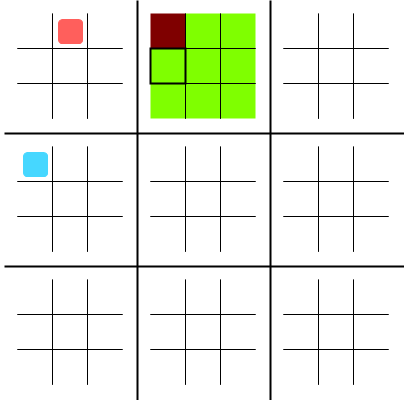
Также был изучен экспериментальный бот. Его результаты практически не отличаются от сложного бота. Так что можно сделать вывод, что глубина обхода будет давать выигрыш в силе бота лишь до определенного момента.

Был исследован фактор ошибки во время игры: т.е. насколько сильно может повлиять ошибка на дальнейшую игру.

Проанализируем это на примере сложного и среднего бота. Как было выявлено: сложный бот выигрывает в среднего, если ходит первым. Проверим то, насколько это будет аналогично при плохом ходе сложного бота.

Начальное состояние – первый рисунок, ответ бота – при ходе в центральную клетку, третий рисунок – ответ бота при ходе в не центральную клетку.



Если походить в любую центральную клетку в маленьком поле, то средний бот всегда походит в левую клетку центрального поля. А дальше игра будет все время идти абсолютно одинаково в симметричном виде.

Если же походить в не центральную клетку – то все будет аналогично, средний бот будет ходить одинаково, только уже в верхнюю клетку поля, в которое заставил походить его сложный бот.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ошибочный ход сложного бота | Ответный ход среднего бота | Количество ходов игры | Разность между оценочной функции между оптимальным и выбранным ходом | Победитель |
| Центральная клетка в маленьком поле | Левая клетка в центральном большом поле | 52 | 4 | Сложный бот |
| Любая клетка кроме центральной в красных полях | Верхняя клетка в том поле, куда вынудил ходить сложный бот | 52 | 4 | Сложный бот |

Как можно заметить, что ошибка на первом ходу игры не так критична по двум причинам:

* Поскольку это начало игры, то существует ещё огромное множество развитий игры, что дает сложному боту шанс реабилитироваться
* Также можно заметить, что “плохой” ход не настолько уж и плохой. Ведь модуль разности оценочной функции между плохим и хорошим ходом равен 4, что по меркам алгоритма очень мало.

После данных рассуждений приходит мысль проверить последствия ошибки на более поздних стадиях игры.

|  |  |  |  |
| --- | --- | --- | --- |
| Стадия игры | Разность между оценочной функции между оптимальным и выбранным ходом | Количество ходов игры | Победитель |
| Средина игры | 10 | 55 | Сложный бот |
| Средина игры | 15 | 56 | Ничья |
| Средина игры | 22 | 58 | Ничья |
| Средина игры | 5 | 53 | Сложный бот |
| Конец игры (3-4 шага до конца партии) | 5 | 56 | Ничья |
| Конец игры | 20 | 52 | Ничья |
| Конец игры | 33 | 51 | Средний бот |
| Конец игры | 59 | 47 | Средний бот |

Можно прийти к увлекательному умозаключению: что каждая сделанная ошибка имеет более плохие последствия, если она сделана ближе к концу игры. Можно увидеть, что незначительные ошибки, сделанные в средине игры могут и не повлиять на исход игры. Более серьезные ошибки могут привести к ничье. Но более интересный момент заключается в исследовании конца игры. Можно увидеть, что даже если всю игру сыграть без ошибок, но в конце допустить даже малейшую ошибку, то можно оказаться в ничье, а при серьезных ошибках – проиграть.

Алгоритм был протестирован на портале hakerrank.com. Среди 159 участников занял 8-ое место.

