

УДК 004.832.24

С. И. Зыкин, магистрант, Ижевский государственный технический университет имени М. Т. Калашникова
А. В. Коробейников, кандидат технических наук, Ижевский государственный технический университет имени М. Т. Калашникова

П. П. Лугачев, Ижевский государственный технический университет имени М. Т. Калашникова

Д. А. Юсупов, магистрант, Ижевский государственный технический университет имени М. Т. Калашникова

ИСПОЛЬЗОВАНИЕ АЛГОРИТМА МИНИМАКС С АЛЬФА-БЕТА УСЕЧЕНИЕМ ДЛЯ ЭТАПА РОЗЫГРЫША ИГРЫ «СПОРТИВНЫЙ БРИДЖ»

Рассматривается решение задачи розыгрыша игры «спортивный бридж». Рассматривается вариант решения задачи с полной информацией (розыгрыш с двумя «болванами»). Для решения задачи используется классический алгоритм минимакс. Для сокращения размеров дерева состояний используется альфа-бета усечение. Разработано программное обеспечение для выполнения розыгрыша с двумя «болванами».

Ключевые слова: спортивный бридж, розыгрыш, минимакс, альфа-бета-усечение, дерево состояний.

Подходы систем искусственного интеллекта (СИИ) находят широкое применение. Одним из направлений исследования СИИ является решение игровых задач [1]. Подходы СИИ применяются во многих интеллектуальных играх: шахматы, шашки, бридж, го. Сейчас наиболее успешно компьютер играет в шахматы и шашки, наименее успешно – в го. Бридж занимает промежуточное положение.

Данная работа отражает результаты применения алгоритма минимакс с альфа-бета усечением к задаче розыгрыша игры «спортивный бридж».

Игра «спортивный бридж»

Спортивный бридж является карточной интеллектуальной игрой [2]. Играют две пары игроков, игроки одной пары сидят напротив друг друга. Игроки называются по сторонам света: Север (*N*), Восток (*E*), Юг (*S*) и Запад (*W*). Пара *NS* играет против пары *EW*. Используется колода из 52 карт – от двойки до туза четырех мастей. В игровой сдаче каждый игрок получает по 13 карт. Каждая сдача состоит из этапов – торговли и розыгрыша.

При розыгрыше первый ход делает левый оппонент разыгрывающего. После этого «болван» (партнер разыгрывающего) кладет свои карты на стол и его картами управляет разыгрывающий. Игра ведется так же, как и в других играх на взятки, бить козырем необязательно. На этапе розыгрыша задачей каждой пары является собрать наибольшее количество взятков.

В спортивном (дубликатном) бридже одна и та же сдача играется многократно разными игроками [2]. Бридж включает элементы передачи информации партнеру по команде, введения в заблуждение противников и расчет результатов своих действий для различных вариантов. В игре присутствует кооперация, конкуренция и неполная информация о раскладе.

Формализация задачи

Процесс розыгрыша представляют в виде графа, узлы которого соответствуют карте, которой сделал ход очередной игрок. Состояние игры при текущем

ходе отражает оставшиеся карты каждого игрока и количество взятков, взятых каждой парой.

Количество вариантов раскладов в бридже по основной формуле комбинаторики составляет $1,6 \cdot 10^{31}$ [3]. Количество вершин в дереве игровых состояний при розыгрыше $1,5 \cdot 10^{39}$ [3], но с учетом требования ходить в масть карты захода примерно получим $6,0 \cdot 10^{18}$. В данной работе рассматривается вариант розыгрыша при наличии полной информации о раскладе (с двумя «болванами») [3].

Для решения игровых задач в СИИ традиционно для нахождения оптимального хода игрока применяют алгоритм минимакс. Данный алгоритм может применяться как на полную глубину просмотра развития игры, так и на ограниченную глубину с использованием эвристик [1, 4].

Прямой алгоритм минимакс требует двух проходов по области поиска. На первом проходе происходит построение дерева игровых состояний вглубь области поиска, где происходит оценка терминальных состояний. На втором проходе происходит минимаксный перенос оценок терминальных состояний вверх по дереву состояний к корню. Минимакс рассматривает все ветви в пространстве поиска, включая те из них, которые могли быть исключены более интеллектуальным алгоритмом [1, 4].

Для розыгрыша в бридже в качестве оценки состояния будем использовать количество выигранных взятков команды. Такая оценка окончательно формируется только на терминальных узлах (листьях) дерева состояний. Однако количество взятков на ограниченном числе игровых кругов может являться неплохой эвристикой. При этом для игрока оцениваемой пары, оценка числа взятков должна стремиться к максимуму, а для пары противников – к минимуму.

При формировании дерева состояния розыгрыша бриджа проявляется следующая особенность: очередность сторон игры (*MIN* и *MAX*) может нарушаться каждые 4 хода или 1 круг (взятка) розыгрыша. Первый ход очередного круга делает игрок, взявший предыдущую взятку, а дальнейшие ходы круга игро-

ки делают по часовой стрелке. Пример очередности ходов:

- 13 карт, круг 1:
[E (MIN) - S (MAX) - W (MIN) - N (MAX)] -
- 12 карт, круг 2:
[S (MAX) - W (MIN) - N (MAX) - E (MIN)] -
- 11 карт, круг 3:
[N (MAX) - E (MIN) - S (MAX) - W (MIN)] -

Результат применения алгоритма минимакса к графу состояний розыгрыша бриджа показан на рис. 1 (глубина дерева равна 4 ходам или 1 кругу розыгрыша). Эвристическая оценка корневого состояния равна $e_0 = 0$ (у команды нет возможности выиграть взятку в этом круге при ходе картой 5♣).

При поиске по дереву состояний только для простых задач возможен полный просмотр дерева на основе прямого алгоритма минимакс. В СИИ разработан ряд методов сокращения просматриваемых

вершин. В работе [5] для сокращения дерева состояний предлагается использование продукций (правил) аналогично тому, как делают бриджисты. В данной работе применяется классический метод сокращения числа просматриваемых состояний при использовании алгоритма минимакс – альфа-бета усечение [1, 4].

Идея альфа-бета усечения заключается в том, что оценивание ветви дерева состояний может быть досрочно прекращено (без просмотра состояний ветви), если было найдено, что для этой ветви значение оценивающей функции в любом случае хуже, чем вычисленное для предыдущей ветви [1, 4].

Пример применения алгоритма альфа-бета усечения, приведенного ранее игрового дерева, показан на рис. 2, где альфа-бета усечение позволило на 30 % сократить количество просматриваемых состояний.

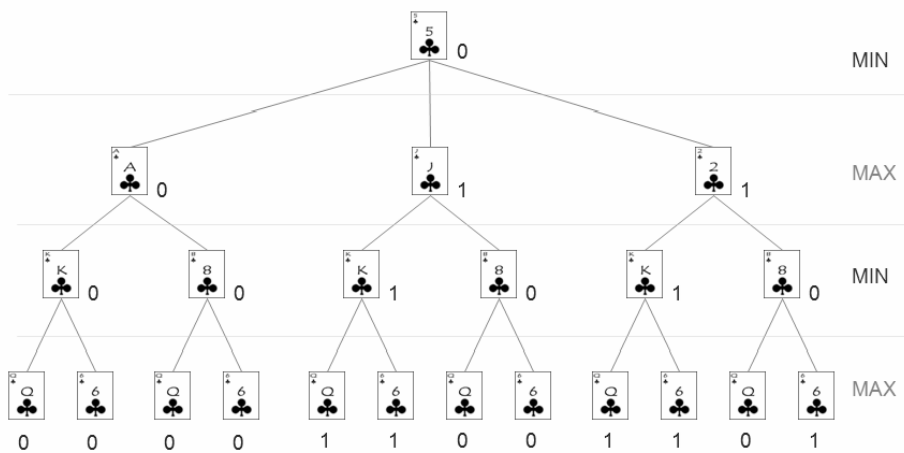


Рис. 1. Пример алгоритма минимакс для розыгрыша спортивного бриджа

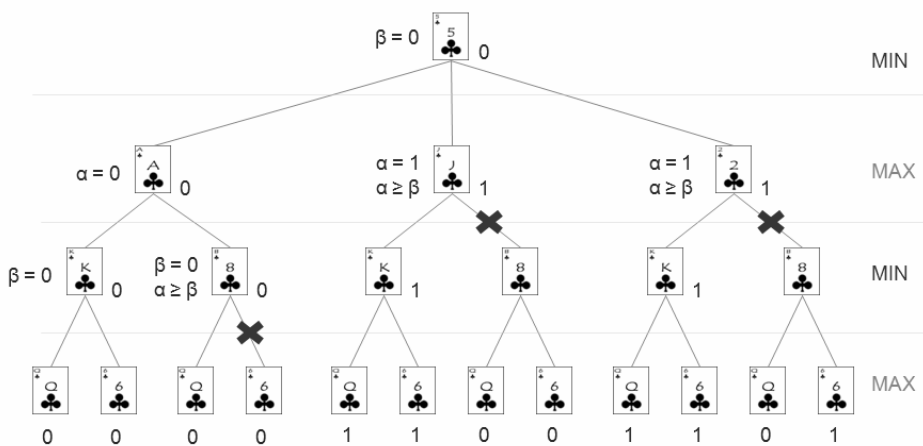


Рис. 2. Пример альфа-бета усечения для розыгрыша спортивного бриджа

На результат альфа-бета усечения оказывает влияние порядок просмотра дочерних вершин узла. При правильном упорядочивании дочерних вершин при просмотре можно повысить эффективность просмотра и сократить число узлов [4].

Программная реализация

Рассмотрим программную реализацию алгоритма минимакс с альфа-бета усечением для розыгрыша на языке Java:

```

/* Поиск наилучшего хода для оцениваемой
пары: player текущий оцениваемый игрок;
depth глубина дерева поиска; score
количество взяток выигранных оцениваемой
парой на данной глубине */
int Search(int player, int depth, int
score){
// если на игровом поле лежат 4 карты
if (isFullTable()){
// определяем победителя
int winner = getResult(player);
// определяем выигрыш оцениваемой
пары: выигрыш оценка 1, иначе 0
score += Evaluate(player, winner);
ClearGameField(); //очищаем игровое
поле
/* если выиграла пара, чей ход был
первый, то переходим на следующий
уровень дерева поиска с неизмененными
картами */
if (isWonByTheFirstPair(player,
winner)){
AlphaBeta ab = new AlphaBeta(-beta,
-alpha);
ab.initializeCards(north, earth,
south, west, table);
player = winner;
return - ab.Search(player, depth-1,
score);
}
player = winner; // текущим игроком
становится победитель
}
// если достигли указанной глубины поиска,
то возвращаем результат
if (depth <= 0)
return score;
// получить возможные ходы игрока
List<Card> moves =
getPlayerCards(player);
int current = 1;
// оценивание возможных ходов игрока
while (!moves.isEmpty() &&
(moves.size() != current) && (alpha <
beta)) {

```

```

AlphaBeta ab = new AlphaBeta(-beta, -
alpha);
// совершаем ход: перебор карт по убываю
ванию номинала
Card move = MakeMove(moves, current,
ab);
// переход на след. уровень дерева:
оценка игрока левее от текущего
int temp = - ab.Search((player + 1) %
4, depth - 1, score);
// возвращаем сделанный ход
UnMakeMove(moves, move);
if (temp > alpha)
alpha = temp;
current++; // переходим к следующему
ходу
}
return alpha;
}

```

Для тестирования и изучения работы приведенного алгоритма была разработана программа *BridgeGame*, позволяющая симулировать процесс бескозырного розыгрыша сдач. Графический интерфейс программы представлен на рис. 3 и 4. Данная программа была написана на языке *Java*. в среде разработки *NetBeans IDE 7.2*.

Структура программы включает в себя следующие модули.

1. Работа с файлами: выбор файла инициализации; загрузка параметров из файла; вывод статистики игры в файл в формате *xml*.
2. Минимаксный алгоритм: определение оптимального хода согласно минимаксному алгоритму с альфа-бета усечением.
3. Симуляция игры: проведение игры выбранным раскладом карт согласно правилам игры в бридж. Ходы игроков делается по минимаксному алгоритму.
4. Интерфейс программы: наглядное представление каждого хода игрока; вывод оценки состояний розыгрыша; счет взяток розыгрыша; управление ходом розыгрыша через панель инструментов.

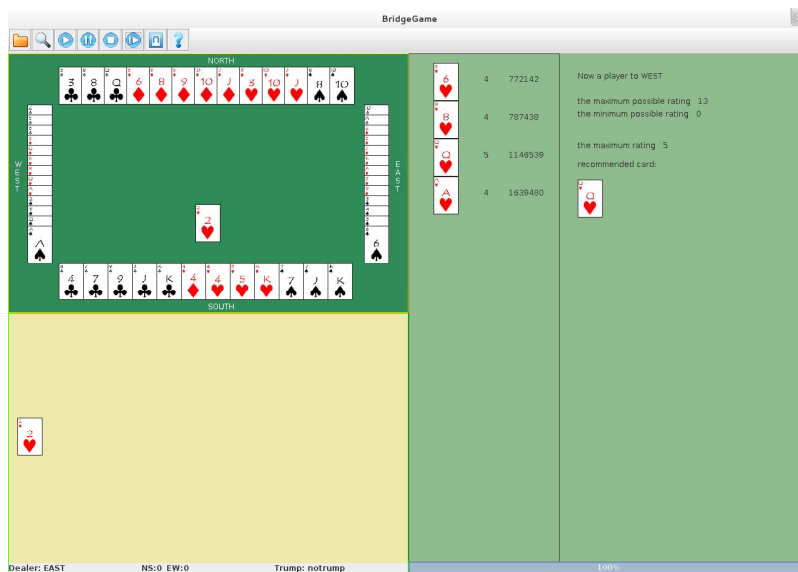


Рис. 3. Интерфейс программы *BridgeGame*

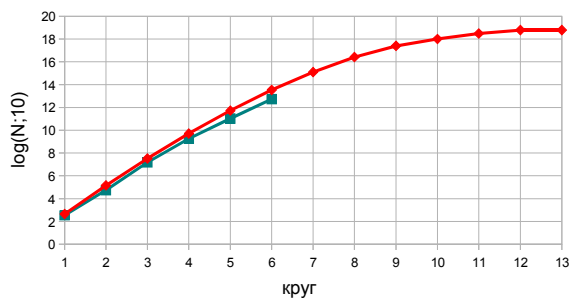


Рис. 4. Количество игровых состояний без альфа-бета усечения

Результаты экспериментов

Для экспериментов был использован набор из 1000 заранее сгенерированных игровых сдач.

На рис. 4 представлено сравнение количества игровых состояний N дерева при использовании алго-

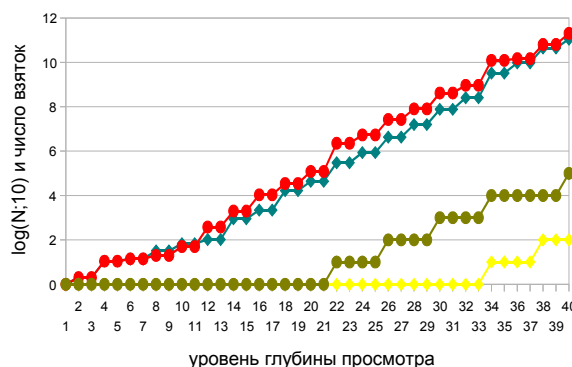
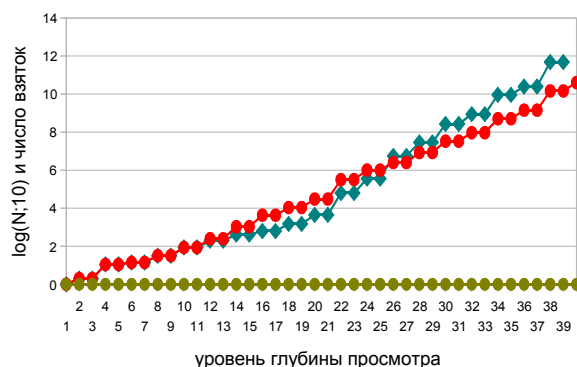


Рис. 5. Количество игровых состояний с альфа-бета усечением и число взяток

Количество взятых взяток при упорядочивании карт масти «старшая-младшая» для нижнего примера возрастает (по убыванию – 2, а «старшая-младшая» – 5 взяток), а для верхнего примера не меняется (0 взяток). Количество просмотров узлов дерева при одинаковом количестве взятых взяток в случае упорядочивания карт масти «старшая-младшая» сокращается (верхний пример).

Максимальная глубина просмотра составила 40 уровня (10 кругов). Для симуляции процесса розыгрыша в реальном времени используется меньшее число уровней просмотра.

Выводы

1. Использование минимакса с альфа-бета усечением для розыгрыша сдач в спортивном бридже с двумя «болванами» позволяет сократить количество просмотров узлов дерева состояний.

2. Разработано программное обеспечение согласно предложенным алгоритмам, позволяющее симулировать процесс розыгрыша.

3. Дальнейшее сокращение количества просмотров игровых состояний целесообразно проводить

ритма минимакс без альфа-бета усечения, рассчитанное для одной из сдач (на 6 кругов), и теоретическое значение (на 13 кругов) в логарифмическом масштабе.

На рис. 5 представлены результаты работы алгоритма минимакс для двух игровых сдач при построении дерева для выбора карты первого хода сдачи. График с метками в виде ромбов соответствует просмотру дочерних вершин по убыванию номинала карты масти, а с круглыми метками – просмотру карт с упорядочиванием номинала «старшая-младшая»: во-первых, проверяется самая старшая карта масти, во-вторых, самая младшая, и так далее с чередованием старших и младших карт, оставшихся в масти. Дополнительно на графиках приведено число выигрываемых взяток. При построении графиков количества просматриваемых состояний глубина просмотра изменялась на два уровня.

с совместным использованием альфа-бета усечения и дополнительных эвристик для оценки игровых состояний, например, так, как это предложено в работе [5], или в соответствии с эвристиками, описанными в [3].

Библиографические ссылки

1. Люгер Д. Ф. Искусственный интеллект : Стратегии и методы решения сложных проблем. – М. : Вильямс, 2003. – 864 с.
2. Риз Т. Бридж для начинающих : Искусство побеждать. – М. : Центрполиграф, 2010 г. – 160 с.
3. Обзор состояния программ спортивного бриджа / А. В. Коробейников, С. И. Зыкин, Р. Х. Судуров, И. С. Ефремова // Информационные системы в промышленности и образовании : сб. трудов молодых ученых. – Ижевск : Изд-во ИжГТУ, 2012. – С. 69–77.
4. Корнилов Е. Н. Программирование шахмат и других логических игр. – СПб. : БХВ-Петербург, 2005. – 272 с.
5. Использование продукций и алгоритма минимакс для этапа розыгрыша в игре «спортивный бридж» / Д. А. Юсупов, А. В. Коробейников, С. И. Зыкин, С. Ф. Егоров // Вестник КИГИТ. – 2012. – № 7. – С. 40–45.

S. I. Zykin, Master's Degree student, Kalashnikov Izhevsk State Technical University

A. V. Korobeynikov, PhD in Engineering, Associate Professor, Kalashnikov Izhevsk State Technical University

P. P. Lugachev, Kalashnikov Izhevsk State Technical University

D. A. Yusupov, Master's Degree student, Kalashnikov Izhevsk State Technical University

Application of Minimax with Alpha and Beta Truncation Algorithm for Playing Stage of "Duplicate Bridge" Game

In article the solution of a problem playing stage "Duplicate bridge" game is considered. The version of the solution of a task with full information (playing with two "dummy") is considered. For the task solution the classical minimax algorithm is used. For reduction of the game states the tree size alpha-beta truncation is used. The software is developed for playing performance with two "dummy".

Key words: duplicate bridge, playing stage, minimax algorithm, alpha-beta truncation, game states tree.