Санкт-Петербургский государственный университет информационных технологий, механики и оптики

Кафедра "Компьютерные технологии"

Е. Д. Долгих, Б.М. Ярцев

Разработка системы, моделирующей поведение агента в стохастической игре с неполной информацией, на примере игры «Покер Техасский Холдем»

Санкт-Петербург
2010

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Игры давно зарекомендовали себя в качестве удобной модели для проведения исследований методов искусственного интеллекта. Во многом это связано с тем, что существуют четко определенные правила и цели в отличие от ситуаций реальной жизни. Благодаря этим свойствам исследователям проще воплощать свои идеи и измерять полученные результаты.

Игры можно классифицировать по многим параметрам, один из важнейших – доступность информации игрокам. Игра, в которой каждый игрок обладает всей информацией о текущем состоянии, называется игрой с полной информацией, в противном случае – игрой с неполной информацией. Шашки, шахматы, го являются играми с полной информацией. Примерами игр с неполной информацией могут служить покер, морской бой, бридж, дилемма заключенного. Другим важным параметром игр является наличие в них элемента случайности. Игра, содержащая случайные события такие, как сдача карт в покере, называется случайной или стохастической, в противном случае игра называется детерминированной.

В прошлом большинство исследований было посвящено детерминированным играм с полной информацией. Данный класс игр является более простым случаем для изучения, и исследователям удалось построить алгоритмы, достигающие достаточно высокого уровня игры, сравнимого с игрой профессионалов(Deep blue[1] в шахматы, Chinook[2] в шашки). В реальной жизни гораздо чаще встречаются ситуации схожие со случайными играми с неполной информацией. В последнее десятилетие данной области было уделено гораздо большее внимание. Большинство исследований были направлены на поиски оптимальных стратегий, в частности равновесия Нэша[3], которое обеспечивает оптимальную игру против идеального, не совершающего ошибок, оппонента. Учитывая размеры и сложность задач, в реальной жизни существование таких «игроков» если и возможно, то маловероятно. Принимая во внимание, что результат игры напрямую зависит от

реакции и действий соперника, крайне полезной для изучения представляется область моделирования поведения игроков с целью обнаружения потенциальных уязвимостей в используемых ими стратегиях.

Цели исследования:

- разработать вероятностную модель для предсказания поведения игрока в стохастической игре с неполной информацией, применимую к игре «Покер Техасский Холдем»;
- разработать алгоритм обучения модели;
- адаптировать алгоритм построения стратегии наказания к разработанной модели;
- реализовать алгоритмы в виде программного обеспечения ЭВМ.

ГЛАВА 1. ОСНОВНЫЕ ПОНЯТИЯ И ОБЗОР СУЩЕСТВУЮЩИХ ИССЛЕДОВАНИЙ

В данной главе даются основные понятия и рассматриваются необходимые в дальнейшем методы теории игр. В их число входят формы представления игры, типы стратегий и их свойства. Так же описывается игра «Покер Техасский Холдем» и разработанные к настоящему времени алгоритмы построения стратегий к данной игре.

1.1. Основные понятия

В данном разделе даются используемые в работе термины теории игр.

1.1.1. Экстенсивная форма

Игра в *экстенсивной* или *расширенной форме*, представляется в виде дерева. Каждая вершина в дереве соответствует состоянию игры. Вершины могут быть одного из двух типов:

- *вероятностные вершины* – вершины, в которых некий случайный процесс, такой как сдача карт, переводит игру в другое состояние;
- *вершины принятия решения*, представляющие состояния игры, в которых один из игроков должен совершить действие.

На рис. 1 приведен пример игры, представленной в экстенсивной форме. Ромбом обозначен случайный процесс, равновероятно переводящий игру в одно из двух состояний. Серым цветом обозначены вершины принятия решения первого игрока, белым – второго. В результате выигрыш первого игрока равен числу, записанному в листе дерева, в котором завершилась игра. Значение выигрыша второго игрока противоположно по знаку значению выигрыша первого игрока.

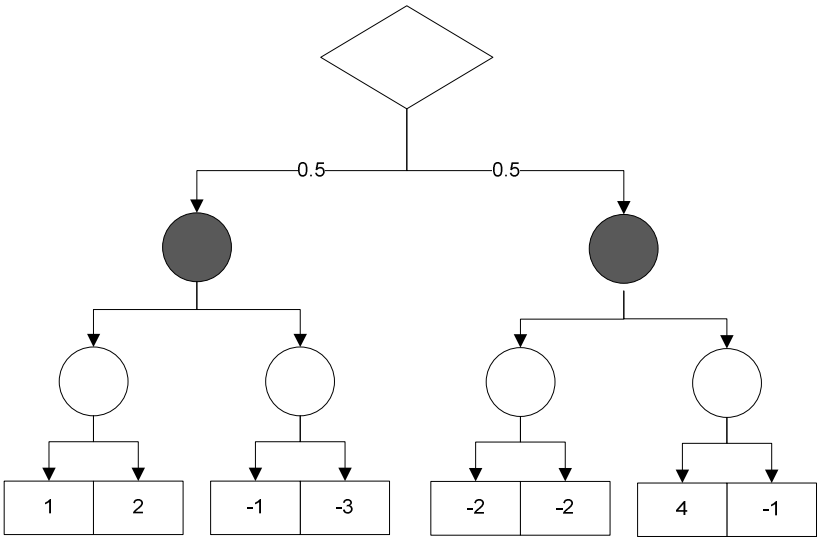

Рис. 1. Игра в экстенсивной форме

В играх с неполной информацией игроки не знают точно вершины, в которой они совершают ход, а могут лишь знать, что эта вершина принадлежит некоторому подмножеству множества вершин графа. Такие подмножества будем называть *информационными множествами*.

На рис. 2 изображено дерево игры аналогичной игре, представленной на рис. 1, за исключением того, что игроки не знают в какое именно состояние перешла игра посредством случайного процесса. Таким образом, в каждой из обведенных пар вершин состояния для игроков неразличимы. Обведенные сплошной линией пары вершин образуют информационные множества в данной игре.

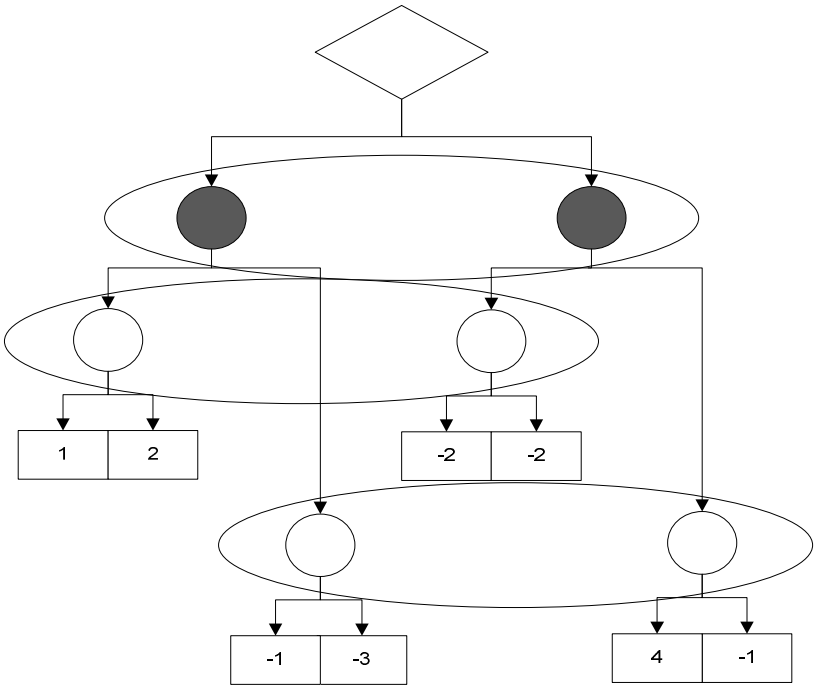

Рис. 2. Игра с неполной информацией в экстенсивной форме

Определим игру в экстенсивной форме формально[4].

Определение 1.

Игра в экстенсивной форме состоит из следующих компонент:

- *Конечное множество игроков* N.
- *Множество последовательностей* H*, такое что:*
  - $\emptyset \in H$;
  - *Если* $(a^k)_{k=1}^{K} \in H$ *и* $L < K$*, то* $(a^k)_{k=1}^{L} \in H$;
  - *Рассмотрим последовательность* $(b^k)_{k=1}^{\infty}$*, если для* $\forall L \in \mathbb{N}$ $(b^k)_{k=1}^{L} \in H$*, то* $(b^k)_{k=1}^{\infty} \in H$.

  Каждый элемент множества H *представляет историю действий, начиная с начала игры, компонента* a^i *элемента представляет* i*-ое действие, совершенное с начала игры. Последовательность* $(\mathrm{a}^k)_1^K \in H$ *называется терминальной, если она содержит бесконечное число элементов или* $\nexists\, \mathrm{a}^{K+1}: (\mathrm{a}^k)_1^{K+1} \in H$. *Множество терминальных последовательностей обозначим за* Z.
- $\forall h \in H \backslash Z$ *множество действий* $A(h) = \{a: (h, a) \in H\}$.

- *Функция* $P: H \backslash Z \to N \cup \{c\}$ *(*$P(h)$ *определяет игрока, действующего после истории* h*, если* $P(h) = c$ *действие определяется природой игры, которую формально представляет игрок* c*).*
- *Функция* f_c*, сопоставляющая каждой последовательности* $h \in H: P(h) = c$ *вероятностную меру* $f_c(\cdot\,|h)$ *на* $A(h)$*. (*$f_c(a|h)$ *представляет вероятность того, что после истории* h *случится событие* a*).*
- *Для каждого игрока* $i \in N$ *разбиение* $\mathcal{I}_i$ *множества* $\{h \in H: P(h) = i\}$*, такое, что для* $\forall I_i \in \mathcal{I}_i,\ \forall h, h' \in I_i \Rightarrow A(h) = A(h')$*. Множество* $I_i \in \mathcal{I}_i$ *представляет информационное множество* i*-го игрока.* $A(I_i) = \{a: (h, a) \in H, h \in I_i\}$
- *Функция выигрыша* $U: N \times Z \to \mathbb{R}$*.*

Элементы множество H в дальнейшем будем называть вершинами.

Часто рассматриваются игры, в которых вершины, принадлежащие одному информационному множеству, различаются только скрытой информацией, которая недоступна игроку. Такие игры называются играми с *полной памятью*. Второй широкий класс игр с неполной информацией – игры с *неполной памятью,* то есть игры, в которых участник может «забыть» часть действий, совершенных им ранее.

1.1.2. Стратегии поведения

Рассмотрим игру в экстенсивной форме $< N, H, P, f_c, \mathcal{I}_{i \in N}, U >$. Под стратегией игрока i будем понимать статический набор правил, определяющий для $\forall I_i \in \mathcal{I}_i$, какое действие должно быть совершено. *Чистой стратегией* будем называть такой набор правил, который для любого информационного множества будет определять ровно одно действие. *Стратегией поведения* будем называть стратегию, которая для каждого информационного множества определяет распределение вероятностей, согласно которому во время игры

определяется, какое именно действие должно быть совершено. Ниже приведено формальное определение стратегии поведения[5].

Определение 2.

Под стратегией поведения σ_i игрока i в игре Γ *в экстенсивной форме* $< N, H, P, f_c, \mathcal{I}_{i \in N}, U >$ *будем понимать функцию* F, *сопоставляющую каждому информационному множеству i-го игрока $I_i \in \mathcal{I}_i$ вероятностную меру на $A(I_i)$. За Σ_i обозначим множество возможных стратегий игрока i.*

Определение 3.

Набор стратегий поведения $\sigma = \{\sigma_i\}_{i=1}^{N}$ для всех игроков игры в экстенсивной форме с N игроками будем называть стратегическим профилем игры. Обозначим за σ_{-i} все стратегии в наборе кроме стратегии σ_i.

Обозначим за $\pi^{\sigma}(h)$ вероятность достижения вершины $h = (a^i)_{i=1}^{K} \in H$, если игроки выбирают свои действия согласно σ. Под $\pi_i^{\sigma}(h)$ будем понимать вероятность того, что игрок i, действуя в соответствии с $\sigma_i \in \sigma$, для любой вершины h', лежащей на пути от корня дерева до вершины h $h' = (a^i)_{i=1}^{j}$: $[\left(\exists (a^i)_{i=j+1}^{K} : \left(h', (a^i)_{i=j+1}^{K}\right) = h\right), P(h') = i]$ выберет действие $a^{h'}$ такое, что $\exists (a^i)_{i=j+2}^{K} : \left((h',\ a^{h'}), (a^i)_{i=j+2}^{K}\right) = h.$ Тогда $\pi^{\sigma}(h) = \prod_{i \in N \cup \{c\}} \pi_i^{\sigma}(h)$. Тогда ожидаемый выигрыш $u_i(\sigma)$ игрока i, в случае если игроки используют стратегический профиль σ, $u_i(\sigma) = \sum_{h \in Z} U(i, h) \pi^{\sigma}(h)$.

1.1.3. Равновесие по Нэшу

Определение 4.

Равновесием по Нэшу[3] будем называть стратегический профиль σ, при использовании которого, в случае если ровно один игрок изменит свою стратегию, значение его ожидаемого выигрыша, по крайней мере, не увеличится:

$$\forall i \; u_i(\sigma) = \max_{\sigma_i' \in \Sigma_i} u_i(\sigma_i', \sigma_{-i})$$

Таким образом, для игр двух лиц никакой из игроков не может получить ожидаемый выигрыш более чем $u_i(\sigma)$, в случае если его соперник используют стратегию принадлежащую равновесию по Нэшу. Для повторяющихся игр двух лиц, в которых после каждой партии происходим смена позиций, использование игроками такого стратегического профиля означает, что ожидаемое значение выигрыша каждого из игроков равно 0. В любой многошаговой игре двух лиц с идеальной памятью и конечным числом состояний в экстенсивной форме представления игры всегда существует равновесие по Нэшу в стратегиях поведения[3].

Определение 5.

Будем называть ϵ-равновесием по Нэшу стратегический профиль σ такой что:

$$\forall i \; u_i(\sigma) + \epsilon \geq \max_{\sigma_i' \in \Sigma_i} u_i(\sigma_i', \sigma_{-i})$$

1.1.4. Стратегии наказания

Определение 6.

В игре двух лиц с нулевой суммой стратегией наказания игрока i, использующего стратегию σ_i, называется следующая стратегия игрока j:

$$\sigma_j^{\text{H}} = \max_{\sigma_j' \in \Sigma_j} u_j(\sigma_j', \sigma_i)$$

Данные стратегии обладают существенным недостатком – если соперник изменит свою стратегию, ожидаемое значение выигрыша может значительно уменьшиться. В дальнейшем под стратегией наказания будем понимать стратегию, использующую особенности стратегии соперника для увеличения собственного выигрыша, при этом, возможно, не гарантируя максимально достижимого значения.

1.2. «Покер Техасский Холдем»

С момента появления теории игр покер неоднократно был использован различными исследователями в качестве примера для своих работ, включая работы фон Неймана и Моргенштерна[6], а также Джона Нэша[7]. В данном

разделе дается определение игры «Покер Техасский Холдем», одной из наиболее стратегически сложных разновидностей покера, и описываются ее свойства.

1.2.1. Покер

Покер – вид карточной игры, где игроки соревнуются в силе своих *карточных комбинаций (рук)*. В игре могут принимать участие от двух до десяти игроков. Игра делится на несколько стадий. Игроки, дошедшие до финальной стадии, открывают свои комбинации. Победителем раздачи становится либо игрок, который на последней стадии игры (шоудауне) показал наиболее сильную комбинацию, либо игрок, который остался в игре один. Игрок может остаться в раздаче один, если в ответ на его действие все игроки сбрасывают свои карты, то есть выходят из игры. В этом разделе будут подробно рассмотрены все стадии игры «Покер Техасский Холдем» с фиксированными размерами ставок.

1.2.2. «Техасский Холдем»

Игроки располагаются по кругу в фиксированном на все время игры порядке. Одна из позиций становится позицией *дилера*. Перед началом каждой партии позиция дилера сдвигается на одну позицию против часовой стрелки. Игрок, располагающийся на позиции, следующей за дилером, делает *малую обязательную ставку(small blind)*, следующий по порядку игрок делает *большую обязательную ставку(big blind)*. Размеры данных ставок фиксированы, обычно большая обязательная ставка в два раза больше малой. Данные ставки добавляются в *банк(pot)* текущей партии. Игра состоит из пяти стадий:

- *Префлоп*. Каждому из игроков сдаются две карты. Ни какой из игроков не получает информации о картах оппонентов. Происходит первый раунд торговли.
- *Флоп*. На стол выкладываются три общие карты. Происходит второй раунд торговли.

- *Терн*. На стол выкладывается четвертая общая карта. Происходит третий раунд торговли.
- *Ривер*. На стол выкладывается пятая общая карта. Происходит четвертый раунд торговли.
- *Вскрытие*. Игроки, не сбросившие карты на протяжении текущего розыгрыша, открывают свои закрытые карты. Игрок с самой сильной комбинацией выигрывает партию и получает всю сумму, содержащуюся в банке. В случае если несколько игроков обладают комбинациями равной силы, они делят сумму, содержащуюся в банке, поровну. Комбинация в покере всегда состоит из пяти карт. У игроков есть только две карманные карты. Три другие можно выбрать из общих карт стола. При составлении комбинации не обязательно использовать карманные карты.

Колода карт состоит из 52 карт. Каждая карта представляется набором

$$\{s, r\}, s \in \{1, 2, 3, 4\}, r \in \{1 \ldots 13\}$$

Параметр s соответствует масти карты, параметр r обозначает ранг карты.

В табл. приведен список возможных карточных комбинаций, отсортированный по старшинству (более сильные комбинации находятся выше).

Название	Пример	Описание
Стрейт-флеш	A♡K♡Q♡J♡T♡	Масти всех пяти карт совпадают и карты идут по порядку по рангу.
Каре	A♡A♢A♧A♤4♤	Четыре карты имеют одинаковый ранг
Фул-хаус	A♤A♡A♢K♢K♧	Комбинация составляется из тройки и пары
Флеш	2♤5♤8♤T♤A♤	Все карты имею одинаковую масть
Стрейт	3♢4♤5♡6♡7♢	Карты располагаются по порядку по

		рангу
Тройка	5♡5♤5♢K♤J♡	Три карты имеют одинаковый ранг
Две пары	2♡2♤A♢A♤8♡	Ранги карт двух пар равны
Пара	2♡2♤A♢T♤8♡	Две карты имеют одинаковый ранг
Старшая карта	K♡4♤5♤6♤7♤	Старшинство комбинации определяется картой с максимальным рангом

Таблица. Возможные комбинации карт в игре покер

Одинаковые комбинации различаются по старшинству карт, составляющих их.

Во время каждого из раундов торговли игроки с заранее заданной очередностью совершают одно из следующих действий:

- Ставка (bet). Если перед игроком никто не делал ставок, он может сделать это первым. Размер ставки определяется на основе текущей стадии игры.
- Уравнивание (call). Если перед ходом игрока уже были сделаны ставки, тогда он может уравнять их. Для совершения данного действия игроку необходимо положить в банк сумму равную разности между максимальной суммой, внесенной в банк каким-либо из оппонентов за текущий раунд торговли, и суммой, внесенной им за текущий раунд торговли.
- Повышение (raise). Ставка может быть повышена на сумму, определяемую на основе текущей стадии игры. Игрок, сделавший повышение, тем самым увеличивает цену для остальных игроков, которую они должны заплатить, чтобы продолжать играть дальше. В течение одного раунда торговли не может быть совершено более трех повышений ставок.

- Сброс карт (fold). Если игрок не хочет продолжать игру со своими картами ввиду того, что ему нужно уравнять ставки противников, он может сбросить их и выйти из розыгрыша текущей партии.
- Передача хода (check). Если перед игроком никто не ставил, он может передать ход дальше. Это означает, что он не делает ставку и передает право действия следующему игроку.

После того как все игроки совершили по крайней мере одно действие и не осталось не уравненных ставок каким-либо из не вышедших из игры игроков раунд торговли заканчивается.

1.2.3. Вариации «Техасского Холдема»

Существует несколько вариаций «Техасского Холдема». Различают «Техасский Холдем» с фиксированными ставками(limit) и неограниченными ставками(no-limit). В игре с фиксированными ставками размеры ставок и повышений всегда строго определены. На первых двух стадиях, префлопе и флопе, размеры ставок и повышений равны размеру большой обязательной ставки, на последних двух – в два раза больше размера большой обязательной ставки. В игре с неограниченными ставками ставка или повышение могут иметь любой размер, не меньший размера предыдущей ставки.

«Техасский Холдем» классифицируют по числу игроков принимающих участие в игре:

- игра двух лиц;
- игра более чем двух лиц.

В данной работе рассматривается «Техасский Холдем» для двух игроков с фиксированными ставками.

1.2.4. Свойства игры

«Техасский Холдем» обладает следующими важными свойствами:

- иерархическая игра – действия осуществляются последовательно;

- игра с неполной информацией – игроки не знают закрытых карт своих оппонентов, текущее состояние внутри информационного множества определяется картами соперника.
- игра с полной памятью;
- стохастическая игра – сдача общих карт происходит случайным образом;
- игра с нулевой суммой;
- игра с частично обозримой информацией – информация о картах игроков, вышедших из игры, никогда не станет общеизвестной;
- цель игры – выиграть как можно большую сумму.

Дерево игры в экстенсивной форме содержит приблизительно $3{,}16 * 10^{17}$ нетерминальных вершин и $3{,}19 * 10^{14}$ информационных множеств[8].

1.3. Обзор существующих исследований

В данном разделе приводится описание существующих исследований по построению стратегий игры «Покер Техасский Холдем», рассматриваются разработанные программы, использованные для их построения методы и их основные характеристики.

1.3.1. Абстракция игры

Одной из основных сложностей при анализе игр является размер дерева игры. Данная проблема решается с помощью построения абстрактной игры, содержащей меньшее число состояний, но сохраняющей основные стратегические свойства исходной игры. Стратегии, построенные для абстрактной игры, в общем случае не являются оптимальными в исходной игре, но зачастую показывают высокие результаты и в полной игре.

Один из общих подходов к построению абстрактной игры сохраняющей свойства исходной – объединение изоморфных поддеревьев вершин дерева игры[9]. Применение данного подхода к игре «Покер Техасский Холдем» описано в работе [10]. В целом анализ свойств конкретной игры позволяет

строить абстракции меньшего размера, сохраняющие точность ее описания, чем данный общий подход. В данном разделе описаны техники, использованные другими исследователями для сокращения размера дерева игры «Покер Техасский Холдем».

В задаче моделирования поведения агента (игрока) первостепенную роль играет определение вероятностей действий моделируемого объекта в различных состояниях игры. Применения большинства описанных ниже подходов не всегда является корректным с точки зрения решения задачи моделирования в виду того что в стратегически одинаковых состояниях агент может принимать различные решения. Метод позволяющий строить абстракцию игры «Покер Техасский Холдем», применимую для моделирования конкретного агента, описан в работе [11].

1.3.1.1. Метрики сил рук

Наиболее распространенная техника сокращения размера дерева игры – разбиение пар закрытых карт на корзины[12]. На каждой стадии игры все возможные пары закрытых карт раскладываются по корзинам таким образом, чтобы в одну корзину попадали руки со схожими стратегически свойствами. Корзины представляют вершины в абстрактной игре. Один из подходов для применения данного методы – проводить разбиение на основе силы руки. Руки, которые потенциально позволяют выиграть большее число ставок, называют более сильными и наоборот. На силу руки влияет множество факторов, в первую очередь то, как часто комбинация окажется сильнейшей на стадии вскрытия карт. При этом стоит различать руки, которые могут потенциально усилиться до очень сильной комбинации, и руки, которые имеют малую вероятность усилиться, так называемые готовые руки. Первые в случае, если на стол не выйдет карта, увеличивающая вероятность победы, могут быть сброшены, не требуя при этом дополнительных расходов. Ниже описаны наиболее распространенные методы для расчета сил рук:

- $E[HS]$ метрика. При использовании данного подхода сила руки рассчитывается как вероятность победы на стадии вскрытия карт против случайной пары карт. Одним из основных недостатков метрики $E[HS]$ является невозможность разграничить рук с различной потенциальной силой. Для его частичного нивелирования выделяются специальные корзины для рук, обладающих высоким потенциалом.
- $E[HS]^a$ метрика. Метрика $E[HS]^a$ присваивает рукам с большим потенциалом большее значение. Вычисление силы руки согласно данной метрике происходит следующим образом:
    - набор общих карт дополняется всеми возможными способами до пяти карт;
    - для каждого варианта дополнения набора общих карт к итоговому значению прибавляется вероятность победы против случайной пары карт, возведенная в степень a.

Рассмотрим применение метрики $E[HS]^2$ на примере. Рука a имеет в половине случаев вероятность победы 90% (0.9), а в половине 10% (0.1), итоговое значение:

$$E[HS]^2(a) = (0.9 * 0.9 + 0.1 * 0.1) * 0.5 = 0.41$$

Для руки b вероятность победы во всех случаях равна 50% (0.5), сила руки:

$$E[HS]^2(b) = 0.5 * 0.5 = 0.25$$

Для разбиения рук по корзинам используются следующие подходы:

- Каждой корзине сопоставляется интервал сил рук, руки определяются в корзину с соответствующим интервалом.
- Корзины нумеруются по порядку от 1 до N. В корзину с номером 1 определяются $\frac{100}{N}$% рук, начиная с руки с наименьшей силой. В корзину с номером 2 – следующие по силе $\frac{100}{N}$% рук и т.д.

- Вместо одного набора корзин используются несколько наборов. По силе руки на предыдущей стадии игры выбирается набор корзин, по силе руки на текущей стадии – корзина из выбранного набора.

1.3.1.2. Ограничение максимального числа повышений

Данная техника ограничивает максимальное число возможных действий за один раунд торговли. В «Техасском Холдеме» с фиксированными ставками число повышений на каждом раунде торговли ограничивается тремя, вместо четырех возможных в полной игре. В работе[13] была построена ϵ -равновесная по Нэшу стратегия с ограничением числа повышений на первой стадии двумя ставками и тремя ставками на остальных стадиях. Построенная стратегия оказалась уязвимой на 11 малых ставок со 100 игр в собственной абстракции и 27 малых ставок со 100 игр в полной игре[13].

1.3.1.3. Изоморфизм карт

Объединение состояний для пар карт с изоморфными мастями не приводит к потере информации в виду того, что такие состояния стратегически не отличаются. Для примера, на префлопе состояния после одной последовательности действий игроков для карт A♡K♡ и A♢K♢ могут быть объединены.

1.3.2. Стратегии с фиксированными правилами

Исторически первым классом стратегий были стратегии с фиксированными правилами. Стратегии данного типа описываются набором условий, в зависимости от выполнения которых, принимается то или иное решение.

Наиболее известным представителем данного класса стратегий в покере является программа Poki[8], разработанная исследовательской группой университета Альберты[14]. Изначально Poki была рассчитана для игры «Покер Техасский Холдем» с фиксированными ставками для десяти человек. Позднее были представлены версии, адаптированные к игре двух лиц. Использованные методы:

- предсказание следующего действия оппонента на основе фиксированного набора правил;
- вычисление силы руки и потенциала руки, являющихся входными параметрами для формулы определяющей возможные действия;
- использование метода Монте-Карло для выбора действия, имеющего наибольшее математическое ожидание.

Моделирующая поведение оппонента программа BRPlayer[15], получила ожидаемое значение выигрыша в игре с Poki в 60.1 малой ставки за 100 партий.

1.3.3. Оптимальные стратегии

В разделе 1.1.3 настоящей работы описана концепция равновесия по Нэшу. Ежегодно на соревнованиях по компьютерному покеру[18] наиболее широко представленным классом стратегий являются ϵ-оптимальные по Нэшу стратегии. На данный момент существует несколько подходов построения таких стратегий, в данном разделе описываются три из них.

1.3.3.1. Группа стратегий PsOpti

В 1994 году Коллером, Мегридо и фон Штенгелем был предложен алгоритм[16] для построения равновесия по Нэшу в играх в последовательной форме (преобразование экстенсивной формы) со сложностью $O(|\mathcal{I}|)$ ($|\mathcal{I}|$ – число информационных множеств). Алгоритм основан на решении задачи линейного программирования. С помощью данного метода был рассчитан набор стратегий PsOpti[8]: PsOpti4, PsOpti6, PsOpti7. Для сокращения размера игры использовалась техника разбиения по корзинам и ограничения максимального числа ставок. Используя различное число корзин, авторы получили несколько приближений равновесия по Нэшу.

1.3.3.2. CFR агенты

С помощью техники Counterfactual Regret Minimization[17] в работе [12] построена ϵ-оптимальная по Нэшу стратегия. Главным преимуществом данного метода является возможность реализации алгоритма на распределенных системах, что позволяет использовать более мощную абстракцию. Для

построения абстракции использовалась техника разбиения по корзинам. Агент, использующий данную стратегию, вошел в команду агентов, занявшую в 2008 году первое место на соревнованиях по компьютерному покеру.

1.3.3.3. Фиктивная игра

В 2008 году на соревнованиях по компьютерному покеру второе место заняла программа FellOmen2. В 2007 году ее предшественница, программа INOT, на соревнованиях по компьютерному покеру заняла второе место. Промежуточной версией является программа FellOmen. Данные программы использовали стратегии, построенные с помощью метода, называемого «Фиктивная игра»[19]. Суть метода заключается в итеративном построении стратегий, являющихся стратегиями наказания для смешанной стратегии, составленной из стратегий, полученных на предыдущих шагах. В общем случае не гарантируется сходимость метода к оптимальной по Нэшу стратегии.

1.3.4. Стратегии наказания и моделирование поведения

Общим подходом для построения данных типов стратегий является использованием алгоритма Expectimax[20], описание которого приводится в главе 2. Различием в программах, реализующих стратегии этого класса, является специфика реализации алгоритма Expectimax и использовании разных подходов к моделированию поведения игрока.

Модели игроков можно разбить на два класса:

- Стратегические модели, в моделях данного класса строятся функции распределения вероятности действия игрока от вершины дерева для каждого информационного множества игры. Основной проблемой при построении модели данного класса для игры «Покер Техасский Холдем» является то, что не всегда информация о закрытых картах доступна после окончания партии. К данному классу относятся модели, построенные на основе байесовских сетей доверия[21] и нейронных сетей[8]. Полученные в данных работах результаты не

подтвердили применимость этих методов к моделированию игрока в игре «Покер Техасский Холдем».

- Модели, основанные на наблюдениях. В моделях данного класса для каждого информационного множества, моделирующего игрока, вычисляются вероятности совершения игроком определенного действия. В отличие от моделей стратегического класса модели этого класса не обрабатывают вероятность нахождения в конкретной вершине внутри информационного множества, что приводит к другому способу обработки терминальных вершин при осуществлении поиска по дереву.

1.3.4.1. BRPlayer

Программа BRPlayer[15] использует алгоритм Expectimax для построения стратегии и модель, основанную на наблюдениях. Сравнительный анализ программы, построенной с помощью методов описанных в настоящей работе, с программой BRPlayer приводится в главе 3.

1.3.4.2. Частотная стратегия наказания

Частотная стратегия наказания (Frequentist Best Response) была предложена в работе [12]. Данная стратегия построена с помощью алгоритма Expectimax и алгоритма построения модели стратегического класса. Данному алгоритму требуется информация о закрытых картах игроков в каждой партии, используемой для обучения модели. Общей идей является использование абстрактной версии игры и вычисление вероятности действий для каждого информационного множества моделируемого игрока с помощью ведения счетчиков числа увиденных действий для каждого информационного множества. Абстракция игры задается как параметр алгоритма человеком. Сравнительный анализ метода моделирования, описанного в настоящей работе, с методом, используемым в данной программе, приводится в главе 3.

1.3.4.3. Безопасная стратегия наказания

В работе [12] описывается метод построения безопасной стратегии наказания (Restricted Nash Response), то есть ϵ-равновесной по Нэшу стратегии, использующей уязвимости в игре оппонента. Данный метод является комбинацией методов Counterfactual Regret Minimization и частотной стратегии наказания.

1.3.5. Команды агентов

Широко распространенным подходом построения программ, играющих в покер, является использование команд стратегий. Перед началом каждой партии независимым агентом выбирается стратегия для розыгрыша партии. Программа Hyperborean6[12], построенная с помощью такого подхода, в 2006 году на соревнованиях по компьютерному покеру заняла первое место. В качестве стратегий, составляющих команду, использовались PsOpti4 и PsOpti6.

ГЛАВА 2. ПОСТРОЕНИЕ МОДЕЛИ И СТРАТЕГИИ

2.1. Алгоритм Expectimax

Алгоритм Expectimax[20] решает задачу построения стратегии наказания. Впервые алгоритм был предложен Доналдом Михе в 1966 году для учета вершин случая в стохастических играх с полной информацией. Для построения стратегии рекурсивно рассчитывается математическое ожидание выигрыша для каждой из вершин дерева игры. После чего выбирается действие с максимальным значением ожидаемого выигрыша. В данном разделе приводится описание алгоритма и его адаптация к игре «Покер Техасский Холдем».

2.1.1. Применение алгоритма в общем случае

Рассмотрим игру Г двух лиц в экстенсивной форме $< N, H, P, f_c, \mathcal{I}_{i \in N}, U >$. $N -$ множество игроков, $H -$ множество вершин, $P -$ функция, определяющая игрока совершающего ход для каждой вершины из H, $f_c -$ функция определяющая вероятностную меру для вершин, в которых природа игры, меняет текущее состояние, $\mathcal{I}_{i \in N} -$ информационные множества всех игроков, $U -$функция выигрыша.

Рассмотрим действия классического варианта алгоритма в зависимости от типа вершины (стратегия строится для игрока σ_{br}, оппонент – σ_m):

- Для вершины принятия решения h, если $P(h) = \sigma_{br}$ ожидаемое значение выигрыша равно $ev(h) = max_{a \in A(h)} ev((h, a))$.
- Если $P(h) \neq \sigma_{br}$, $ev(h) = \sum_{a \in A(h)} p(a|h) * ev\big((h, a)\big)$, где $p(a|h)$ обозначает вероятность возникновения события a в вершине $h, ev((h, a))$ обозначает ожидаемое значение выигрыша игрока σ_{br} в случае достижения вершины (h, a).
- Для терминальной вершины $t, ev(t) = U(\sigma_{br}, t)$.

Модификация алгоритма для игр с неполной информацией:

- Для информационного множества $I_{\sigma_{br}} \in \mathcal{I}_{\sigma_{br}}$ игрока σ_{br}

$$ev\left(I_{\sigma_{br}}\right) = max_{a \in A\left(I_{\sigma_{br}}\right)} \left[\sum_{h \in I_{\sigma_{br}}} ev\left((h,a)\right) \pi_{\sigma_m}\left(h|I_{\sigma_{br}}\right) \right],$$

где $\pi_{\sigma_m}\left(h|I_{\sigma_{br}}\right)$ обозначает вероятность нахождения в вершине $h \in I_{\sigma_{br}}$, в случае достижения информационного множество $I_{\sigma_{br}}$. Выбрать действие, ведущее к достижению максимального ожидаемого значения выигрыша.

- Для вершины $h: P(h) \neq \sigma_{br}$ ожидаемое значение выигрыша

$$ev(h) = \sum_{a \in A(h)} p(a|h) * ev((h,a)),$$

если $P\left((h,a)\right) = \sigma_{br}$, то

$$ev\left((h,a)\right) = ev(I_c): I_c \in \mathcal{I}_{\sigma_{br}}, (h,a) \in I_c.$$

- Для терминальной вершины t ожидаемое значение выигрыша $ev(t) = U(\sigma_{br}, t)$.

На рис. 3 приведен пример дерева поиска алгоритма Expectimax. Ромбом обозначена вершина, в которой состояние игры равновероятно меняется на одну из двух других вершин. Серым цветом обозначены вершины принятия решения первого игрока, для которого используется Expectimax, белым цветом – второго игрока. Вершины внутри одного информационного множества обведены. Терминальные вершины обозначены прямоугольниками. Значение ожидаемого выигрыша в случае достижения вершины, записано внутри фигуры обозначающей ее. Значение ожидаемого выигрыша в случае достижения информационного множества первого игрока записано внутри фигуры объединяющей вершины информационного множества. Стрелки обозначают события, переводящие игру из одной вершины в другую. На стрелках соответствующих событиям о принятии решения вторым игроком или

случайным процессом записана вероятность возникновения события. На утолщенных стрелках записан ожидаемый выигрыш в случае принятия соответствующего решения.

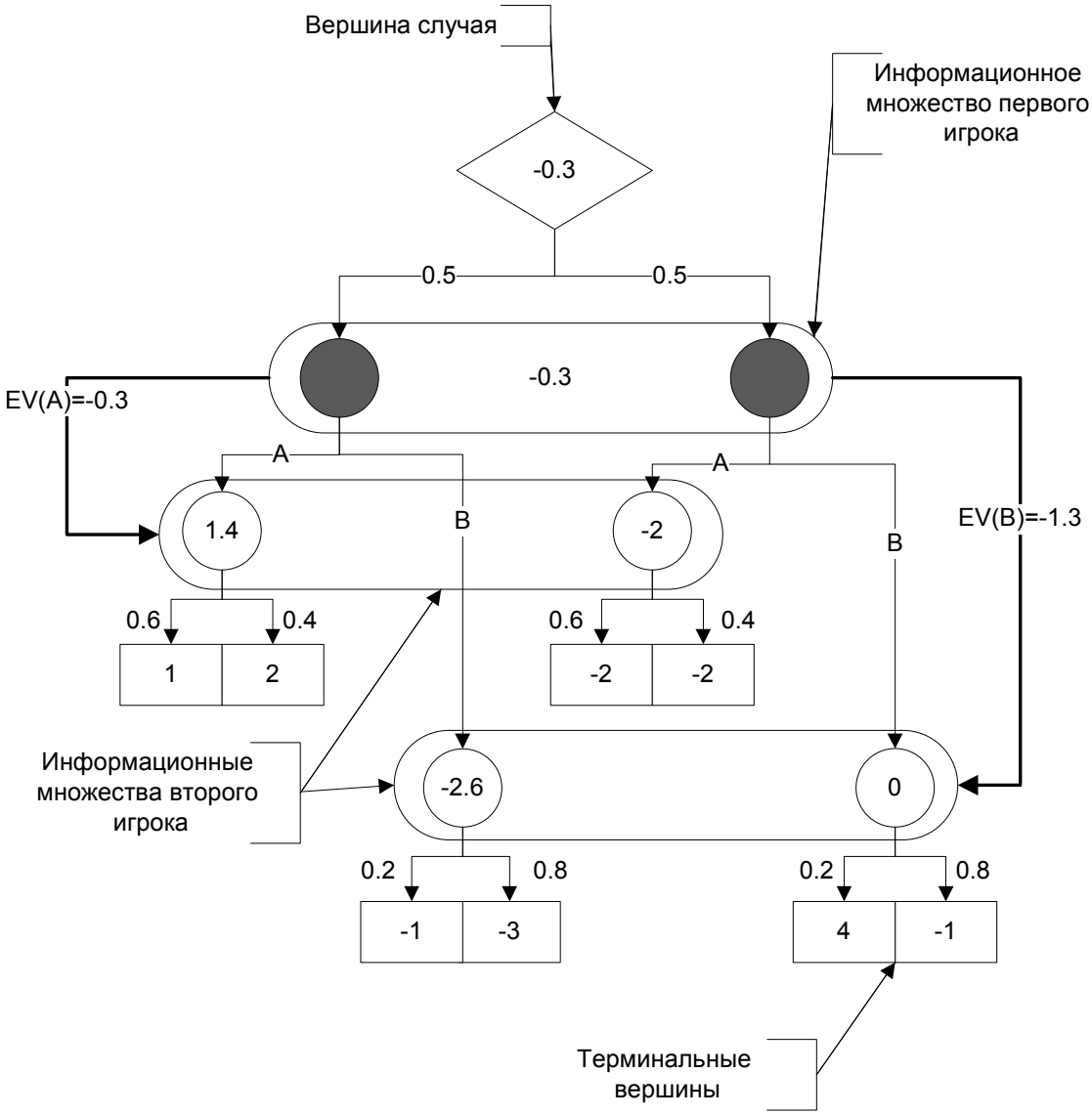

Рис. 3. Дерево поиска алгоритма Expectimax

2.1.2. Адаптация алгоритма к игре «Покер Техасский Холдем»

Для применения модификации алгоритма Expectimax, описанной в предыдущем разделе, необходимо вычисление следующих значений:

- вероятность $p(a|h)$ возникновения события a в вершине h;

- при достижении информационного множества $I: P(I) = \sigma_{br}$, для каждой вершины $h \in I$ вероятность $\pi_{\sigma_m}(h|I)$ нахождения игры в вершине h.

В настоящей работе вычисление значения $p(a|h)$ производилось с помощью моделирования поведения игрока. Значение $\pi_{\sigma_m}(h|I)$ может быть вычислено по следующей формуле:

$$h = (a^i)_1^K, \qquad \pi_{\sigma_m}(h|I) = \frac{\prod_{k:k\leq K, P\left((a^i)_1^k\right)\neq\sigma_{br}} p(a^k|(a^i)_1^{k-1})}{\sum_{k:k\leq K, P\left((a^i)_1^k\right)\neq\sigma_{br}} p(a^k|(a^i)_1^{k-1})}$$

В игре «Покер Техасский Холдем» события, соответствующие появлению карт на столе на каждой стадии игры, равновероятны (за исключением карт, розданных ранее, вероятность их появления равна нулю). С точки зрения участника игры вероятность появления определенной карты зависит от вероятности обладания его соперниками этой картой. В данной работе обработка вершин, соответствующих смене стадии игры, производилась способом отличным от используемого в алгоритме Expectimax. С точки зрения игрока в дереве полной игры, после одной истории действий (следует помнить, что игрок может не знать всех произошедших действий) существует несколько вершин случая, соответствующих сдаче очередной карты. Каждая из таких вершин соответствует определенным закрытым картам оппонента. Различные вершины имеют различные распределения появления карт на столе. При реализации алгоритма Expectimax автор объединил эти вершины в одну, также объединив при этом распределения вероятностей, соответствующие вершинам. Для совершения данной операции необходима информация о том, какая вероятность обладания соперником каждой картой. При наличии знания вероятности обладания оппонентом конкретной парой закрытых карт это значение может быть легко вычислено. После сдачи игрокам закрытых карт все пары карт $< C1, C2 >$ равновероятны. После возникновения события a в множестве состояний I (конкретное состояния неизвестно) данные о

вероятности наличия у игрока пары карт $< C1, C2 >$ могут быть обновлены с помощью формулы Байеса:

$$p(< C1, C2 > |a, I) = \frac{p(< C1, C2 >)p(a| < C1, C2 >, I)}{p(a|I)}$$

Заметим, что после внесенного изменения в граф игры, можно установить взаимно однозначное соответствие между вершинами любого информационного множества и закрытыми картами соперника.

2.2. Построение модели игрока

В данном разделе описывается алгоритм построения модели оппонента. Общая идея алгоритма – построение распределений вероятностей действий для каждого информационного множества дерева игры. Модели такого класса называются стратегическими моделями.

2.2.1. Использование абстракции игры

Важным параметром системы моделирующей поведение оппонента является скорость обучения – число партий необходимых для достижения достаточной точности предсказания. Принимая во внимание число информационных множеств, в которых решение принимает один из игроков в рассматриваемой нами игре, для того, чтобы увидеть хотя бы одно действие из каждого состояния принятия решения игрока, потребовались бы миллиарды записей партий. На практике распределения вероятностей действий игрока во многих состояниях если и не совпадают, то близки друг к другу. В виду этого одним из способов ускорения построения модели является выделение факторов влияющих на принимаемые моделируемым объектом решения и объединение распределений для состояний (информационных множеств) описывающихся одинаковым набором факторов. Под фактором понимается скалярная функция от состояния игры. В данной работе используются абстракции, построенные с помощью использования фиксированного набора заранее определенных факторов.

2.2.2. Структура модели

Пусть моделируемый игрок – s, а моделирующий игрок – t.

Модель игрока состоит из следующих компонент:

- абстракции игры – функция $f: \mathcal{I}_s \to \mathcal{I}_a$, где элемент I_a множества $\mathcal{I}_a$, представляет абстрактное состояние, для которого из $f(I_1) = I_a$ и $f(I_2) = I_a \Rightarrow A(I_1) = A(I_2)$;
- метрики ρ, заданной на $f^{-1}(I_a)$ для каждого I_a, определяющей насколько стратегически близки состояния исходной игры;
- $\mathcal{F}$ – функции распределения вероятностей действий заданной на $f^{-1}(I_a)$ для каждого абстрактного состояния I_a.

2.2.3. Метрика состояний

В данной работе была использована фиксированная метрика, расстояние между двумя вершинами определялось по следующей формуле:

$$\rho(h1, h2) = |E[HS]^{a_{h1}}(< C1, C2 >_{h1}) - E[HS]^{a_{h2}}(< C1, C2 >_{h2})|,$$

Запись $< C1, C2 >_h$ обозначает соответствующую вершине h пару закрытых карт оппонента $< C1, C2 >$, значения a_{h1} и a_{h2} определяются в зависимости от стадии игры, которой соответствуют состояния $h1$ и $h2$. Стоит отметить, что множество значений данной метрики содержится в отрезке $[0; 1]$.

Основное назначение метрики – распространение информации во время обучения с состояний, в которых было увидено действие, на другие состояния. Рассмотрим, для примера, две ситуации:

Ситуация 1:

Первый игрок находится на позиции дилера.

Закрытые карты первого игрока: A♤A♡

Последовательность действий на первой стадии игры:

1. первый игрок делает ставку;

2. второй игрок уравнивает ставку.

На стол выходят карты: A♢A♧K♤

На второй стадии второй игрок первым своим действием передает ход. Ход должен совершить первый игрок.

Ситуация 2:

Первый игрок находится на позиции дилера.

Закрытые карты первого игрока: K♤K♡

Последовательность действий на первой стадии игры:

1. первый игрок делает ставку;
2. второй игрок уравнивает ставку.

На стол выходят карты: K♢K♧J♤

На второй стадии второй игрок первым своим действием передает ход. Ход должен совершить первый игрок.

Данные ситуации представляются различными состояниями в дереве игры. В каждой из ситуаций первый игрок имеет комбинацию (каре), с которой он выигрывает почти всегда. Разница в данных ситуациях лишь в наборах карт на столе и на руках у игроков. Предположим для простоты, что игрок использует чистую стратегию. В таком случае с большой вероятностью можно сказать, что какое бы действие игрок не совершил в первой ситуации, во второй он совершит то же действие, и наоборот. Данная вероятность зависит от стратегии игрока и в общем случае игрок может совершать различные действия в этих ситуациях.

Одной из причин использования абстракции является упрощение задания метрики (ввести метрику на всем множестве вершин исходной игры – сложная задача). Фактически, задание метрики на подмножестве множества вершин исходной игры, соответствующем одному абстрактному состоянию, означает,

что расстояние между вершинами, относящимися к разным абстрактным состояниям, бесконечно, то есть информация не может быть распространена с одного такого множества на другое.

2.2.4. Обучение в случае полной обозримости

Алгоритм обучения в выбранной вероятностной модели отличается для двух случаев: случая, когда информация о скрытых параметрах (закрытых картах оппонента) доступна после окончания игры и случая, когда данная информация недоступна.

Алгоритм обучения для случая, когда финальное состояние известно:

- Добавить в набор состояний L все состояния I^i, в которых оппонент принимал решение в текущей партии. Сформировать соответствующий набору L набор L_a абстрактных состояний, добавив в него элементы $I_a^i = f(I^i)$.
- Для каждой пары $I^i \in L, I_a^i \in L_a$ обновить функцию распределения вероятностей действий для всех $I' \in f^{-1}(I_a^i)$ следующим образом: $F_{I_a^i}(a, I') = F_c(a, I') + \frac{1}{\sigma * \sqrt{2\pi}} * e^{\frac{-\rho\left(I^i, I'\right)^2}{2*\sigma^2}}$, где $\sigma = \frac{\ln(n+1)}{n+1}$, n число обновлений $F_{I_a^i}$, включая текущее, a – действие, совершенное в состоянии I^i, F_c – предыдущая функция распределения вероятностей действий для абстрактного состояния I_a^i.
- Для каждого элемента $I_a^i \in L_a$ нормировать $F_{I_a^i}$.

За счет использование указанной выше метрики для игры «Покер Техасский Холдем» функции распределения вероятностей действий не представлялись на всем множестве $f^{-1}(I_a^i)$. Строились проекции данных функций на отрезок [0;1], представляющий возможные значения сил рук. При этом вместо функции распределения вероятностей действий, получается функция плотности распределения вероятности для каждого действия. Таким образом, функция плотности распределения вероятности действия фактически

апроксимируется суммой нормальных распределений, каждое из которых имеет центр в $E[HS]^a(<C1,C2>)$ и дисперсию $\sigma = \frac{\ln(n+1)}{n+1}$, где n число действий, совершенных оппонентом из данного абстрактного состояния за все время моделирования.

2.2.5. Обучение в случае частичной обозримости

Обучение в данном случае рассматривается только на примере игры «Покер Техасский Холдем» с использованием указанной выше метрики. Также от абстракции требуется выполнение следующего свойства: если для информационного множества I моделируемого игрока $f(I) = I_a$, то для любого информационного множества оппонента I', такого что данные множества неразличимы с точки зрения моделирующего игрока, $f(I') = I_a$. Поясним данное свойство: в момент принятия решения оппонентом игрок точно не знает, в каком информационном множестве находится его соперник, но знает множество, которому принадлежит данное информационное множество.

Основной проблемой при моделировании оппонента в игре «Покер Техасский Холдем» является недоступность информации о закрытых картах игрока после окончания каждой партии, более того, информация, о том какие карты игрок сбрасывает никогда не может быть получена, так как приватные карты становятся известными только на стадии вскрытия карт.

Обновления распределений вероятностей действий в данном случае производится с помощью следующего алгоритма:

- Для каждой пары карт $< C_1, C_2 >$ сформировать набор состояний $L_{<C1,C2>}$, соответствующий истории действий совершенных в игре, в случае если бы оппонент обладал данной парой карт.
- Для каждой пары карт $< C_1, C_2 >$ вычислить вероятность ее наличия у оппонента в данной партии по формуле: $p(< C_1, C_2 >) = \prod_{I \in L_{<C1,C2>}} F_{f(I)}(a, I)$, а – действие, совершенное оппонентом в данном состоянии. В общем случае вычисляемое на данном шаге

значение соответствует вероятности окончания текущей партии в конкретной терминальной вершине.

- Если в ходе текущей партии оппонент сбросил карты, изменить значение $p(< C_1, C_2 >)$ описанным ниже способом. Обозначим за I_f абстрактное состояние, в котором оппонент принял решение сбросить карты. $\mathrm{FoldPercent}$ – процент действий сброса карт для состояния I_f. Для каждой пары карт, попадающей в старшие по силе $100 - \mathrm{FoldPercent}$ процентов пар карт согласно метрике $E[HS]^a$, вероятность обладания оппонентом данной рукой в текущей партии принимается за 0. Вероятности обладания для остальных пар карт нормируются до суммы 1.
- Для каждого абстрактного состояния I_a, соответствующего состоянию I, в котором оппонент принимал решение в ходе партии построить функцию g: $[0; 1] \to [0; 1]$, по следующей формуле:

$$g(x) = \sum_{<C1,C2>:E[HS]^a\left(h_{<C1,C2>}\right)=x} p(< C1, C2 >),$$

$E[HS]^a\left(h_{<C1,C2>}\right)$ – сила руки $< C1, C2 >$ на момент совершения действия из состояния I_a. Интерполировать функцию g на отрезке [0;1]. Нормировать g так, чтобы

$$\int_0^1 g(x)dx = 1.$$

- Для каждого абстрактного состояния I_a, соответствующего состоянию I, в котором оппонент принял решение d, обновить функцию плотности распределения вероятности действия d $F_{I_a}^d$ следующим образом:

$$F_{I_a}^d(x) = F_c^d(x) + g(x),$$

где F_c^d – предыдущая функция плотности распределения вероятности действия d для данного абстрактного состояния.

- Для каждого абстрактного состояния I_a, соответствующего состоянию I, в котором оппонент принимал решение, нормировать $F_{I_a}^d$, где d совершенное действие.

2.2.6. Вычисление вероятности совершения действия

Вероятность совершения действия a моделируемым игроком из вершины h:

$$p(a,h) = F_{f(I(h))}(a, I(h)),$$

где $I(h)$ – информационное множество моделируемого игрока, которому принадлежит вершина h, $F_{f(I(h))}$ – функция распределения вероятностей действий, соответствующая $f\big(I(h)\big)$.

Если функции распределения проецируется на отрезок [0;1], то вероятность совершения действия a из вершины h:

$$p(a,h) = \frac{F^a_{f(I(h))}(E[HS]^{b_h}(< C1, C2 >_h))}{\sum_{a' \in A(h)} F^{a'}_{f(I(h))}(E[HS]^{b_h}(< C1, C2 >_h))},$$

где $I(h)$ – информационное множество моделируемого игрока, которому принадлежит вершина h, $F^{a'}_{f(I(h))}$ – функция плотности распределения вероятности действия a', соответствующая $f\big(I(h)\big)$, $< C1, C2 >_h$ – пара закрытых карт моделируемого игрока, соответствующая состоянию h, значение b_h зависит от стадии игры.

2.2.7. Построение пробной модели

В случае если число наблюдений в абстрактном состоянии недостаточно велико, чтобы считать построенное распределение удовлетворительным для использования, используется распределение по умолчанию. Данные распределения построены на основе экспертных знаний и в большинстве состояний сопоставляют сильным рукам большую вероятность совершения ставки или повышения, а слабым рукам большую вероятность сброса карт.

2.3. Построение агента для игры в покер

2.3.1. Практические аспекты применения алгоритма Expectimax

При применении алгоритма Expectimax совместно с моделью стратегического класса для игры «Покер Техасский Холдем» критичным является время для вычисления ожидаемых значений действий. На практике применить такой поиск для первой стадии игры (префлоп) не представляется возможным, в связи с этим для игры на данной стадии используется стратегия, состоящая из фиксированного набора правил. Для второй стадии игры (флоп) алгоритм поиска прерывается после обработки вершины, соответствующей сдаче пятой общей карты (переход на стадию ривер), а значение ожидаемого выигрыша вычисляется, исходя из предположения, что каждый из игроков внесет в банк еще по одной большой ставке.

2.3.2. Выбор абстракции

При тестировании использовались абстракции, содержащие заранее определенный набор факторов, составленный из следующих:

- текущая стадия игры;
- линия игры – последовательность действий совершенных игроками на текущей стадии игры;
- число ставок на первой стадии игры;
- число ставок на второй стадии игры;
- число ставок на третьей стадии игры;
- число ставок на четвертой стадии игры;
- число ставок на предыдущей стадии игры;
- владелец инициативы – идентификатор игрока совершившего последнее повышение (ставку) на предыдущей стадии игры.

2.3.3. Выбор метрик

Для определения силы руки оппонента на разных стадиях использовались различные метрики:

- на первой стадии использовалась метрика $E[HS]^2$;
- на второй стадии (флоп) использовалась метрика $E[HS]^{1.5}$;
- на третьей стадии (терн) стадии использовалась метрика $E[HS]^{1.25}$;
- на четвертой стадии (ривер) использовалась метрика $E[HS]$.

ГЛАВА 3. РЕЗУЛЬТАТЫ

В данном разделе приведены результаты тестирования разработанной программы. Для сравнения использовались программы, находящиеся в общем доступе. Тестирование проводилось в программе PokerAcademy[23], в которой существует возможность подключать стратегии в качестве плагинов.

3.1. Оценка результатов

Одной из трудностей в сравнении стратегий игры покер является необходимость в проведении большого числа партий для того, чтобы свести фактор удачи на нет. В 2005 году была опубликована работа[22], в которой, в частности, содержится эмпирическая оценка отклонение математического ожидания функции выигрыша:

$$\frac{6}{\sqrt{N}},$$

N – число сыгранных партий. Существуют методы, которые позволяют значительно сократить число необходимых партий для того, чтобы уточнить значение ожидаемого выигрыша[22]. В данной работе они не были использованы в виду особенностей программного окружения, в котором проводилось тестирование: исходные коды большинства конкурентоспособных покерных стратегий являются закрытыми, а фреймворки, предоставляющие доступ к стратегиям не реализуют такие методики.

3.2. PsOpti4

На чемпионате мира по компьютерному покеру в категории «Покер Техасский Холдем с фиксированными ставками. Игра один на один» в 2006 году первое место заняла программа Hyperborean06[12], разработанная группой исследователей компьютерного покера университета Альберты, описанная в разделе 1.3.6. Свободный доступ к исходному коду данной программы отсутствует. В данной программе несколько стратегий объединены в команду, одна из них, PsOpti4, доступна посредством программы PokerAcademy[23]. PsOpti4 является ε-эквилибриум стратегией, уязвимость данной программы

оценивается в 11 малых ставок за 100 партий в собственной абстракции и в 27 малых ставок за 100 партий в полной игре.

На рисунке. 4. изображен график функции выигрыша в игре с PsOpti4. Ожидаемое значение выигрыша равняется 13 ± 4 малых ставок за 100 партий. Информация о закрытых картах была доступна после каждой партии. Игра начиналась с моделью, построенной по 40 000 записей сыгранных партий, с использованием следующих факторов: текущая стадия игры, последовательность действий, совершенных игроками на текущей стадии игры, число ставок на предыдущей стадии игры, число ставок на первой стадии игры, владелец инициативы.

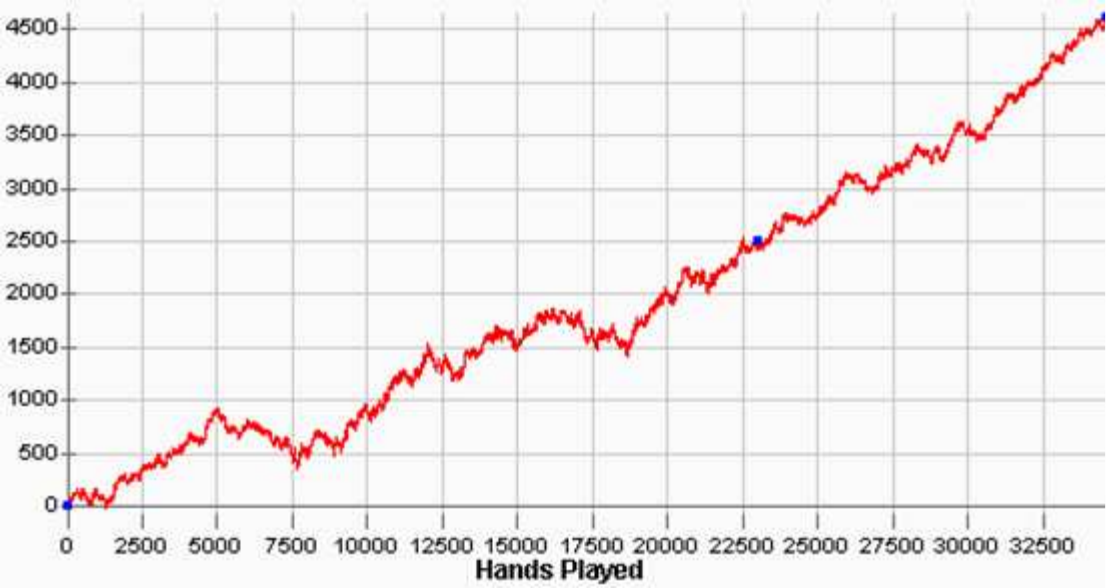

Рис. 4. График выигрыша в игре с PsOpti4

На рис. 5. изображен график функции выигрыша в игре с PsOpt4. Ожидаемое значение выигрыша равняется 5.7 ± 3 малых ставок за 100 партий. Информация о закрытых картах участников была доступна только в случае, если игра закончилась на стадии вскрытия карт. Игра начиналась с моделью, построенной по 40 000 записей сыгранных партий, с использованием следующих факторов: текущая стадия игры, последовательность действий,

совершенных игроками на текущей стадии игры, число ставок на предыдущей стадии игры, число ставок на первой стадии игры, владелец инициативы.

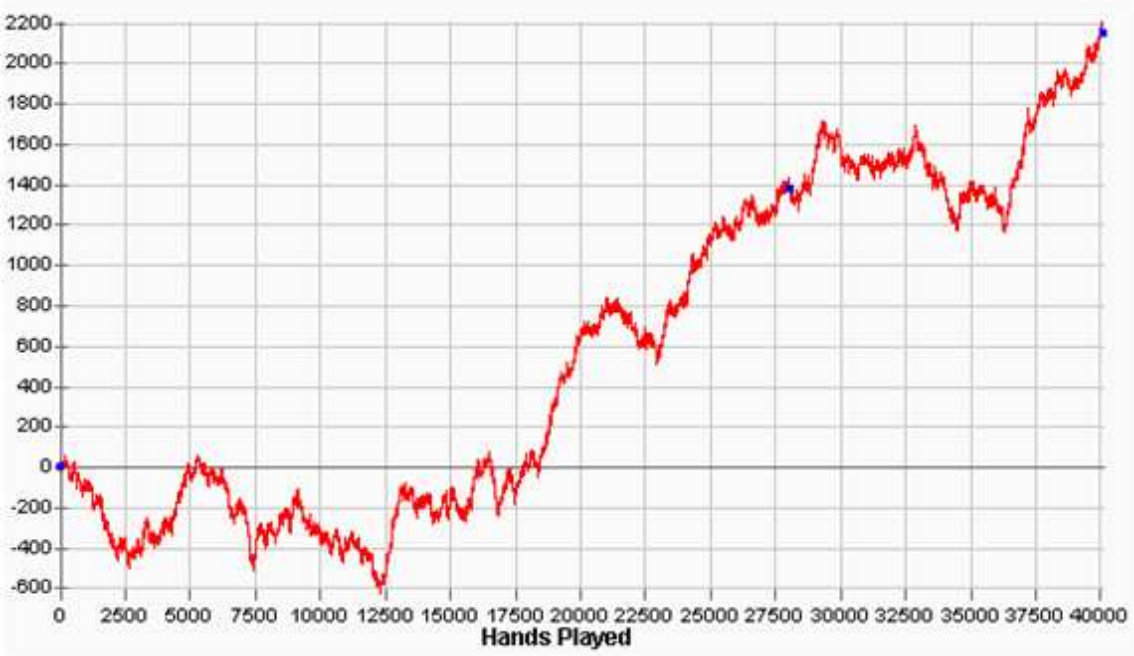

Рис. 5. График выигрыша в игре с PsOpti4

3.3. Poki

На рис. 6. изображен график функции выигрыша в игре с Poki, описание данной программы приводится в разделе 1.3.2. Ожидаемое значение выигрыша равняется 40 ± 6 малым ставкам за 100 игр. Информация о закрытых картах участников была доступна только в случае, если игра закончилась на стадии вскрытия карт. Предварительно модель не обучалась. Использовались следующие факторы: текущая стадия игры, последовательность действий, совершенных игроками на текущей стадии игры, число ставок на предыдущей стадии игры, владелец инициативы.

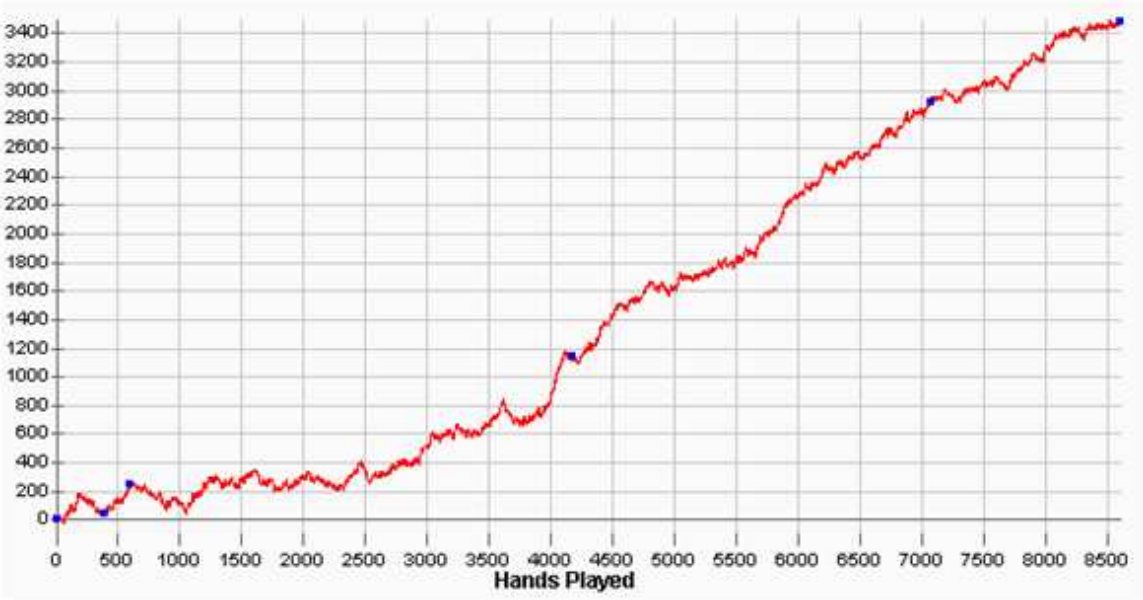

Рис. 6. График выигрыша в игре с Poki

3.4. FellOmen

Программа FellOmen является промежуточной версией между серебряным призером соревнований по компьютерному покеру 2007 года и серебряным призером соревнований по компьютерному покеру 2008 году (FellOmen2, краткое описание приведено в разделе 1.3.4.3).

На рис. 7. изображен график функции выигрыша в игре с FellOmen. Ожидаемое значение выигрыша равняется 1 ± 3 малых ставок за 100 игр. Информация о закрытых картах была доступна после каждой партии. Игра начиналась с моделью, построенной по 40 000 записей сыгранных партий, с использованием следующих факторов: текущая стадия игры, последовательность действий, совершенных игроками на текущей стадии игры, число ставок на предыдущей стадии игры, число ставок на первой стадии игры, владелец инициативы.

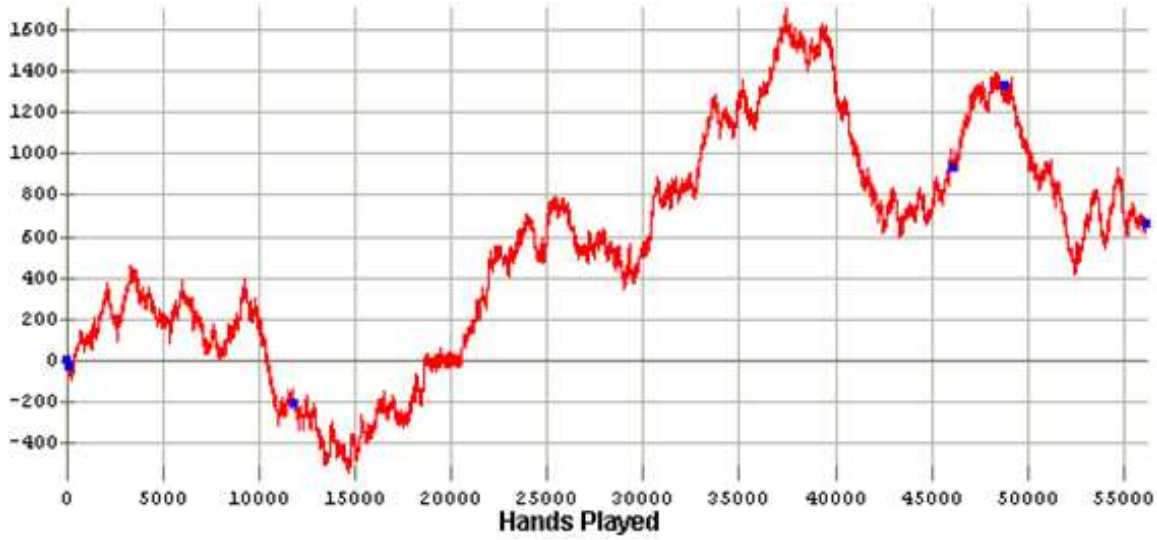

Рис. 7. График выигрыша в игре с FellOmen

На рис. 8. изображен график функции выигрыша в игре с FellOmen. Ожидаемое значение выигрыша равняется 0 ± 3 малые ставки за 100 игр. Информация о закрытых картах участников была доступна только в случае, если игра закончилась на стадии вскрытия карт. Игра начиналась с моделью, построенной по 40 000 записей сыгранных партий. Использовались следующие факторы: текущая стадия игры, последовательность действий совершенных игроками на текущей стадии игры, число ставок на предыдущей стадии игры, число ставок на текущей стадии игры, владелец инициативы.

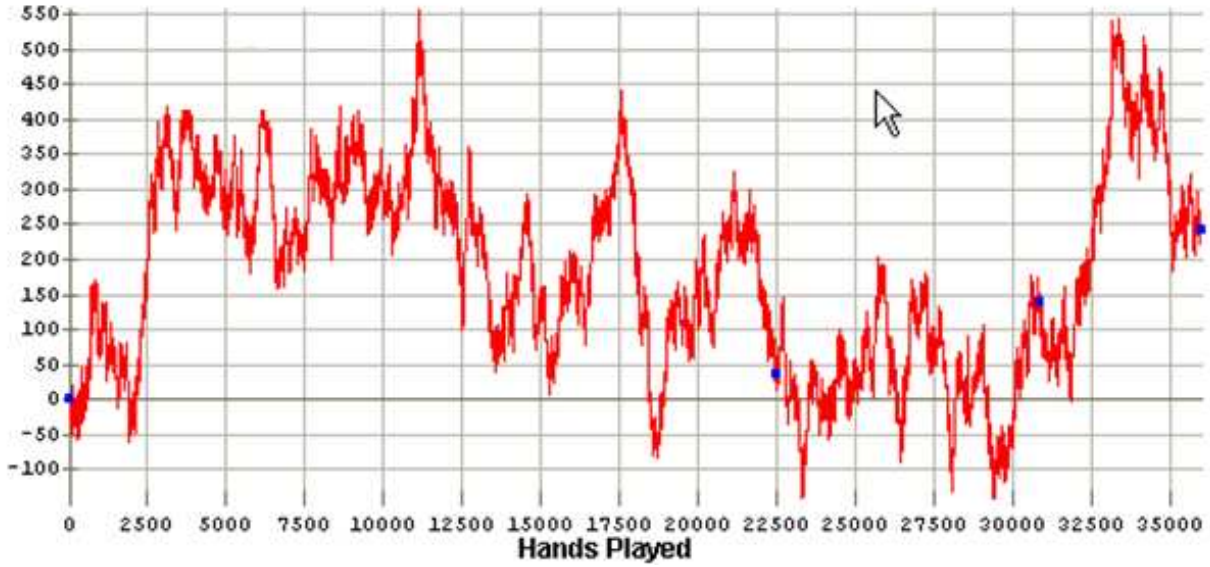

Рис. 8. График выигрыша в игре с FellOmen

3.5. Частотная стратегия наказания

На рис. 9 представлен график значения ожидаемого выигрыша агента построенного с помощью частотной стратегии наказания в зависимости от числа записей партий использованных для обучения, приведенный в работе [12]. Красным цветом отображен график функции выигрыша против PsOpti4. По оси абсцисс отложено число партий, использованных для обучения модели, по оси ординат – значение функции выигрыша в тысячных долях малой ставки. Напомним, что данной стратегии для обучения необходима информация о закрытых картах игроков после каждой партии.

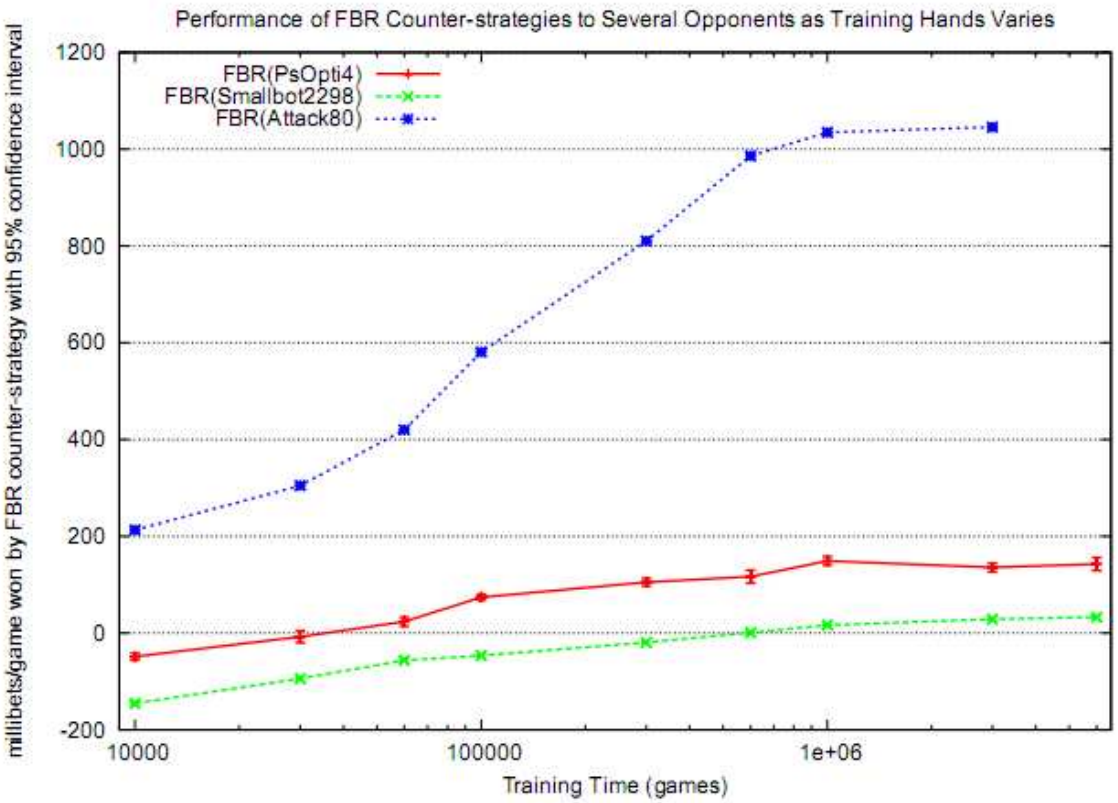

Рис. 9. График значения ожидаемого выигрыша частотной стратегии наказания в игре с различными оппонентами

При обучении на 60 000 партий, частотная стратегия наказания имеет значение ожидаемого выигрыша менее чем в 2.5 ± 0.1 малых ставок за 100 партий. В игре с тем же оппонентом (PsOpti4) стратегия, построенная при помощи методов описанных в настоящей работе, имеет значение ожидаемого выигрыша 13 ± 3 малых ставок за 100 партий в случае, если использовалось 40 000 партий для обучения и информация о закрытых картах игроков была доступна после каждой партии. В случае если информация о закрытых картах не была доступна, значение ожидаемого выигрыша стратегии, построенной при помощи методов описанных в данной работе, равно 5.7 ± 3 малых ставок со 100 партий. Стоит также отметить, что при обучении на 5 000 000 частотная стратегия наказания имеет значение ожидаемого выигрыша в игре с PsOpti4 в 13.7 ± 0.1 малую ставку за 100 партий. Данное значение ожидаемого выигрыша в игре с PsOpti4 является максимальным среди значений ожидаемого выигрыша

программ известных автору. Экспериментов с использованием большего числа записей партий для построения стратегии авторами работы [12] не проводилось.

Из данных результатов можно сделать вывод о том, что методы, описанные в настоящей работе, позволяют строить стратегию наказания, показывающую в игре с PsOpti4 результаты схожие с результатами частотной стратегии наказания за меньшее число партий.

3.6. BRPlayer

BRPlayer является единственной известной автору программой, моделирующей поведение соперника и не требующей для обучения информации о закрытых картах после каждой партии, достигнувшей положительного значения ожидаемого выигрыша в игре с PsOpti4.

На рис. 10 представлены графики выигрыша программы BRPlayer в игре с PsOpti4 в трех независимых матчах, приведенные в работе [15]. Данная программа строила модель оппонента, в течение матча. Напомним, что для построения модели программа BRPlayer использует информацию о закрытых картах оппонента, только в случае если игра достигла стадии вскрытия карт.

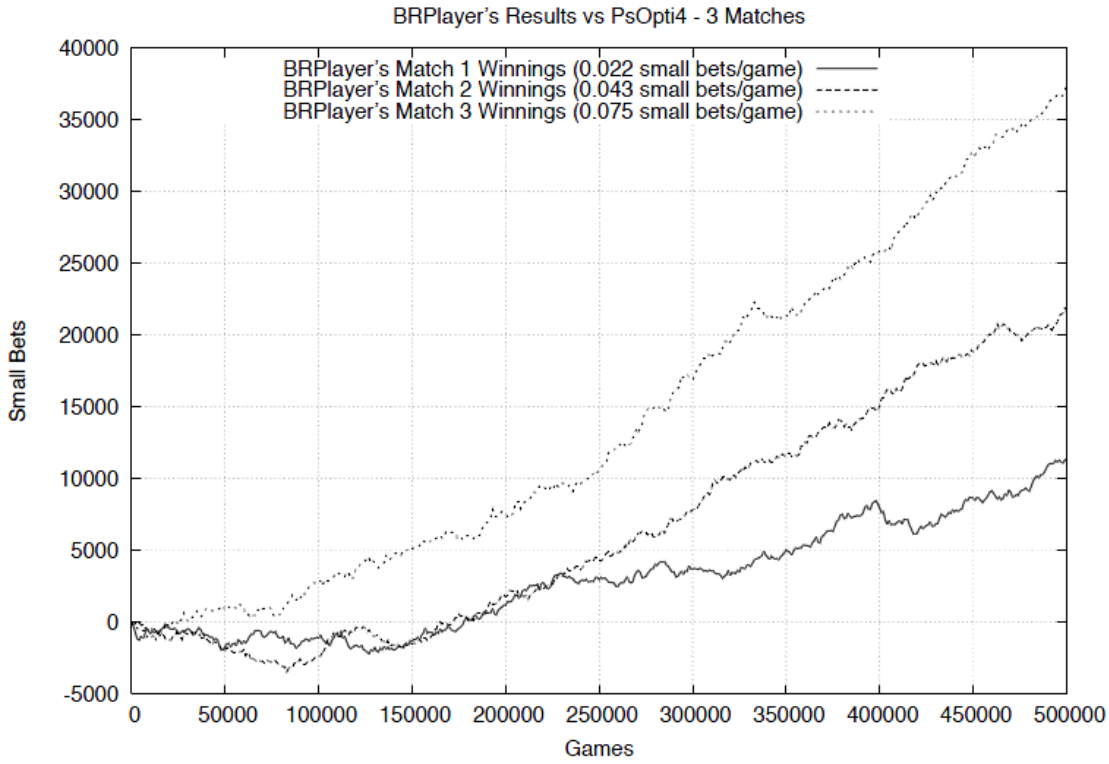

Рис. 10. Графики выигрыша BRPlayer в игре с PsOpti4

Из графика можно увидеть, что выигрыш программы BRPlayer за 150 000 партий при обучении на 50 000 партий не превосходит 2500 (суммарный выигрыш за отрезок от 50 000 до 100 000 партий) малых ставок, что соответствует значению ожидаемого выигрыша в 1.6 ± 2 малых ставок за 100 партий. Стратегия, построенная в данной работе, имеет значение ожидаемого выигрыша в 5.7 ± 3малых ставок за 100 партий, при обучении на 40 000 записей партий.

ЗАКЛЮЧЕНИЕ

Результаты квалификационной работы, выносимые на защиту:

- построена вероятностная модель для моделирования поведения игрока;
- разработан алгоритм обучения модели для игры «Покер Техасский Холдем»;
- алгоритм построения стратегии наказания адаптирован к разработанной вероятностной модели;
- разработано программное обеспечение, реализующее построенные алгоритмы, для игры «Покер Техасский Холдем» двух лиц;
- применимость разработанных алгоритмов обоснована результатами сравнения построенной стратегии со стратегиями, разработанными другими исследователями.

Возможные направления дальнейших исследований:

- модификация разработанной вероятностной модели для учета динамики изменения стратегии моделируемого агента;
- построение стратегии, совмещающей концепты стратегии наказания и оптимальной по Нэшу стратегии;
- разработка быстрого алгоритма построения стратегии наказания для игры «Покер Техасский Холдем» трех лиц.

ИСТОЧНИКИ

1. *Hsu F. H.* Behind Deep Blue: Building the Computer that Defeated the World Chess Champion. Princeton University Press, 2002.
2. *Schaeffer J., Lake R., Lu P., Bryant M.* CHINOOK: The world manmachine checkers champion. AI Magazine, 17(1):21–29, 1996.
3. *Nash J. F.* Non-cooperative games. Annals of Mathematics, 54:286–295, 1951.
4. *Osborne M., Rubenstein A.* A Course in Game Theory. The MIT Press, Cambridge, Massachusetts, 1994.
5. *Zinkevich M., Johanson M., Bowling M., Piccione C.* Regret minimization in games with incomplete information. In NIPS07, 2008.
6. *von Neumann J., Morgenstern O.* The Theory of Games and Economic Behavior. Princeton University Pres, 1947.
7. *Nash J., Shapley L.* A Simple Three-Person Poker Game. Annals of Mathematical Statistics, 1950.
8. *Billings D.* Algorithms and Assessment in Computer Poker. PhD thesis, University of Alberta, 2006.
9. *Gilpin A., Sandholm T.* Better automated abstraction techniques for imperfect information games, with application to texas hold’em poker. AAMAS’07, 2007.
10. *Gilpin A., Sandholm T.* A competitive texas hold’em poker player via automated abstraction and real-time equilibrium computation. In Proceedings of the Twenty-First Conference on Artificial Intelligence (AAAI-06), 2006.
11. *Терескин А.* Метод выделения факторов, влияющих на решения объекта моделирования, на примере игры «Покер Техасский Холдем», СПБГУ ИТМО, 2010.
12. *Johanson M.* Robust Strategies and Counter-Strategies: Building a Champion Level Computer Poker Player. University of Alberta, 2007.

13. *Zinkevich M., Bowling M., Burch N.* A new algorithm for generating strong strategies in massive zero-sum games. In Proceedings of the Twenty-Second Conference on Artificial Intelligence (AAAI-07), 2007.
14. http://poker.cs.ualberta.ca/
15. *Schauenberg T.* Opponent modeling and search in poker. Master's thesis, University of Alberta, 2006.
16. *Koller D., Megiddo N., von Stengel B.* Fast algorithms for finding randomized strategies in game trees. In 26th Annual ACM Symposium on the Theory of Computing, pages 750–759, 1994
17. *Zinkevich M., Johanson M., Bowling M., and Piccione C.* Regret minimization in games with incomplete information. Technical Report TR07-14, Department of Computing Science, University of Alberta, 2007.
18. http://www.computerpokercompetition.org/
19. *Brown G.W.* Iterative Solutions of Games by Fictitious Play. In Activity Analysis of Production and Allocation, 1951.
20. *Michie D.* Game-playing and game-learning automata. In L. Fox (ed.), Advances in Programming and Non-Numerical Computation, pp. 183-200, 1966.
21. *Nicholson E., Korb B.* Bayesian Poker. Monash University, 2004.
22. *Billings D., Kan M.* A tool for the direct assessment of poker decisions. In The International Association of Computer Games Journal, 2006.
23. http://www.poker-academy.com/