7. ПОИСК НА ИГРОВЫХ ДЕРЕВЬЯХ

7.1. Игры с полной информацией

В интеллектуальных играх соревнование между участниками заключается в том, что они поочередно принимают решения, не зная, какое следующее решение принимает противник. Нас будут интересовать те игры, в которых либо один игрок выигрывает (а другой проигрывает), либо между ними заключается ничья. Таковы игры в шашки, крестики-нолики, шахматы и им подобные. Классический подход, реализуемый интеллектуальным агентом для решения этой задачи, состоит в прогнозировании последующих своих ходов и ответных ходов противника: если я сделаю такой ход, тогда противник может ответить тем или иным ходом, на каждый из этих ходов в моем распоряжении имеются такие-то ответы и т.д. В итоге можно построить дерево (или граф) допустимых ходов и возможных игровых позиций. В результате анализа, взвесив все «за» и «против» той или иной позиции, агент делает ход (ход первого уровня), который ему представляется наилучшим.

Эти идеи можно проиллюстрировать на примере следующей игры типа «ним». Перед двумя игроками в одну кучку сложены некоторые предметы, допустим, что это монеты. Первый игрок делит исходную кучку на две обязательно неравные части. Далее игроки по очереди делят на неравные части одну из получающихся кучек (выбирая ее каждый раз по своему усмотрению). Игра продолжается до тех пор, пока во всех кучках не окажется по одной или по две монеты, после чего продолжение игры станет невозможным. Проигрывает тот, кто первым не сможет сделать свой ход. Назовем наших игроков (агентов) MAX и MIN, и пусть первый ход делает MAX.

Рассмотрим ситуацию, когда в кучке 7 монет. Для этой игры состояния среды можно представить неупорядоченной последовательностью чисел, представляющих число монет в разных кучках, а также указанием на то, чей ход следующий. Так (7,MAX)-это исходная конфигурация. В ней MAX имеет три различных хода, приводящих к конфигурации (6,1 MIN), (5,2 MIN) или(4,3 MIN). Полный граф этой игры, полученный применением ко всем состояниям среды всех возможных действий агентов, показан на рис.7.1.

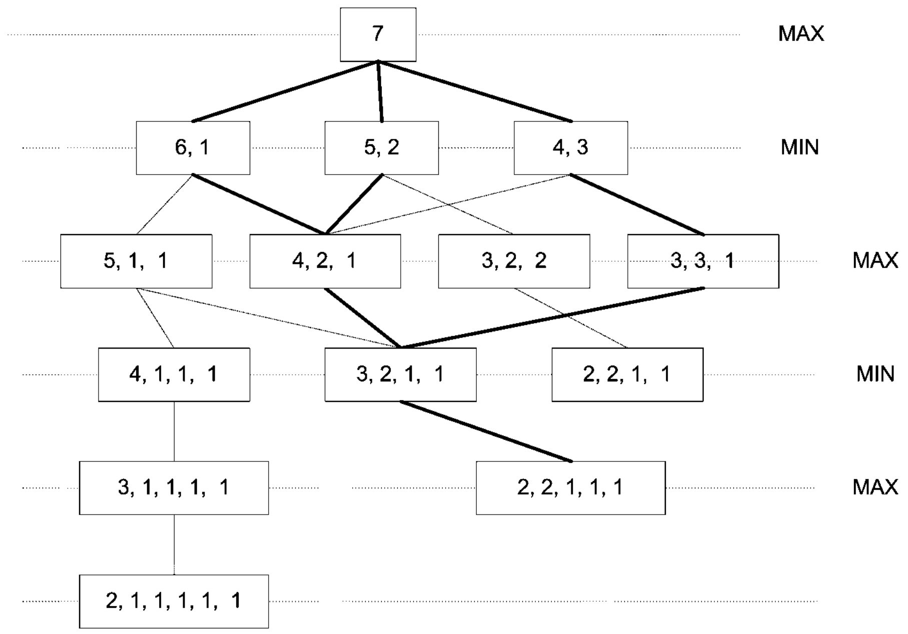

Рис. 7.1 Полный граф игры

Все концевые вершины соответствуют позициям, проигрышным для игрока, делающего следующий ход. С помощью этого игрового графа можно показать, что независимо от поведения игрока MAX игрок MIN всегда может выиграть. На рис.7.1 выигрышная стратегия показана жирными линиями.

Из рисунка видно, что для игровых деревьев (графов) не обязательно указывать чей следующий ход. Каждый уровень соответствует ходу одного из игроков. В нашем примере четные уровни это ходы игрока MAX, а нечетные уровни - ходы игрока MIN (самая верхняя, исходная вершина лежит на нулевом, т.е. четном уровне).

7.2. Минимаксная процедура

Поиск на графе выигрышной стратегии в сложных играх, таких как шахматы или шашки, для целиком взятой партии является совершенно нереальной задачей. Из-за большой комбинаторной сложности этих игр примитивный алгоритм поиска, который останавливается только в заключительных (окончательных) позициях игры, становится полностью непригодным.

Известны оценки, утверждающие, что игровое дерево партии в шашки содержит 10^{40} вершин, а в шахматы – 10^{120}. Эти оценки основаны на предположении, что из любой шахматной позиции может быть сделано приблизительно 30 допустимых ходов, а заключительные позиции возникают на глубине 40 ходов. Каждый ход состоит из двух полуходов (по одному полуходу на каждого участника игры).

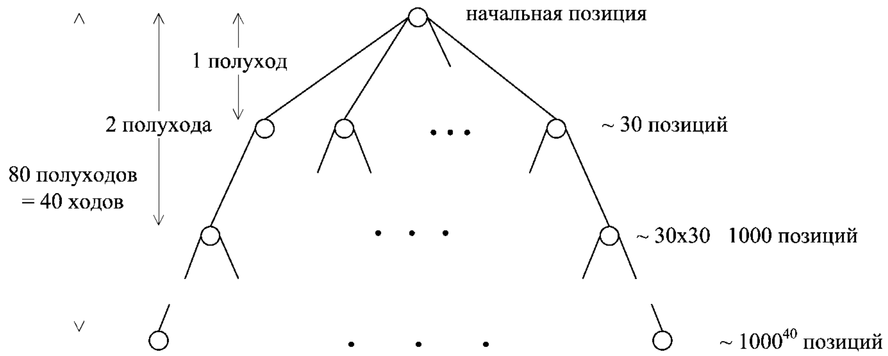

Рис. 7.2 Оценка дерева игры в шахматы

Естественно, в различных местах дерева, показанного на рис.7.2, встречаются одинаковые позиции. Тем не менее, доказано, что количество различных позиций на шахматной доске намного превосходит возможности любых компьютеров, которые могут быть созданы в обозримом будущем. Для порождения полного дерева игры потребуется время, измеряемое столетиями. Применение методов эвристического поиска, использующих оценочные функции вершин не спасает ситуацию – коэффициент ветвления не снижается до приемлемого уровня.

Однако, цель поиска на игровом дереве может быть просто в отыскании хорошего первого хода. После этого можно сделать найденный ход, подождать ответного хода противника и снова начать поиск хорошего первого хода. Можно применить или поиск в ширину, или поиск в глубину, или эвристические методы. Надо лишь ввести несколько искусственных условий остановки, основанных на таких факторах, как ограничение на время и объем памяти или наибольшая допустимая глубина вершин в дереве поиска.

По окончании поиска нужно выделить на графе поиска претендента на «наилучший» первый ход. Претендента можно найти, применив к концевым вершинам графа поиска оценочную функцию. Эта функция измеряет «ценность» позиции, представленной концевой вершиной. Выбор оценочной функции основан на учете свойств игровых позиций. Например, для шашек это может быть:

- относительный перевес в фигурах;
- контроль над центром;
- контроль дамок над центром.

При анализе игровых деревьев обычно используют соглашение, по которому позиции, выгодные для игрока MAX, оцениваются *положительными* значениями, а позиции выгодные MIN – *отрицательными.* Значения оценочной функции, близкие к 0, соответствуют тем «ничейным» позициям, которые не особенно выгодны ни MAX, ни MIN. Хороший первый ход может быть найден с помощью процедуры, называемой МИНИМАКСНОЙ.

Будем считать, что если бы игроку MAX был бы представлен выбор концевых вершин, то он бы выбрал ту, на которой значение оценочной функции – наибольшее. Следовательно, родителю концевых вершин MIN-вершин (который сам является MAX-вершиной) присваивается *возвращенное* значение, равное максимальному значению оценочной функции на концевых вершинах.

С другой стороны, если игроку MIN надо выбирать среди концевых вершин, то он скорее всего выберет ту, на которой значение оценочной функции наименьшее (т.е. наибольшее по модулю отрицательное число). Следовательно, родителю концевых MAX-вершин (который сам является MIN-вершиной) присваивается возвращенное значение, равное минимальному значению оценочной функции на концевых вершинах. После присвоения возвращенных значений всех концевых вершин строятся возвращенные значения на следующем уровне, причем считается, что игрок MAX предпочитает выбирать вершины с наибольшим возвращенными значениями, а игрок MIN – с наименьшими.

Так, уровень за уровнем значения «возвращаются» до тех пор, пока возвращенные значения не будут присвоены непосредственным потомкам корневой вершины. Если эта вершина соответствует ходу игрока MAX, то он должен выбирать в качестве первого хода тот, который соответствует дочерней вершине с наибольшим возвращенным значением. Пример минимаксного дерева приведен на рис. 7.3.

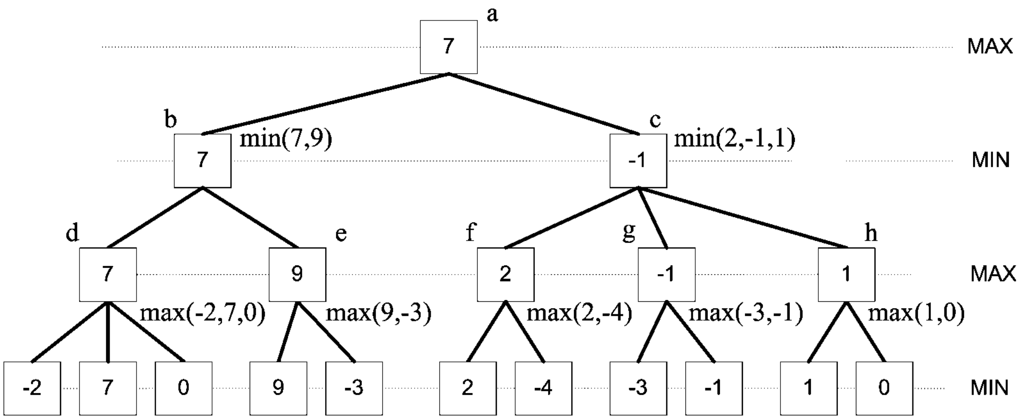

Рис. 7.3 Минимаксное дерево

На этом рисунке уровни позиций, в которых должен ходить игрок MAX, чередуются с позициями, в которых право сделать ход передается игроку MIN. Значения позиций нижнего уровня определяются с помощью функции оценки. Стоимость внутренних вершин можно вычислить, поднимаясь снизу вверх, от одного уровня к другому до тех пор, пока не будет достигнута корневая вершина.

Тогда возвращенная оценка (стоимость) вершины ***d*** будет определяться максимальной оценкой дочерних вершин - max(-2, 7, 0)=7, т.к. право выбора хода из позиции ***d*** передано игроку MAX. При этом возвращенная оценка (стоимость) вершины ***b*** будет определяться как минимальной оценкой дочерних вершин - min(7,9)=7, т.к. право выбора хода из позиции ***b*** находится у игрока MIN.

На рис. 7.3 результирующая возвращенная оценка корневого узла равна 7, поэтому наилучшим ходом для игрока MAX в позиции ***a*** является ***a-b***, наилучшим ответом для игрока MIN является ***b-d***, и т.д. Такая последовательность позиций в игре называется основным вариантом. Основной вариант определяет для обоих участников игру, оптимальную в соответствии с принципом минимакса. Оценка позиций на основном варианте не изменяется.

Полезность этой процедуры, в целом основана на предположении, что возвращенные значения для потомков корневой вершины более надежны в качестве оценок окончательной относительной ценности соответствующих им позиций, нежели значения, которые можно было бы получить непосредственным применением к этим позициям оценочной функции. В конечном итоге, возвращенные значения основаны на «просмотре вперед» по игровому дереву и поэтому зависят от свойств, проявляющихся ближе к окончанию игры.

7.3. Пример использования минимаксной процедуры

Рассмотрим простой пример игры в «крестики-нолики» на поле 3x3. Известно, что MAX ставит «крестик» (x), MIN ставит нолик (o), первый ход делает MAX. Проведем «поиск в ширину», пока не получим все вершины второго уровня, а затем к позициям, соответствующим этим вершинам, применим оценочную функцию.

Пусть для позиции b_i оценочная функция $f(b_i)$ задается следующими условиями:

1. Если в позиции b_i не выигрывает ни один из игроков, то $f(b_i)$ = (число строк + число столбцов + число диагоналей, на данный момент целиком свободных для игрока MAX) – (число строк + число столбцов + число диагоналей, на данный момент целиком свободных для игрока MIN).
2. Если в позиции b_i выигрыш получает игрок MAX, то $f(b_i) = \infty$ (или большое положительное число).
3. Если в позиции b_i выигрыш получает игрок MIN, то $f(b_i) = -\infty$ (или большое отрицательное число).

Пример: позиция b_i имеет вид (рис.7.4), тогда для нее значение оценочной функции $f(b_i)$ = 6 – 4 = 2.

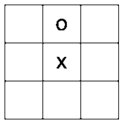

Рис. 7.4

При порождении дочерних вершин (позиций) используется свойство симметрии игрового поля относительно главных диагоналей, средней строки и среднего столбца. Поэтому следующие позиции (рис. 7.5) будут считаться идентичными:

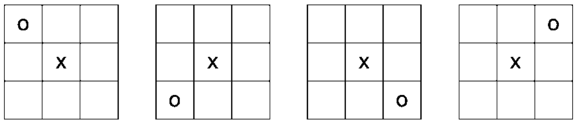

рис. 7.5

Симметрия позволяет уменьшить коэффициент ветвления игрового дерева на начальной стадии игры, а на более поздних стадиях игры он остается малым вследствие уменьшения числа свободных клеток на доске.

На рис. 7.6 показано дерево, полученное в результате поиска на глубину 2. Под концевыми вершинами указаны значения оценочной функции, а рядом с вершинами второго уровня показаны возвращенные значения.

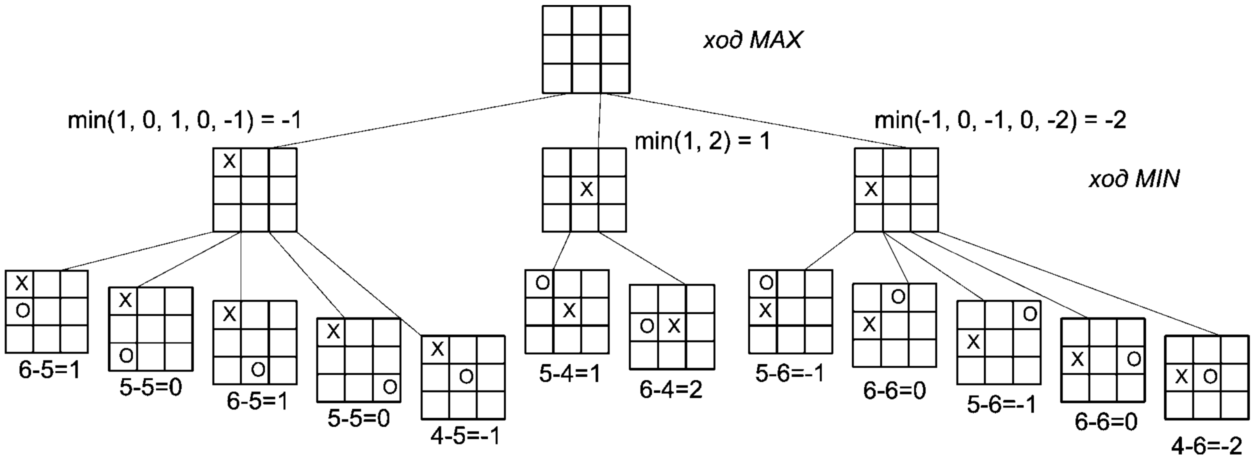

рис. 7.6 Дерево игры на глубину в два полухода

Поскольку наибольшим возвращенным значением обладает позиция с оценкой $f(b_i)$ =1, то она и выбирается в качестве первого хода (это наилучший первый ход игрока MAX). Предположим, что MAX сделал этот ход, а MIN в ответ поставил «нолик» в одну из свободных клеток (это плохой ход для MIN, но он может придерживаться любой стратегии поиска). Затем из новой позиции игрок MAX осуществляет поиск на два уровня вниз. Получается новое дерево поиска (рис.7.7), MAX выбирает наилучший ход из двух возможных и т.д. до окончания игры.

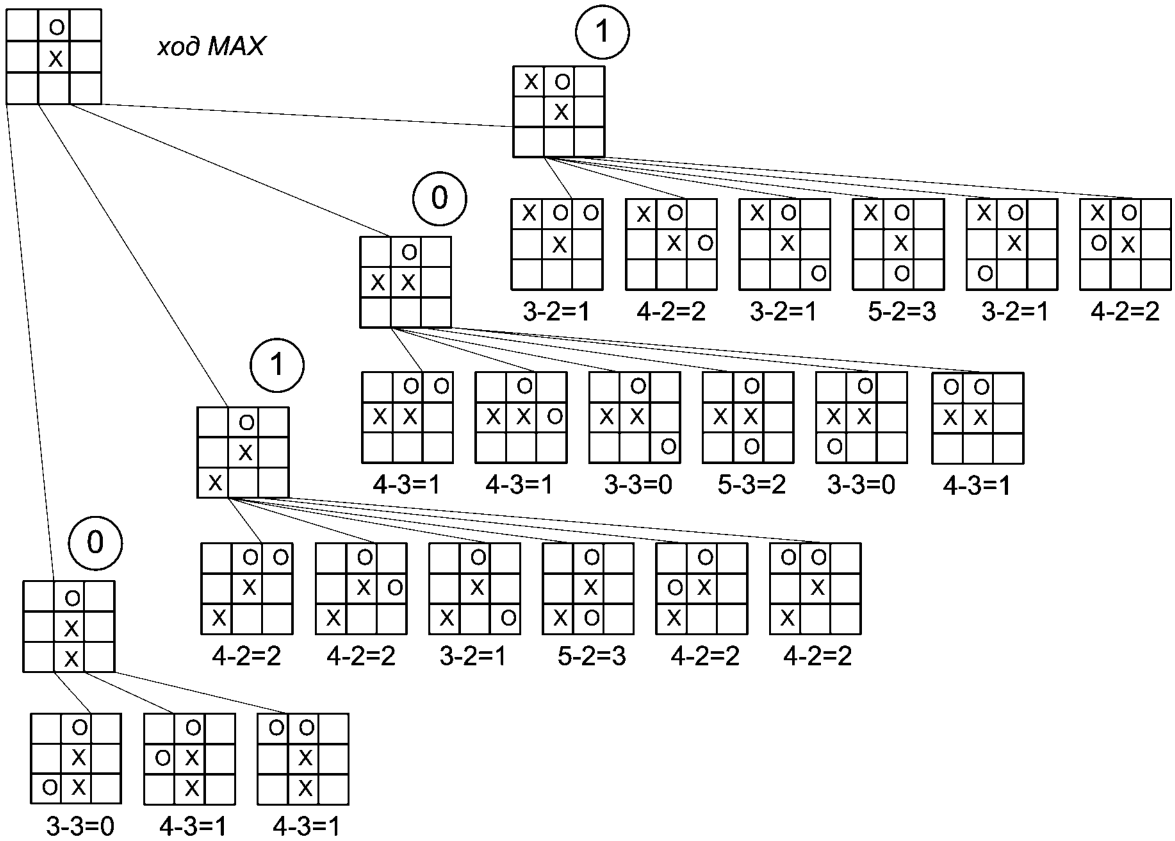

Рис. 7.7 Дерево игры на глубину в два полухода (второй этап).

7.4. Альфа-бета отсечение.

В минимаксной процедуре поиска на игровом дереве с оценочной функцией процесс порождения дерева поиска и процесс оценки позиции полностью отделены друг от друга. Оценка позиции может начаться только после завершения порождения дерева поиска.

В некоторых случаях такое разделение приводит к реализации крайне неэффективной стратегии. Если же оценивать концевые вершины и вычислять возвращенные значения по мере порождения дерева, то для нахождения столь же хорошего хода длительность требующегося на это поиска можно существенно сократить.

Основная идея метода состоит в сравнении наилучших оценок, полученных для полностью изученных ветвей, с наилучшими предполагаемыми оценками для оставшихся ветвей. Можно показать, что при определенных условиях некоторые вычисления являются лишними. В отличие от метода минимакса, который заключается в построении пространства ходов путем их прямого перебора, в методе отсечений значительная часть ходов подвергается неявному перебору, проводимому с помощью процедуры отбрасывания частей дерева.

Пусть вершина *S* соответствует позиции (рис.7.8), в которой ход принадлежит игроку MAX. При этом в распоряжении игрока MAX несколько возможных ходов, два из которых показаны на рисунке. В результате одного из них будет получена позиция *A*, в результате другого – позиция *Y*. Пусть позиция *A* уже полностью проанализирована и найдено ее значение оценки - α. Перейдем к анализу позиции *Y*. Допустим, что один ход из этой позиции приведет к позиции *Z*, оценка которой по методу минимакса равна *z*. Пусть $z<\alpha$, а *y* – оценка вершины *Y*, тогда в любом случае выполняется неравенство $y<z$, так как в позиции *Y* ход за MIN, который из всех возможных ходов выберет такой, который приведет к позиции с минимальной оценкой.

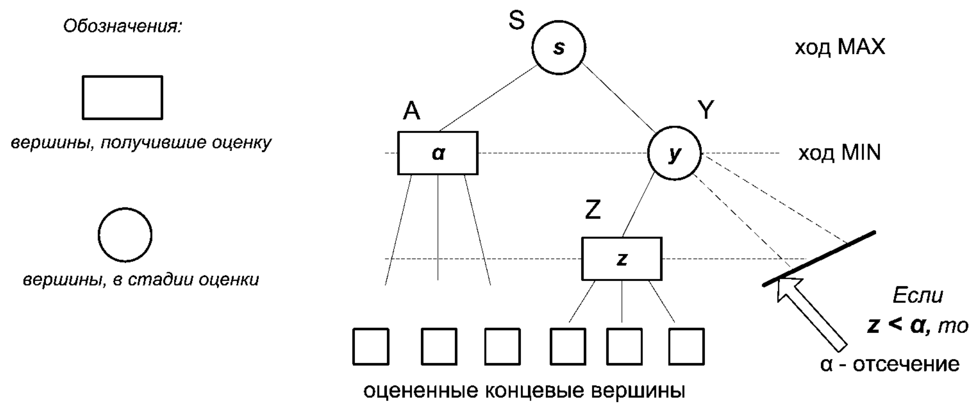

рис. 7.8 Альфа-отсечение

Отсюда следует, что $y<z<\alpha$. Это неравенство показывает, что неизвестное значение оценки y для вершины *Y* не будет оказывать влияния на последующие результаты, и анализ всех последующих ветвей, выходящих из узла *Y*, кроме *Z*, оказывается бесполезным. Таким образом, оценка позиции *Y* будет заведомо хуже, чем оценка позиции *A*. Другими словами, после анализа узла Z остальные ветви дерева, выходящие из узла Y, могут быть отброшены. Эта процедура называется *альфа-отсечением*.

Эти рассуждения легко распространить на случай, в котором на первом уровне ход принадлежит игроку MIN. При этом оценка вершины Z на глубине 2 должна удовлетворять условию $z>\beta$. В этом случае процедура называется *бета-отсечением* (рис.7.9).

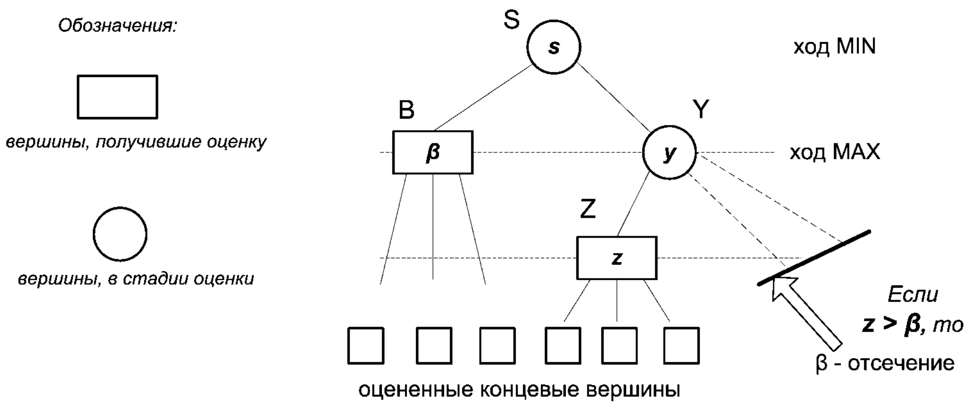

Рис. 7.9 Бета-отсечение

Применив эту процедуру к ветвям анализируемого дерева, находящимся на большей глубине, можно показать, что они также могут быть отброшены, при условии, что вершины дерева принадлежат уровням хода одного игрока.

Пусть игрок MAX должен сделать ход из позиции S (рис. 7.10) и имеет две возможности выбора позиций A или U. Положим, что полный анализ A завершен и получена возвращенная оценка α.

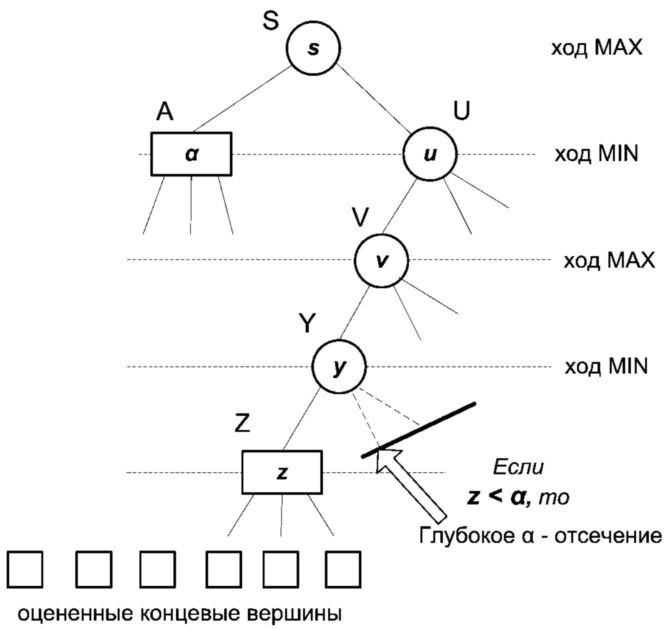

Рис. 7.10.Глубокое альфа-отсечение

Анализируя позицию U, можно заметить, что после нечетного числа промежуточных полуходов, в данном случае трех, возникает позиция Z с оценкой z. Снова допустим, что $z<\alpha$. Для вершины Y, расположенной на более высоком уровне, непосредственно следующим за Z, найдем оценку y, которая удовлетворяет условию $y\leq z$, так как ход из позиции Y принадлежит игроку MIN. И в данном случае результат анализа остальных позиций, вытекающих из Y, не может изменить конечного результата оценки s. Покажем это. Для оценки позиции V справедливо неравенство $v\geq y$, так как право хода из этой позиции принадлежит игроку MAX. Если $v>y$, то результаты анализа других ходов не могут привести к изменению значения оценки v, так как $y\leq z$, и величина y может быть только уменьшена. Если $v=y$, тогда справедливо неравенство $v=y\leq z<\alpha$, то есть вершина V получает оценку меньшую, чем вершина A. Эта ситуация уже рассмотрена выше.

Следовательно, в обоих случаях оценка s вершины S не зависит от других ветвей дерева, выходящих из узла Y и анализ последних можно не производить. Такая ситуация называется глубоким альфа-отсечением. Рассуждая аналогично, можно рекуррентно спуститься до вершины Z, лежащей на уровне произвольной глубины, принадлежащем тому же игроку, что и уровень S.

Процедура отсечения может быть применена только начиная с того момента, когда получены оценки по крайней мере двух конечных узлов.

На рис. 7.11 показан пример игрового дерева к которому применена минимаксная процедура возвращения оценок позиций путем восхождения по ветвям. На рис 7.12 показано применение процедуры альфа-бета отсечения к тому же игровому дереву.

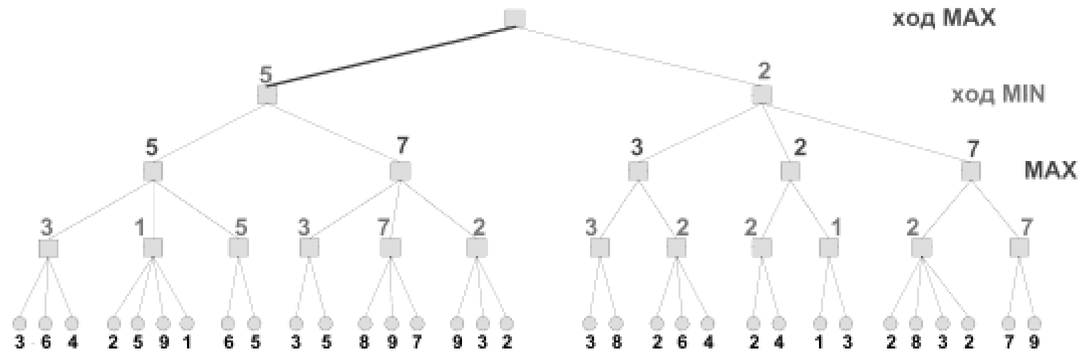

рис. 7.11 Пример применения минимаксной процедуры.

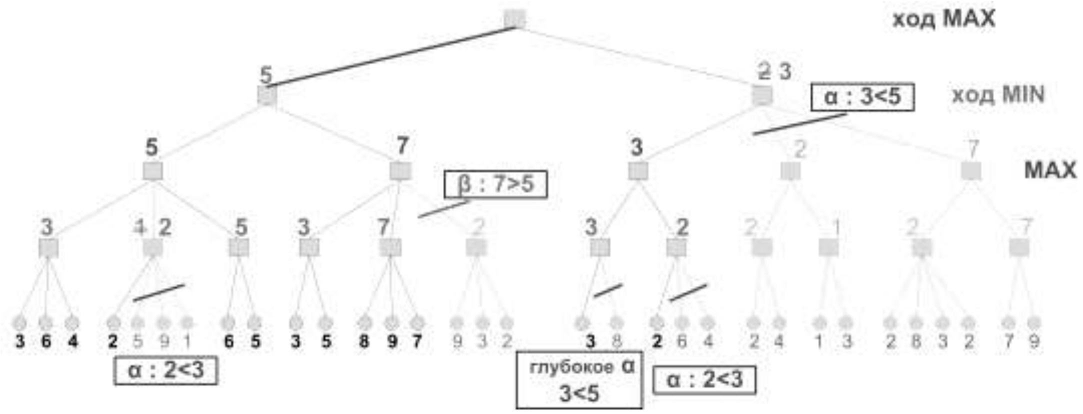

Рис. 7.12 Пример применения процедуры альфа-бета отсечений.

Процедура отсечений дает тот же результат, что и метод минимакса, но выполняется быстрее. С ее помощью можно получить хорошие результаты при удачно составленной оценочной функции и достаточно большой глубине анализа, что требует большого объема вычислений. Процедура отсечений является тем более производительной, чем более упорядоченно расположены вершины при отыскании каждого хода. В идеальном случае, недостижимом на практике, вершины на каждом уровне должны располагаться по монотонно убывающим значениям оценки.

Рассмотренные процедуры систематического перебора допустимых ходов обладают рядом существенных недостатков.

Первый недостаток заключается в полном отсутствии игровой стратегии. Каждая новая позиция рассматривается в отрыве от всех остальных. Процедуры указанного типа не ведут игру, а анализируют последовательность позиций, полностью независимых друг от друга. Минимакс не позволяет действовать в соответствии с заранее выбранным планом. Все принимаемые решения определяются оценочной функцией.

Второй недостаток состоит в так называемом эффекте горизонта. Процедура не может оценить последствия хода, которые проявляются на глубине, превышающей предельную глубину анализа дерева.