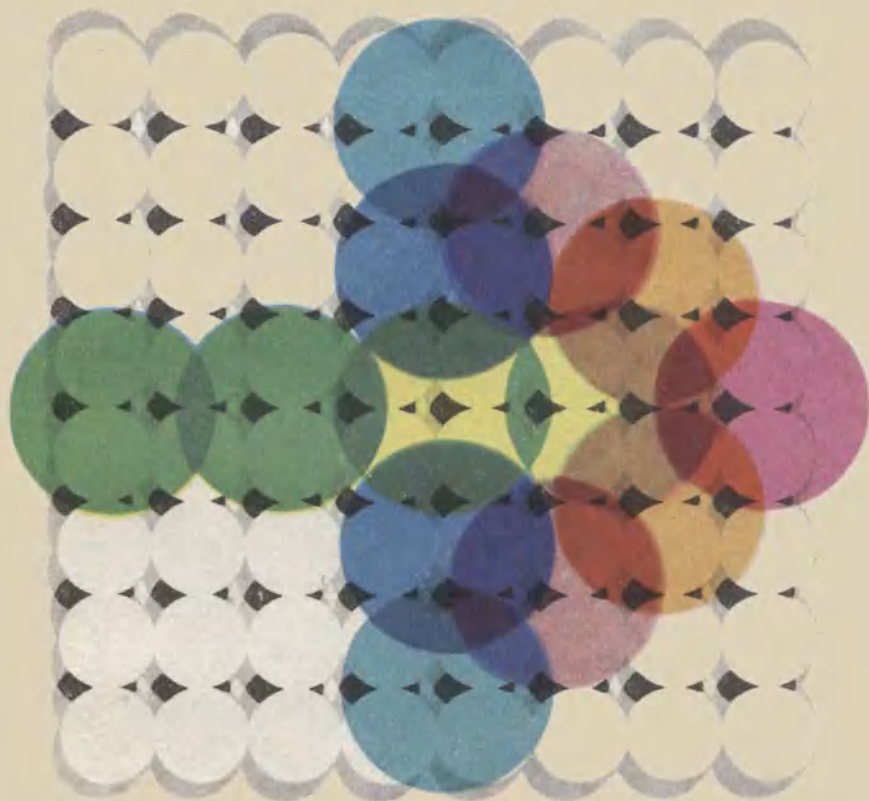


КИБЕРНЕТИКА



М. М. БОТВИННИК

# **О РЕШЕНИИ НЕТОЧНЫХ ПЕРЕБОРНЫХ ЗАДАЧ**

Настоящая серия печатается по рекомендации IX Международного Совещания руководителей научно-технических издательств социалистических стран (июнь 1975 г.). В серии участвуют:

**Издательство «Советское радио» (СССР),  
Издательство технической литературы  
(ВНР)**

**Издательство «Техника» (ГДР)  
Издательство научно-технической литературы (ЧССР)**

КИБЕРНЕТИКА

М. М. БОТВИННИК

# О РЕШЕНИИ НЕТОЧНЫХ ПЕРЕБОРНЫХ ЗАДАЧ



МОСКВА  
«СОВЕТСКОЕ РАДИО» 1979

ББК 32.81

Б86

УДК 007+518.92

**Ботвинник М. М.** О решении неточных переборных задач. — М.: Сов. радио, 1979, 152 с. (Кибернетика).

В книге приводится алгоритм игры шахматного мастера, на основе которого выполнена программа «Пионер». Эта программа частично проверена в эксперименте на шахматных этюдах, и анализ шахматной позиции, полученный «Пионером», оказался весьма близок к анализу шахматного мастера. Отметим, что этюд Г. Надареишвили, решенный «Пионером», не смог решить чемпион мира 1977 г. среди компьютеров — американская программа «Чесс 4.6».

Новая работа экс-чемпиона мира по шахматам **М. М. Ботвинника** рассчитана на широкий круг читателей, интересующихся проблемами управления, планирования, шахматами и др.

48 рис., 1 табл., библи. 11 назв.

**Редакция кибернетической литературы**

Б  $\frac{30501-013}{046(01)-79}$  БЗ-97-8-78 1502000000

## ПРЕДИСЛОВИЕ

В книге излагается теория решения неточных переборных задач, которая является в какой-то степени гипотезой, ибо практика еще не подтверждает полностью теоретические положения. Когда искусственный шахматист — программа «Пионер» — начнет играть в силу мастера, эта теория получит достаточно прочную основу.

Незаконченная история создания сильного шахматного автомата связана с борьбой двух направлений. Долгое время почти безраздельно господствовало мнение, что компьютер не должен копировать методы мышления шахматного мастера и что основной метод машинной игры должен базироваться на полном переборе ходов. После первых успехов «Пионера» положение несколько изменилось; можно предположить, что в будущем компьютер все больше и больше будет подражать человеку.

В книге сначала приводятся общие положения, которые, по мнению автора, надлежит использовать при решении неточных переборных задач управления, а затем на примере шахматной игры показывается, как эта общая теория привела к успешным результатам.

Приложения к книге написаны математиками, принимающими участие в создании программы «Пионер». Надо полагать, что эти работы заинтересуют программистов и помогут им применить приведенные методы к решению других задач, имеющих практическое значение.

## 1. ОБЩИЕ ПОЛОЖЕНИЯ

### ОПРЕДЕЛЕНИЕ НЕТОЧНОЙ ЗАДАЧИ

Понятие неточной задачи было введено автором несколько лет назад [1], но тогда не было дано ее строгого определения.

Неточной будем называть такую переборную задачу, которая решается с помощью минимаксной процедуры на усеченном дереве перебора (см. также словарь терминов на с. 141).

Процедура минимакса, дерево перебора — понятия, хорошо известные. Поясним, что здесь подразумевается под понятием усеченного дерева перебора.

Задачи, решаемые с помощью формирования дерева перебора возможностей (элементарных действий), могут быть различной трудности, могут быть связаны с построением как малого, так и большого (иногда бесконечно большого) дерева перебора. Если ресурсы устройства по переработке информации (память и быстродействие) таковы, что невозможно сформировать и изучить дерево перебора полностью, то либо надо отказаться от решения задачи, либо соглашаться на приближенное, неточное ее решение. В этом случае (если неточное решение приемлемо) приходится ограничить глубину вариантов; если мы идем на усеченное (по глубине) дерево перебора и на неточность решения, то, по сути дела, переборная задача и становится неточной.

Определение неточной задачи, таким образом, неразрывно связано с общим методом ее решения и ресурсами устройства, перерабатывающего информацию.

Если удастся провести минимаксную процедуру на полном дереве перебора, задача остается точной. Вообще, задачи могут решаться точно и другими методами, например с помощью уравнений или по какому-либо точному алгоритму. Отметим, впрочем, что задача может быть неточной и в других случаях, а не только в том, который соответствует принятому определению, — решение может быть неточным и при применении уравнений...

Примем, однако, то узкое определение неточной задачи, которое было дано: неточные задачи, не соответствующие этому определению, рассматриваться не будут.

## НЕТОЧНЫЕ ЗАДАЧИ И СИСТЕМЫ УПРАВЛЕНИЯ

Система управления выполняет три функции: получение информации, ее переработку и исполнение решения. При переработке информации система управления может решать задачи различной сложности и различной трудности — все зависит от назначения системы управления. Когда задача решается системой управления по методу формирования усеченного дерева перебора, возникает неразрывная связь между теорией решения неточных задач и теорией управления.

Таким образом, кажущаяся на первый взгляд отвлеченной проблема решения неточных задач имеет большое практическое значение, поскольку большинство задач управления являются задачами неточными.

### ДВА МЕТОДА РЕШЕНИЯ НЕТОЧНЫХ ЗАДАЧ

На возможность решения неточных задач двумя принципиально различными методами указал еще в 1949 г. К. Шеннон [2], когда он поставил вопрос о программировании шахматной игры; игра в шахматы — одна из типичных неточных задач.

По первому методу в усеченное дерево перебора включаются все возможности (ходы) без каких-либо исключений. По второму — возможности, заведомо лишённые смысла (см. с. 142), в усеченное дерево перебора не включаются; дерево состоит лишь из тех возможностей, которые могут иметь смысл.

Первый метод и способ его реализации в принципе просты; Шеннон дал определенные рекомендации по применению этого метода, которые и были приняты математиками на вооружение. При этом Шеннон отметил, что второй метод является более надеживающим, но никаких путей для его реализации не указал.

Между тем, первый метод, видимо, безнадежен, если стремиться к хорошему решению неточной задачи. Покажем это на примере игры в шахматы.

В среднем в шахматной позиции каждая сторона может сделать примерно 20 ходов. Пусть в исходной позиции ход белых; все 20 ходов белых фигур должны быть включены в перебор. Теперь ход черных: на каждый ход белых черные могут ответить 20 способами, а всего будет 400 ответных ходов черных. Если продлить варианты еще на один ход белых, то общее количество ходов, включенных в дерево перебора, составит 8420 ходов. Если удвоить длину вариантов, иначе говоря, рассматривать варианты на три полных хода (шесть полуходов), то общее количество ходов возрастет примерно до 67 000 000!

Если глубина дерева растет линейно, то ширина — как показательная функция. Поэтому дерево и разрастается катастрофи-

чески (рис. 1). Как правило, дерево изображается формирующимся сверху вниз.

Мы должны найти в таком дереве оптимальный вариант; трудно понять, что он может состоять всего из шести ходов. Таким образом, излишних ходов оказалось более чем достаточно. Это задача аналогична той, когда надо перебрать мусорную кучу, чтобы найти жемчужное зерно! Однако, перебирая кучу, мы знаем, к чему стремимся; при анализе же усеченного дерева это бывает далеко не всегда.

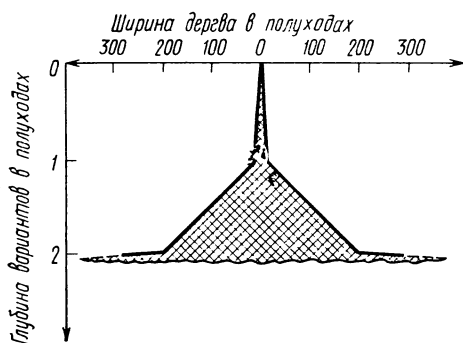


Рис. 1. Разрастание дерева перебора в шахматной игре при полном переборе

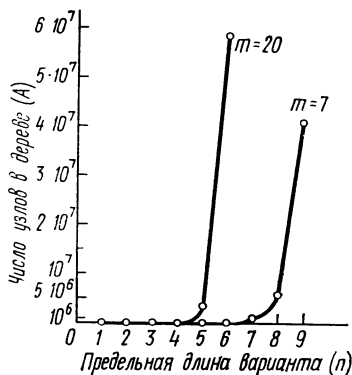


Рис. 2. Зависимость числа узлов в дереве перебора от предельной длины варианта

Если мы создаем по подобному алгоритму шахматную программу для ЭВМ, то ресурсы машины (память и быстродействие) не имеют такого большого значения, как это может показаться на первый взгляд. Как только мы удлиняем глубину вариантов на один полный ход, ресурсы ЭВМ должны возрастать в десятки раз — практически невыполнимое требование. И первостепенное значение приобретает не мощность ЭВМ, а искусство программиста, насколько он сумеет, опираясь на основной принцип программы — полный перебор ходов, отойти от этого принципа, чтобы сэкономить ресурсы машины. Видимо, этот факт и объясняет победу советской программы «Каисса» на первом шахматном чемпионате мира в Стокгольме в 1974 г., хотя использовалась машина, сравнительно скромная по ресурсам; были программы, которые играли с помощью машин, в несколько раз более мощных, но они спасовали перед «Каиссой». При таком методе решения неточной задачи увеличение мощности ЭВМ мало может помочь делу. За четверть века удалось повысить шах-



матную квалификацию ЭВМ примерно до уровня второго разряда; за последние лет пять дальнейшего продвижения не наблюдается.

Две проблемы могут возникнуть перед программистом при совершенствовании программы: усиление (нахождение более сильного) оптимального варианта и ускорение решения задачи. Усиление необходимо, если получаемое оптимальное решение оказывается не на высоте — в этом случае напрашивается увеличение предельной длины вариантов, т.е. увеличение объема дерева перебора. Покажем более подробно (на примере шахмат), что этот путь имеет весьма ограниченные возможности — если программа основывается на полном переборе.

Примем, что в шахматах, в среднем, в позиции можно сделать  $m$  ходов. Тогда при предельной длине варианта, равной  $n$  полуходам, общее количество узлов в дереве перебора (если включать в перебор все ходы), в среднем, будет равно

$$A_n = m + m^2 + m^3 + \dots + m^n.$$

Хорошо известный метод «граней и оценок» (за рубежом этот метод называют « $\alpha - \beta$ -отсечением») в пределе может уменьшить это число до

$$A'_n = \sqrt[n]{m + m^2 + m^3 + \dots + m^n}.$$

Примем  $m=20$ , что вполне правдоподобно, а  $n=6$ . Тогда

$$A'_n = \sqrt[6]{20 + 400 + 8000 + 160\,000 + 3\,200\,000 + 64\,000\,000} \approx 8000.$$

Практически метод ветвей и границ никогда не приводит к такому предельному снижению числа узлов. Так, программа «Чесс 4.6» при предельной длине вариантов, равной шести полуходам (есть в дереве и так называемые форсированные варианты, которые продлеваются при взятиях или шахах, но вряд ли это может оказать существенное влияние на число узлов), имеет в дереве перебора примерно 400 000 узлов, т. е. различие получается существенным. Примем поэтому  $A'_n$  равным не  $A_n^{1/2}$ , а  $A_n^{2/3}$ . Тогда вместо 8000 получим 160 000, что значительно ближе к истине.

Построим зависимость числа узлов в дереве перебора от предельной длины варианта (без учета  $\alpha - \beta$ -отсечения) ( $m=20$  и  $m=7$ ). Из рассмотрения графика рис. 2 видно, что при увеличении числа узлов предельная длина варианта практически увеличивается лишь до значения  $n=5$  и  $n=8$ , а затем эта зависимость становится вялой. Поскольку число узлов можно условно считать

пропорциональным быстродействию ЭВМ, то ясно, что увеличение быстродействия, начиная с некоторой длины варианта, не может способствовать усилению игры.

При снижении среднего числа возможностей в каждом узле до  $m=7$  ситуация улучшается: увеличение  $A'_n$  (увеличение быстродействия) оказывает большее влияние на предельную длину вариантов  $n$ . Этим и воспользовались авторы программы «Чесс 4.6» — при переходе в эндшпиль, когда количество фигур сокращалось и величина  $m$  снижалась, они, используя быстродействие компьютера «Сайбр-176» (12 000 000 оп./с), увеличивали предельную длину варианта с 6 до 12 и получили более сильные оптимальные варианты.

Если же сильная игра уже достигнута и требуется лишь ускорить поиск решения, то увеличение быстродействия компьютера полностью решает проблему.

**Перспективность второго метода** основывается прежде всего на том, что так решает неточные задачи человек, в частности сильный шахматист. При исключении из формирующегося дерева возможностей, заведомо не имеющих смысла, человек в поисках оптимального варианта анализирует узкое и глубокое дерево. В этом случае количество ходов в оптимальном варианте уже соизмеримо с тем количеством ходов, которое содержится в сформированном дереве.

Некоторые математики, не отрицая, что так мыслит человек, все же считают, что ЭВМ должна действовать иначе; они полагают, что человек мыслит неоптимальным для ЭВМ образом и нечего переносить человеческие методы мышления на машинные программы. Если в этих рассуждениях есть свой резон, то позволительно спросить: почему же за четверть века попытки обучить ЭВМ играть в шахматы нечеловеческими путями дали слабые результаты? Утверждать, что ЭВМ должна действовать в направлениях, не свойственных человеку, можно было бы, если шахматные программы, основанные на включении всех ходов в усеченное дерево перебора, превзошли бы шахматных мастеров — пока этого не наблюдается.

В подтверждение того, что ЭВМ должна действовать по-своему, некоторые специалисты приводят пример с аппаратами тяжелее воздуха — самолет и птица летают по-разному. Но так ли это? Действительно, с аппаратной точки зрения самолет и птица действуют по-разному — самолет использует тяговую силу двигателей, а птица машет крыльями. Но и самолет, и птица преодолевают силу тяжести и сопротивление атмосферы; по аналогии можно предположить, что и человек, и ЭВМ, решая неточные задачи, должны одолеть проблему формирования узкого и глубокого де-

рева. Человек это давно умеет делать, теперь надо проверенный тысячелетиями опыт передать ЭВМ.

Если это будет сделано и шахматные программы будут формировать узкое и глубокое усеченное дерево перебора, тогда, вероятно, и скажутся способности ЭВМ — машина, имеющая большие ресурсы, превзойдет своих конкурентов.

В заключение уместно поставить вопрос: какому из двух методов и в каких случаях следует отдать предпочтение? Пожалуй, в общем виде теперь уже можно дать соответствующие рекомендации.

Из рассмотрения рис. 2 следует, что когда число ходов в узле (в позиции) падает, то кривая  $y=f(x)$  ( $y$  — число узлов в усеченном дереве перебора, а  $x$  — предельная длина варианта) сдвигается вправо и метод полного перебора дает более глубокое решение. При увеличении числа ходов в узле указанная кривая сдвигается влево, длина вариантов уменьшается и решение становится более слабым.

Таким образом, в сложных задачах управления, где число возможностей, подлежащих рассмотрению, велико, ценность метода полного перебора сомнительна, а в сравнительно простых задачах, видимо, этот метод может быть рекомендован, поскольку при втором методе программа значительно сложнее. Второй же (человеческий) метод особенно должен иметь ценность в сложных задачах, а в простых — вряд ли может выдержать конкуренцию с методом полного перебора.

### ЦЕЛЬ ИГРЫ И ОЦЕНОЧНАЯ ФУНКЦИЯ

Цель игры является основой алгоритма игры (решения неточной задачи). Так, известно, что цель точной игры в шахматы — мат. При усеченном дереве перебора точная цель (мат) практически не играет роли, поскольку ни один вариант до мата не доходит. Должна появиться новая цель — неточная\*, которая соответствует неточной игре, неточной задаче (рис. 3). Напомним, что при усеченном дереве перебора любая задача (даже если она конечна) превращается в неточную.

Как же используется при решении задачи цель игры, как цель игры влияет на формирование дерева перебора? Неточная цель игры позволяет логически оборвать вариант перебора; в том случае, когда вариант обрывается до предельной длины, на этом он и заканчивается. Вариант доходит до предельной длины лишь тогда, когда в соответствии с целью игры он должен был бы продолжаться далее.

---

\* Э. М. Вайсборд считает, что более удачен термин «промежуточная» цель.

Таким образом, цель игры проявляется в логическом обрыве варианта. Если цель достигнута или установлено, что ее заведомо нельзя достичь, вариант обрывается.

Во всех играющих программах варианты продолжают до предела (а если некоторые варианты переходят этот предел, это означает, что они продолжают до какого-нибудь другого ограничения), соответствующего усечению дерева перебора. Продолжение всех вариантов до предельной длины — верный признак

того, что в этих алгоритмах (без исключения!) *отсутствует цель неточной игры*. Этот поразительный факт должен быть особо отмечен; в неявной форме цель игры осталась — мат (лишь из-за мата вариант прекращается до предельной длины), но мат — это цель точной игры.

Да, К. Шеннон был последователен, когда в 1949 г. предложил обучать ЭВМ игре в шахматы с помощью усеченного дерева перебора. Это был важный шаг вперед, но, оставив точную цель игры, не установив неточной цели, не совершив этого второго необходимого шага, Шеннон — вследствие своего авторитета — обрек некоторых математиков на поистине «бесцельную» работу. Конечно, польза от этих математических изысканий была, но лишь единственная — удалось убедиться в том, что без

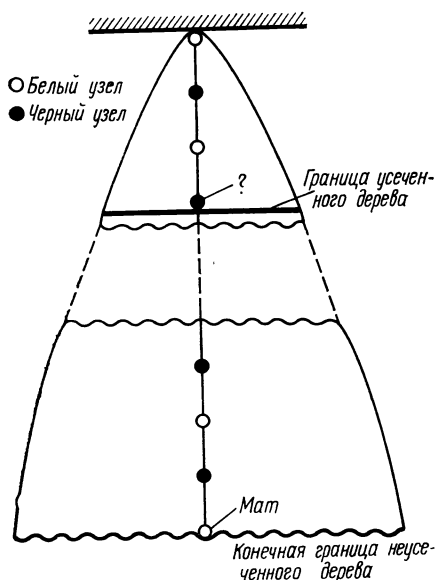


Рис. 3. Точная и неточная цели игры в шахматах

неточной цели нельзя решать неточные задачи!

А чему же служит оценочная функция?

Если неточной цели нет, от оценочной функции один толк — она оценивает вариант, когда он закончен. Если неточная цель есть, оценочная функция используется и для решения вопроса об обрыве варианта, и (как это будет ясно из дальнейшего) вообще для формирования узкого и глубокого дерева; при этом сохраняется и обычное, основное ее назначение — оценка завершенного варианта, используемая в процедуре минимакса.

Вновь отметим, что когда неточной цели нет, оценочная функция используется лишь для оценки усеченного варианта.

Цель игры отвечает на вопрос — к чему мы стремимся. Оценочная функция способна свидетельствовать о том, насколько успешно цель игры реализуется. В алгоритме Шеннона дана лишь оценочная функция, но нет цели игры. Поэтому по Шеннону можно оценить вариант и сравнить его с другим вариантом, но нельзя разумно его закончить из-за отсутствия цели.

Таким образом, цель игры и оценочная функция суть разные понятия, и для полноценного решения неточной задачи и то, и другое должно быть формализовано и использовано.

### ЦЕЛЬ И ПРОГНОЗ (ОПТИМАЛЬНЫЙ ВАРИАНТ)

Оптимальный вариант получается в результате минимаксной процедуры на усеченном дереве перебора. Следует рассматривать оптимальный вариант как прогноз того, насколько хорошо может быть реализована цель неточной игры.

Ни оптимальный вариант, ни его оценка не являются и не могут быть целью игры. Оптимальный вариант — всего лишь прогноз достижения цели на данный момент. Этот прогноз со временем по мере получения дополнительной информации может претерпевать изменения — все зависит от ситуации.

Изменения оптимального варианта могут быть связаны прежде всего с усечением дерева перебора. Часть вариантов имеет предельную длину, или, точнее говоря, они заканчиваются не логически, а принудительно. Оценки этих вариантов известны, но надежны ли они? С течением времени мы продвигаемся по оптимальному варианту, и если предельная длина вариантов не меняется, мы должны все время «досчитывать» оптимальный вариант. При этом он может и перестать быть оптимальным... В этом и состоит одна из неточностей задачи — неточность прогноза. Тип цели — понятие устойчивое; ее значение, а также прогноз — изменчивы.

Оценку оптимального варианта можно рассматривать как прогнозируемую обратную связь. Примем, что система управления действует по оптимальному варианту — настолько удачно он был высчитан. Тогда оценка реализованного оптимального варианта и станет реальным сигналом обратной связи. Когда же мы находимся еще в исходном положении, когда лишь был найден оптимальный вариант и система управления еще не действовала, то его оценка является по сути дела прогнозируемой обратной связью.

Еще раз отметим, что оценка оптимального варианта (прогноз) ни в коей мере не может быть эквивалентна цели игры. Эта оценка может быть и хорошей, и плохой, а плохая оценка не может рассматриваться как цель системы управления.

Система управления может быть совокупностью систем управления; совокупности систем могут образовывать новую совокупность систем. На рис. 4 изображена трехступенчатая система управления.

Каждая ступень системы управления должна иметь свою индивидуальную цель игры, но эти цели должны быть одного типа. Лишь в этом случае многоступенчатая система может удовлетворительно функционировать. Если у какой-либо ступени цель игры по типу будет отличаться от целей других ступеней, то данная ступень будет чужеродным элементом в многоступенчатой системе управления.

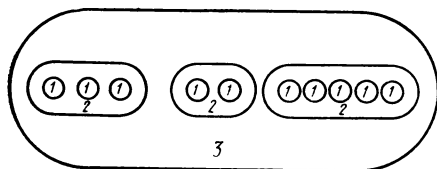


Рис. 4. Трехступенчатая система управления:

1, 2, 3 — соответственно первые, вторые и третья ступени системы

Понятие «ступень» многоступенчатой системы управления может и должно быть формализовано. Будем называть ступенью такую часть системы управления, которая имеет свою индивидуальную (но типовую) цель игры и в которой прекращается перебор при достижении (недостижении) этой

цели (т. е. когда выясняется возможность достижения цели).

Это определение имеет решающее значение при формировании ступеней и одновременно указывает на важнейшее значение цели игры в формировании многоступенчатой системы.

С точки зрения формирования узкого и глубокого дерева многоступенчатая система имеет большое преимущество перед одноступенчатой системой управления. Суть дела в том, что когда идет формирование дерева перебора и ситуация в какой-либо ступени выяснилась — цель достигнута или стала уже недостижимой, — то перебор в этой ступени прекращается (хотя он и продолжается в других ступенях). Это приводит к сокращению дерева перебора. Далее будут отмечены и другие достоинства многоступенчатых систем.

Многоступенчатые системы имеют самое широкое распространение. Завод, например, можно рассматривать как четырехступенчатую систему управления: рабочий — бригада — цех — завод. Если какая-нибудь бригада уже использовала за половину рабочего дня все заготовленные детали и ей больше делать нечего (цель достигнута или недостижима), то начальник цеха, управляя своим производством, будет искать оптимальный вариант, исключив эту ступень управления — бригаду из формируемого дерева перебора.

Многоступенчатые системы управления могут различаться, прежде всего, распределением устройств переработки информации по ступеням систем, или, иначе говоря, по местам принятия решений. Возможны следующие типы многоступенчатых систем.

А — управление многоступенчатой системой происходит из единого центра с одним устройством переработки информации. Типичная модель такой системы управления — игра в шахматы. Далее будет показано, что игру в шахматы можно рассматривать как трехступенчатую систему. Управление всеми тремя ступенями происходит в едином центре — мозге шахматиста.

В — каждая ступень имеет свое устройство переработки информации, свой центр управления. Приведенный ранее пример четырехступенчатой системы управления заводом иллюстрирует этот случай; он, по-видимому, более сложен с точки зрения управления.

Г — только низшие ступени имеют центры управления, а совокупности их центров управления не имеют. Этот тип противоположен типу А. Тип Г характерен, по-видимому, для растительного мира и неживой природы. Можно сказать, что в этом случае низшие ступени управления, по сути дела, образуют не совокупности ступеней, а их множества. У ступеней, образующих множество, присутствуют лишь «эгоистические» интересы; у множества ступеней общая цель отсутствует.

Разница между типами А и В не только принципиальная, но также аппаратная. Наличие множества центров управления (тип В) аналогично единому комплексу ВЦ или многопроцессорной ЭВМ. В тех случаях, когда задачу можно решать параллельно, множественность центров управления приводит как бы к увеличению памяти и быстродействия устройства переработки информации. Увеличение ресурсов может быть очень большим — это важное преимущество многоступенчатой системы типа В.

Если несколько устройств переработки информации (люди или ЭВМ) решают одинаковую (одну и ту же) задачу, то она может быть решена лишь с теми скоростью и глубиной, которые свойственны наиболее сильному устройству. Конечно, и другие устройства будут трудиться не напрасно — при решении задачи можно будет избавиться от случайных ошибок, но и только! Если эти несколько устройств переработки информации решают разные задачи, составляющие общую, то эта составная задача будет решена примерно так, как если ЭВМ, которая одна решала эту задачу, обладала бы быстродействием и памятью, равными суммарному быстродействию и суммарной памяти всех используемых при решении ЭВМ.

Таким образом, выгода от многоступенчатой системы управления состоит не только в том, что в ступенях может обрываться

перебор по выяснении в данной ступени вопроса о достижении цели, но и в том, что многоступенчатую систему может обслуживать множество ЭВМ и это способствует как ускорению, так и более глубокому решению задачи.

Когда многоступенчатая система имеет один центр управления — примером такой системы является шахматная игра — используется мощь лишь одного устройства переработки информации. В шахматах трудно создать систему с множеством центров управления, ибо ступени в шахматах (как это будет объяснено далее) неустойчивы, они непрерывно меняются. Но в других многоступенчатых системах (примером такой системы является завод) управление может использовать мощь многих устройств по переработке информации, ибо ступени в этой системе управления устойчивы и управлять такими ступенями возможно с помощью отдельных устройств переработки информации.

Как уже отмечалось, у ступеней должны быть однотипные, но индивидуальные цели игры. При использовании устройств переработки информации в отдельных ступенях системы весьма существенна также автономность этих центров управления. Должно быть обеспечено «служение» цели игры, это должно быть запрограммировано, это должно быть выгодно ступени — в остальном она должна быть автономной. Лишь в этом случае может быть польза от применения множества устройств переработки информации, лишь тогда будет проявляться связанное с этим увеличение быстродействия и памяти. Если решение, принятое в автономной ступени, не будет использовано на уровне вышестоящей ступени, то ресурсы будут расходоваться неэкономно.

Итак, мы рассмотрели три типа многоступенчатой системы управления (А, В и F), которые различаются размещением центров управления в ступенях системы. Классифицируем теперь многоступенчатые системы типа В по приоритету целей игры в ступенях системы при принятии решения.

У каждой ступени есть своя местная, «эгоистическая» цель; но местная цель ступени, которая является совокупностью ступеней, является общей для ступеней нижестоящих. Лишь местная цель последней, высшей ступени будет общей целью всей многоступенчатой системы, общей целью всех ступеней системы управления.

Если при принятии решения исходить из местных целей, то в результате будет принят оптимальный вариант, который будет функцией местных интересов ступеней.

Принятие решения, базирующегося на местных интересах, не означает, что интересы общие игнорируются полностью, поскольку эти общие интересы могут совпадать с местными. Если же исходить из общих интересов, это не значит, что местными всегда пренебрегают, ибо они могут совпадать с общими интересами.



Суть дела в том, какие интересы преобладают при принятии решения — местные или общие?

В программах ступеней, таким образом, должна быть двойственность целей — местной ступени и вышестоящих ступеней. По приоритету этих целей и можно разделить многоступенчатые системы В на два типа:

С — системы, в которых программы управления ступеней отдают предпочтение местным интересам. Интересы вышестоящих ступеней они учитывают лишь тогда, когда эти более общие цели не находятся в противоречии с местными целями.

Е — системы, в которых программы управления ступеней отдают предпочтение общим целям. Местные цели они учитывают лишь тогда, когда эти местные интересы не вступают в противоречие с целями общими.

#### ПРЕИМУЩЕСТВА ОБЩЕЙ ЦЕЛИ

Если в программах центров управления ступеней многоступенчатой системы преобладают общие цели (тип Е), то оценка оптимального варианта для этой системы будет выше, чем оценка того же оптимального варианта, когда в программах центров управления ступеней преобладают местные цели (тип С). В шахматной игре данное положение может быть подтверждено практикой \*. Вообще, это положение хорошо всем известно: басня о лебеде, раке и щуке тому иллюстрация.

Пусть наша система будет, для простоты, двухступенчатой и типа С. Обозначим первые ступени (их общее число равно  $k$ ) индексами  $1, \dots, i, \dots, k$ , а систему в целом — индексом  $S$ . Покажем, что если перейти от управления многоступенчатой системой типа С к управлению системой типа Е, то можно не только обеспечить достижение оценок оптимальных вариантов ступеней для типа С, но и получить дополнительный выигрыш.

Обозначим оценку оптимального варианта всей системы  $S$  типа С индексом  $D'$ , а типа Е индексом  $D''$ . В соответствии с положением, приведенным выше,  $D'' > D'$ , а разность этих двух оценок  $\Delta D = D'' - D'$ .

При управлении нашей системой  $S$  по методу С оценки оптимальных вариантов ступеней (от 1-й до  $k$ -й) будут  $d'_1, \dots, d'_n, \dots, d'_m, \dots, d'_k$ , а сумма их будет равна величине  $d'_1 + \dots + d'_n + \dots + d'_m + \dots + d'_k = D'$ .

---

\* Как-то мне пришлось читать замечание одного ученого-кибернетика о том, что шахматы — более сложная задача, нежели экономика и политика. Это заблуждение; шахматная игра является моделью многоступенчатой системы управления с единым центром управления, и поэтому она в принципе проще, чем многоступенчатая система с множеством центров управления у ступеней, что характерно для сферы экономики и политики.

Теперь будем управлять системой  $S$  по методу  $E$ . В этом случае получим оптимальный вариант всей системы с оценкой  $D''$ , равной сумме оценок оптимальных вариантов по ступеням ( $d''_1, \dots, d''_n, \dots, d''_m, \dots, d''_k$ ):  $d''_1 + \dots + d''_n + \dots + d''_m + \dots + d''_k = D''$

Примем, что оценки  $d''_1, \dots, d''_n$  больше соответствующих оценок  $d'_1, \dots, d'_n$ ; оценки  $d''_{n+1}, \dots, d''_m$  те же, что оценки  $d'_{n+1}, \dots, d'_m$ , а оценки  $d''_{m+1}, \dots, d''_k$  меньше оценок  $d'_{m+1}, \dots, d'_k$ . Оставим оценки  $d''_{n+1}, \dots, d''_m$  без изменения, оценки  $d''_1, \dots, d''_n$  понизим до оценок  $d'_1, \dots, d'_n$  соответственно, а оценки  $d''_{m+1}, \dots, d''_k$  увеличим до оценок  $d'_{m+1}, \dots, d'_k$  также соответственно. Тогда получим прогноз при управлении системой при игнорировании местных интересов

$$D'' = D' + \Delta D = d'_1 + \dots + d'_k + \Delta D.$$

Рассмотрим случай, когда оптимальный вариант был определен «точно»; мы получим по методу  $E$  выгоду, равную  $D''$ . Израсходовав часть этой выгоды

$$D' = d'_1 + \dots + d'_k$$

на распределение благ по ступеням  $1, \dots, k$ , мы еще сохраним величину  $\Delta D$  на дополнительное распределение благ по ступеням.

Таким образом, управление системой типа  $E$  может дать большее значение оценок вариантов по ступеням системы. При этом в системе *ничего не надо менять*, кроме способа управления, как этой сейчас будет показано.

#### О МЕТОДЕ СБЛИЖЕНИЯ ОПТИМАЛЬНЫХ ВАРИАНТОВ СТУПЕНЕЙ ПРИ УПРАВЛЕНИИ ТИПОВ $C$ И $E$

Продолжим рассмотрение двухступенчатой системы управления  $S$ . Выделим одну из первых ступеней  $i$ ; при управлении типа  $C$  оценка оптимального варианта для этой ступени будет  $d'_i$ ; при управлении типа  $E$  —  $d''_i$ . Примем, что  $d'_i \neq d''_i$ , и рассмотрим сначала случай, когда  $d'_i > d''_i$ . Естественно, что программа управления из-за преобладания местных, эгоистических интересов не примет ни управления типа  $E$ , ни связанной с ней оценки  $d''_i$ , ибо в этом случае система  $i$  получит меньше благ. Какие же изменения программ управления ступенью  $i$  и системой  $S$  могут дать желаемый результат?

Поскольку после определения выгоды  $d'_i$  по методу управления  $C$  в программе управления ступенью  $i$  нет каких-либо ограничений в достижении выгоды, то прежде всего надо дополнительно ввести в программу ступени  $i$  ограничение выгоды значением  $d'_i$ ; эта выгода должна быть предельной. После этого можно переходить на метод управления типа  $E$ ; при этом, чтобы ступень  $i$  не

получила убытка, равного  $d'_i - d''_i$ , следует переоценить (в данном случае — увеличить) стоимость материальной единицы только для этой ступени; в шахматах это означало бы увеличение средней стоимости тех фигур, которые были бы выиграны в этой ступени управления (например, в зоне). Увеличение этого «курса» должно быть таково, чтобы в новом исчислении выгода  $d''_i$  была бы равна выгоде  $d'_i$  в старом исчислении. При двух этих изменениях (ограничении выгоды значением  $d'_i$  и изменении курса стоимости) цель игры эгоистической программы ступени  $i$  не противоречила бы общей цели системы  $S$ . Первое изменение (ограничение выгоды) относится к ступени  $i$ ; второе изменение (изменение курса для системы  $i$ ) относится к системе  $S$ .

При соотношении  $d'_i < d''_i$  изменение курса должно быть обратным, а при  $d''_i < 0$  изменение курса должно быть связано с изменением знака стоимости.

При этих изменениях в местных и общей программах управления система  $S$  может получить в случае применения метода  $E$  выгоду  $\Delta D$  без принципиальной ломки (изменения характера целей) эгоистических программ ступеней.

#### ПРОГРАММЫ ЭВМ И ЧЕЛОВЕКА

До сих пор подразумевалось, что в центрах управления используется ЭВМ. Но известно, что в наши дни в системах управления в основном используется человек. Какими же особенностями обладает человек по сравнению с ЭВМ в системах управления?

Различие между ЭВМ и человеком с кибернетической точки зрения формализуется просто. Программа ЭВМ, составленная программистом, может быть любой (в пределах ресурсов ЭВМ). Человек же не может получить программу таким элементарным путем; его программа формируется медленно в процессе общения с внешним миром при использовании программы самообучения.

Менять программу человека, что и говорить, дело сложное. Но все же воздействовать на эту программу можно, меняя реакцию внешнего мира, используя обратную связь и программу самообучения.

Примеров этому — хоть отбавляй: ленивый парень призывается в армию, лентяем он, может быть, остается, но это не проявляется. Он знает, что последует, если солдатские обязанности не будут выполнены. Водитель не прочь выпить, но на работу в автобазу он является трезвым — наказание может быть грозным. Молодая мать отказывается от своей страсти — ходить в театр — бабушка заболела и не с кем оставить ребенка. Инженер считает, что его начальник принял ошибочное решение, но избегает споров, опасаясь, что будет отстранен от интересной работы, и т. д.

Программа ЭВМ не может быть эгоистичной — с точки зрения

ЭВМ (она может быть эгоистичной лишь постольку, поскольку она «печется» о местных интересах той системы, которой ЭВМ управляет). Программа человека может включать собственные эгоистические интересы, противоречащие интересам той системы, которой человек управляет, и это может отрицательно повлиять на управление системой.

Если у ЭВМ выбор решения и его принятие суть понятия однозначные, то у человека они могут различаться. Человек может, исходя из интересов системы управления, выбрать один вариант, а принять другой — исходя из своих эгоистических интересов.

Таким образом, с кибернетической точки зрения искусственный интеллект предпочтительнее, нежели естественный. Вопрос заключается лишь в том, что пока еще нет в наличии мощного искусственного интеллекта...

#### ПРОБЛЕМА РАСПРОСТРАНЕНИЯ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Если в отношении мощного исполнительного органа систем управления (имеется в виду ядерное оружие и вообще ядерная энергия) в наши дни существует проблема нераспространения, поскольку чем большее число стран обладают этой энергией, тем вероятнее злоупотребление этим мощным средством, то в отношении мощного устройства по переработке информации (имеется в виду искусственный интеллект) возникает проблема скорейшего его распространения.

Следует иметь в виду, что проблема нераспространения ядерного оружия существует лишь потому, что ошибочное решение о его применении (злоупотребление этим оружием) может быть принято устройством по переработке информации, не соответствующим по уровню мощи исполнительного органа. Если будет создан искусственный интеллект, по уровню соответствующий современным мощным исполнительным органам, то в значительной мере снижается опасность возникновения термоядерной войны\*.

В этом случае возникает новая проблема — проблема распространения искусственного интеллекта, как одного из средств обеспечения безопасности народов. Если сейчас действуют меры, препятствующие распространению мощных компьютеров (эмбарго на продажу ЭВМ), то как только в различных важных областях управления появятся мощные программы, следует не только со всей возможной скоростью отменить все запреты, но и поощрять распространение по земному шару мощной вычислительной техники.

Частичная отмена эмбарго на продажу сильных компьютеров обязательна уже сейчас; для научных исследований в области

---

\* Об этом автору уже приходилось писать ранее [1].

искусственного интеллекта должны предоставляться мощные ЭВМ.

Эти соображения базируются на принципиальном различии в вопросе распространения ядерной энергии и искусственного интеллекта. И ядерная энергия, и искусственный интеллект могут служить как мирным, так и военным целям. Но сколько бы ни было успешным распространение ядерной энергии в мирных целях, это не может снизить опасности применения этой энергии и в военной области. И напротив, чем шире будет применяться искусственный интеллект в мирной жизни, тем безопаснее будут чувствовать себя люди, так как они станут умнее в том смысле, что будут находить более сильные оптимальные варианты.

## 2. МЕТОДЫ ОГРАНИЧЕНИЯ ДЕРЕВА ПЕРЕБОРА

### УСЕЧЕНИЕ ДЕРЕВА ПЕРЕБОРА

Когда дерево перебора велико, его усечение неизбежно. Здесь и рассматриваются лишь подобные задачи.

Усечение может быть сделано близко к исходной позиции или далеко. Последнее возможно, если дерево узкое (рис. 5); при широком дереве неизбежно близкое к исходной позиции усечение, т. е. предельная длина вариантов малая.

Предельная длина вариантов оказывает большое влияние на точность решения. Что ждет нас за предельной длиной вариантов? Это можно определить с трудом, и к этому вопросу мы еще вернемся. Мы можем более точно прогнозировать перспективы, пока есть возможность оценивать варианты перебора: чем варианты длиннее, тем прогноз будет точнее.

Чем оптимальный вариант длиннее, тем медленнее мы продвигаемся к его окончанию и тем легче нам исправить наш прогноз, если это окажется необходимым.

Например, шахматный мастер высчитал оптимальный вариант на 10 полуходов. Сделан первый ход по варианту — при той же предельной длине варианта можно продлить вариант на один полуход. Варианты различаются всего на 10%; мы все еще находимся сравнительно далеко от той области, где вариантов вообще нет (они не сосчитаны), где могут быть любые неожиданности.

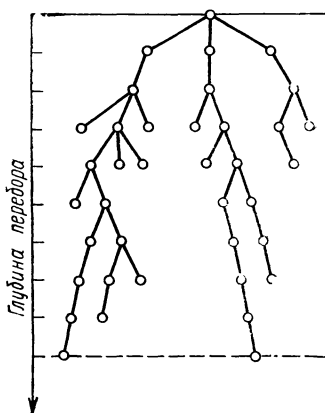


Рис. 5. Узкое и глубокое усеченное дерево перебора

Если ЭВМ считает варианты всего на три полухода, то один полуход укорачивает длину варианта на 33%, мы оказываемся существенно ближе к неизвестному будущему и здесь возможность просчета в прогнозировании более вероятна.

### ЦЕЛЬ НЕТОЧНОЙ ИГРЫ

Как только введено усечение дерева перебора, точная цель игры потеряла свое значение. Необходимо вместе с усечением дерева ввести цель неточной игры. Без этого игра будет бесцельной, она не может быть сильной. Цель неточной игры способствует формированию узкого и глубокого дерева.

В шахматах цель неточной игры — выигрыш материала. Подобная цель должна быть найдена в любой «игре», моделирующей систему управления, в любой неточной задаче. Пытаться решать неточную задачу, не формализовав цель неточной игры, — потерянное время. Эта цель — основа сильного алгоритма решения неточной задачи, без цели узкое и глубокое дерево не создать. Почему именно так, будет ясно из дальнейшего.

Цель игры позволяет установить, к чему надо стремиться; именно в этом случае и можно определить возможности, заведомо не способствующие достижению цели, и исключить их из перебора. Цель позволяет нам определить направление перебора.

### ОЦЕНОЧНАЯ ФУНКЦИЯ

Если цель игры позволяет определить направление перебора, то оценочная функция позволяет закончить и оценить вариант перебора. Цель помогает сформировать дерево перебора; оценочная функция дает возможность подвести итоги.

Оценочная функция действует во взаимосвязи с целью неточной игры; поэтому и функция эта является неточной. В отличие от цели, которая должна быть единой, оценочная функция должна состоять из двух составляющих: первая составляющая позволяет оценить результаты, достигнутые при выполнении цели в пределах усеченного дерева перебора, а вторая — прогнозировать возможные достижения цели за пределами усеченного дерева.

Первая составляющая оценочной функции дает точный ответ (в пределах установленной ограниченной точности) на вопрос о достигнутых целях; вторая составляющая оценки (позиционная оценка) дает предположительный ответ на вопрос о том, что будет дальше, когда граница усечения отодвинется. Суммой этих двух составляющих определяется оценка законченного варианта.

## ОБРЫВ ВАРИАНТА

Соотношение между стоимостью цели и оценкой варианта позволяет решить вопрос об обрыве варианта еще до того, как вариант достиг предельной длины. Повторяем, это возможно лишь тогда, когда в алгоритме есть и цель игры, и оценочная функция. Когда часть вариантов заканчивается до предельной длины, дерево перебора уменьшается (рис. 5).

Как уже указывалось, во всех известных шахматных программах цель неточной игры отсутствует и поэтому нет вариантов, короче предельной длины.

## ОТСЕЧЕНИЕ ВЕТВЕЙ

Известный метод граней и оценок ( $\alpha$ — $\beta$ -отсечение) в том виде, как он обычно применяется, в нашем случае не действует; здесь применим простой метод, неразрывно связанный с наличием цели игры.

Если при подъеме по варианту оценка текущего оптимального варианта (ТОВ) в более верхнем узле не меньше суммы текущей оценки (той, что была получена в данном узле при спуске по варианту) и стоимостей целей (мишеней) того же цвета, что и данный узел, то далее строить поддерево нечего, ибо таким путем нельзя будет получить ТОВ с большей оценкой.

## ГОРИЗОНТ

Метод горизонта для ограничения дерева перебора был введен автором в 1968 г. [1]. Этот метод применяет человек в своей повседневной деятельности.

Пешеход переходит дорогу — он обязательно посмотрит, на каком расстоянии слева и справа находятся автомашины. Если автомашины далеко, т. е. вне «горизонта», переходить шоссе можно; если автомашины находятся в 50 метрах, пешеход подождет, когда автомобили проедут.

Экономисты участвуют в проектировании нового завода. Чтобы определить, будет ли обеспечен завод рабочей силой, они должны изучать состав и численность населения в определенном радиусе дальности — здесь у экономистов есть свой горизонт.

Экспедиция продвигается на санях к северному полюсу, и надо разведать ледовую обстановку. Это будет сделано, скажем, в пределах одного километра, но отнюдь не до конечной цели экспедиции — полюса. И здесь есть свой «горизонт».

Это грубый, но неизбежный способ ограничения задачи.

Не следует смешивать горизонт с предельной длиной варианта (до усечения дерева). Это разные понятия. Так, ледовые развед-

чки, наметив первую цель на расстоянии одного километра, могут, продолжая разведку, двигаться еще дальше, но каждый раз следующий пункт назначения у них будет не далее одного километра.

Предельная длина варианта всегда превышает горизонт. Поэтому усечение дерева перебора является более мягким способом ограничения задачи, чем ограничение по горизонту. Это относится только к случаю, когда формируется узкое и глубокое дерево.

Дадим следующее определение горизонта: горизонт — это предельное время, которое должно быть затрачено для достижения цели. Поэтому те возможные цели, достижение которых связано с расходом времени больше предельного, находятся вне видимости; программа, решая задачу, эти цели не замечает.

### ДВА ДЕРЕВА. МАТЕМАТИЧЕСКОЕ ОТОБРАЖЕНИЕ (МО)

Обычно решение неточной задачи начинается с построения дерева перебора. Но если задача решается по методу формирования узкого и глубокого дерева, работа начинается не с построения дерева перебора.

Прежде всего следует найти цель в пределах принятого горизонта, и лишь после того, как цель найдена, после того, как стало известно, в каком направлении надо начать перебор и к чему следует стремиться, должно быть начато построение дерева перебора. В процессе построения дерева перебора в пределах горизонта могут появляться новые цели, и тогда становятся известными новые направления для дальнейшего формирования дерева перебора.

Таким образом, кроме дерева перебора, формируется дерево целесообразных действий. Оба дерева взаимосвязаны, тесно переплетены. Но следует помнить, что все начинается с образования дерева целесообразных действий, оно лежит в основе основ. Дерево перебора может расти по направлениям, указанным деревом целесообразных действий, хотя дерево этих действий формируется по мере разрастания дерева перебора. Это дает большую выгоду в экономии ресурсов, ибо дерево перебора получается целенаправленным, а дерево действий формируется в пределах, которые необходимы для развития дерева перебора — не больше и не меньше.

Дерево целесообразных действий было названо математическим отображением задачи. Неточная задача решается лишь в пределах сформированного математического отображения (МО).

Дерево перебора — понятие общее для всех типов неточных задач. Дерево целесообразных действий (математическое отображе-



ние) всегда специфично для данной конкретной неточной задачи. В шахматной игре математическим отображением является дерево траекторий целесообразного движения фигур.

Если дерево перебора в процессе его формирования непрерывно растет, то МО должно поддерживаться ограниченным и возможно меньшим (МО нужно лишь для управления формированием дерева перебора в данном его узле). Чем МО меньше, тем быстрее его можно обследовать и принять соответствующее решение о направлении перебора. Поэтому часть МО, которая уже сыграла свою роль и больше не нужна, должна быть стерта в памяти ЭВМ; та же часть МО, которая не нужна в данном узле дерева перебора, но может еще понадобиться в дальнейшем, также должна быть исключена из действующего в данном узле МО, но сохранена в памяти. Ограниченное «действующее МО» помогает формировать дерево перебора, как световое пятно от фар автомашины помогает водителю выбирать направление движения.

### МНОГОСТУПЕНЧАТОСТЬ СИСТЕМЫ

Уже отмечалось, что многоступенчатая структура системы управления способствует образованию узкого и глубокого дерева, поскольку перебор в отдельной ступени прекращается, как только в какой-то мере достигается цель игры в этой ступени либо становится ясным, что цель недостижима. Однако не следует забывать (и это также отмечалось), что многоступенчатость системы в тех случаях, когда в отдельных ступенях имеются центры управления и задачу можно решать параллельно (по ступеням), приводит к увеличению ресурсов, используемых для решения задачи. А это, в свою очередь, способствует формированию более глубокого дерева перебора.

Следует отметить еще одно преимущество многоступенчатой системы по сравнению с единой системой управления. Множественность целей позволяет избежать включения в перебор некоторых ступеней управления, если мишени (цели) в этих ступенях таковы, что включение их в перебор заведомо не способно изменить оптимального варианта (прогноза). Кроме того, по стоимости целей и вероятности их достижения можно установить приоритет (очередность) включения ступеней в перебор, что также может способствовать рациональному формированию дерева перебора.

Таким образом, прекращение или отказ от перебора в отдельных ступенях, а также возможное увеличение ресурсов при многоступенчатости системы способствуют более глубокому решению (из-за большей предельной длины вариантов и более далекого усечения дерева перебора).

Итак, установлены методы ограничения дерева перебора, методы получения узкого и глубокого дерева (усечение, цель игры, оценка варианта, обрыв варианта, горизонт, МО, многоступенчатость). Но без общих принципов, определяющих применение этих методов, удачного узкого и глубокого дерева не получить. Каковы же эти принципы?

1. Принцип надежды (или, скорее, принцип безнадежности, но поскольку он был найден еще в 1968 г. как принцип надежды, то не будем менять его названия): до тех пор, пока с какой-либо возможностью связана надежда на что-то хорошее, на достижение цели, эта возможность включается в МО или дерево перебора.

Существенно здесь не столько то, что эти возможности учитываются при решении задачи, сколько то, что возможности, с которыми уже не связаны надежды на что-то хорошее, не включаются в МО и в дерево перебора — весьма сильный путь с точки зрения ограничения задачи.

Так действует человек. Абитуриент, потерявший слух, не будет держать вступительные экзамены в консерваторию, хотя он до болезни успешно учился в музыкальной школе. Шахматист не станет рассматривать взятие пешки (при отсутствии других целей), если его ладья погибает при попытке подойти к этой пешке.

2. Принцип наибольшего выигрыша: новая возможность (в МО и в деревне перебора) будет рассматриваться лишь в том случае, если есть надежда получить что-то лучшее по сравнению с уже рассмотренным.

Например, мастер может выиграть коня за пешку; теперь он рассматривает новую возможность — выигрыш пешки. Эта новая возможность отвергается. Но вот есть еще одна возможность — атака на короля. В соответствии с приведенным выше принципом эта атака непременно должна быть включена в рассмотрение.

Абитуриент колебался — куда подать заявление о поступлении: в МИФИ или МФТИ? Подал в МФТИ, но выяснилось, что в МИФИ конкурс существенно меньше. Молодой человек обязан рассмотреть вопрос: не следует ли взять документы назад и подать их в МИФИ? Если же в МИФИ был такой же конкурс, то вопрос и не возник бы!

Этот принцип также приводит к сокращению рассматриваемых возможностей.

3. Принцип своевременности: рассматривать надо лишь те возможности, при которых «действующие лица» могут успеть принять участие в игре. Не следует привлекать к контролю какого-либо поля шахматной доски фигуру, которая лишь тогда успевает начать контролировать это поле, когда неприятельская фигура его уже проскочит. Командующий зоной ПВО не будет посылать на

перехват учебный самолет — летчик все равно опоздает, но и самолет-перехватчик он пошлет с такого аэродрома, чтобы летчик успел атаковать воздушную мишень.

Принцип своевременности (а может несвоевременности?) также является могучим средством устранения возможностей, потерявших смысл или не имеющих смысла!

Эти три принципа взаимно связаны: именно тогда возможности исключаются из рассмотрения, когда нет надежды на их своевременное участие в игре или заведомо нет надежды на больший успех в достижении цели.

## УЛУЧШЕНИЕ РЕЗУЛЬТАТОВ ПЕРЕБОРА

В процессе развития МО неизбежно возникает вопрос: полезно ли данное конкретное расширение МО, есть ли надежда на большую выгоду, на наилучший результат перебора? Иначе говоря, можно ли в случае расширения МО получить новый оптимальный вариант с более высокой оценкой? Используя принципы надежды, наибольшего выигрыша и своевременности, и надлежит принять решение. В общем виде ответить на поставленный вопрос нельзя, ответ можно получить лишь в конкретной ситуации для данной задачи. Но все же некоторые общие рекомендации могут быть даны.

Математики исследуют дерево перебора (с помощью минимаксной процедуры) вариант за вариантом. После того, как найден и сохранен в памяти ЭВМ один вариант, находится второй. Варианты сравниваются с помощью оценочной функции, и вариант с большей оценкой сохраняется в памяти (другой — стирается); затем определяется еще один вариант, снова происходит сравнение и т. д. Нетрудно заметить, что при этом в памяти ЭВМ находятся одновременно только два варианта, один из которых будет ТОВ.

Если эта процедура поиска оптимального варианта подходит для решения задачи при включении всех возможностей в усеченное дерево перебора, то для решения, основывающегося на формировании узкого и глубокого дерева, эта общепринятая методика не может быть использована.

Узкое и глубокое дерево можно сформировать лишь в том случае, если МО будет ограниченным, если оно не будет разрастаться. Этого можно достичь, применяя упомянутые три принципа. Но применить эти принципы, запоминая лишь два варианта из дерева перебора, невозможно.

Вся суть в том, что когда ниже определенного узла поддерево сформировано, и в этом узле надо принять решение о возможном расширении МО, мы можем решить этот вопрос только тогда, когда будут исследованы варианты этого поддерева. Лишь собрав

определенную информацию об этих вариантах (в том числе и о ТОВ), можно ответить на вопрос: есть ли надежда на получение в данном узле нового ТОВ с более высокой оценкой?

Поскольку надо собирать информацию о вариантах поддерева, его надо запоминать. В этом состоит одно из отличий метода решения задачи при формировании узкого и глубокого дерева от метода, основанного на включении в дерево перебора всех возможностей.

Хорошо известно, что шахматный мастер помнит не два варианта, а дерево перебора. Он запоминает то сформированное окончательное дерево перебора, которое и является основой для принятия решения.

Итак, мы рассмотрели методы, ограничивающие дерево перебора. Все они могут быть разделены на два типа: по времени и по материалу (цели игры). Действительно, усечение дерева перебора производится по времени, горизонт ограничивает наибольшее время, необходимое для достижения цели, принцип своевременности связан со временем по самой своей сути. Принцип наибольшего выигрыша, улучшение результатов перебора, обрыв варианта, оценочная функция связаны с материалом (целью игры). Можно добавить, что и позиционная оценка, казалось бы не связанная с материалом, зависит от него — это будет ясно из дальнейшего.

### 3. ПОИСК РЕШЕНИЯ И ОПЫТ ПРОШЛОГО

#### ОРИГИНАЛЬНАЯ СИТУАЦИЯ

Когда поиск решения идет по программе принятия решения в оригинальной ситуации — в этом случае формируется узкое и глубокое дерево по всем правилам и в надлежащем объеме. Разумеется, это связано со значительным расходом ресурсов устройства по переработке информации (времени и памяти). В этом случае опыт прошлого используется лишь косвенно; он сказывается при формировании алгоритма, на структуре программы. Но кроме этого опыта, имеющего отношение к методике решения, обычно существует большой запас знаний, накопленный за столетия, — этот опыт специфичен для той конкретной задачи, которую необходимо решить.

Опыт прошлого обязательно должен быть использован, и специалист, решая ту или иную задачу, всегда использует свои знания, свой опыт, если ситуация позволяет. Это приводит к экономии ресурсов, т. е. к более быстрому или более глубокому решению.

Как это происходит, сейчас будет изложено.

#### ПОИСК РЕШЕНИЯ ПО АССОЦИАЦИИ

Человек, решая какую-либо неточную задачу, при возникновении ситуации, *напоминающей* ему ту, что уже встречалась в практике людей, использует опыт прошлого.

Что же это такое — «напоминающая» ситуация? Это, безусловно, ситуация, не полностью совпадающая с ранее встречавшейся, — наблюдается лишь частичное совпадение. Но это частичное совпадение дает ключ для более простого пути принятия решения, чем тот, который приходится искать в оригинальной ситуации. В чем же может состоять этот более простой путь? Только в том, что для принятия решения необходимо попытаться сформировать часть МО (и соответственно часть дерева перебора). Если обратиться к шахматам, то надо сформировать именно ту часть МО, где играют совпадающие фигуры. Суть дела в том, что ранее, в прошлом, эти фигуры уже играли; сейчас надо проверить — приводят ли к успеху те же варианты, что встречались раньше? Если да, то на этом и кончается поиск решения в данном поддереве — остальные части МО и поддерева, которые пришлось бы формировать, если этот ассоциативный метод дал бы отрицательный результат, уже не нужны.

Ассоциативный метод подсказывает программе направление перебора вариантов; иначе говоря, он определяет приоритет игры в данной ступени управления.

Раз дело сводится к формированию части МО, к частичному формированию дерева перебора, это означает экономию ресурсов. Экономленны время и объем памяти можно использовать при поиске решения в оригинальной ситуации в дальнейшем. В целом найденные решения будут более глубокими.

#### ПОИСК РЕШЕНИЯ ПО СПРАВОЧНОМУ МЕТОДУ

Здесь все просто: если возникающая ситуация, в которой принимается решение, полностью совпадает с ситуацией, известной из прошлого опыта, то решение найдено — продолжать вариант не нужно. Раз ситуация встречалась в прошлом, то оценка ее известна; стало быть, становится известной и оценка варианта, и вариант надлежит оборвать. В этом случае мы получаем оценку варианта (с того момента как ситуации совпали) — без дальнейшего формирования МО и дерева перебора. Весьма экономный путь!

Хитрость здесь состоит лишь в том, что следует стремиться получить выгодное совпадение ситуаций. Это приводит к направленному формированию МО и дерева перебора. Но как только совпадение достигнуто и оценка известна, — вариант закончен.

Итак, мы рассмотрели три типа поиска решения. Первый — в оригинальной ситуации, когда опыт прошлого помочь не может, здесь надо обычным путем строить МО и соответствующее дерево перебора. Второй — когда частично ситуация уже встречалась ранее, здесь известно дальнейшее направление формирования части МО и соответствующего поддерева. Если это имеет успех, приводит к совпадению ситуаций по существу, а не только к внешнему сходству, то дальнейшее формирование МО не нужно и здесь экономятся ресурсы. Для применения этого метода поиска решения необходимо частичное формирование МО и соответствующего поддерева. И, наконец, третий тип поиска, когда происходит полное совпадение ситуации в данной задаче и ситуации, известной из прошлого (и для этого может быть необходим направленный перебор). Оценка варианта становится известной без какой-либо дальнейшей затраты ресурсов. Это — справочный метод поиска решения.

Человек — специалист в определенной области — применяет все три метода поиска решения, когда решает неточную задачу по своей специальности. Для этого он и получает специальное образование. Надо полагать, что и ЭВМ должна действовать таким же образом: ей должен быть передан опыт прошлого, специальные знания и ЭВМ должна быть обучена пользованию всеми тремя методами.

И когда создается шахматная программа для ЭВМ, машине должны быть переданы специальные знания в области дебюта, миттельшпиля и эндшпиля, причем в такой форме, чтобы эти знания можно было бы использовать как при ассоциативном, так и при справочном методе поиска решения.

#### **4. ПРИМЕР РЕШЕНИЯ НЕТОЧНОЙ ЗАДАЧИ (ИГРА В ШАХМАТЫ)**

Общие положения, на которых должно базироваться решение неточной задачи, были изложены. Проиллюстрируем применение этих положений на конкретном примере — составлении шахматной программы для ЭВМ. Игра в шахматы, как это уже отмечалось, является типичной неточной задачей — моделью многоступенчатой системы управления с единым устройством по переработке информации.

Когда составлялся алгоритм для шахматной программы, была поставлена задача моделирования мышления шахматного мастера. Было принято, что человек за многовековую историю шахмат рационально приспособился к этой неточной задаче — поэтому

и было решено, не мудрствуя лукаво, передать человеческий опыт ЭВМ.

Мастер, играя в шахматы, владеет двумя методами игры (это также уже отмечалось): 1) алгоритмом поиска решения (хода) в оригинальной ситуации (позиции) и 2) алгоритмом поиска хода в позиции, которая частично или полностью встречалась ранее в практике шахматных мастеров (в этом последнем случае мастер и использует свои специальные знания в области шахмат). ЭВМ должны быть переданы оба метода игры; поэтому ЭВМ, кроме алгоритма поиска хода, должны быть сообщены и специальные сведения по шахматам из дебюта, миттельшпиля и эндшпиля.

Почти все программы, играющие в шахматы, имеют библиотеку дебютов — это простое дело; библиотек позиций из середины игры и концов игр у этих программ нет. Далее будет объяснено, почему программы, основанные на включении в усеченное дерево всех возможных ходов, органически лишены способности пользоваться библиотеками миттельшпиля и эндшпиля.

## **А. ПОИСК ХОДА В ОРИГИНАЛЬНОЙ ПОЗИЦИИ**

### **УСЕЧЕННОЕ ДЕРЕВО ПЕРЕБОРА ХОДОВ**

Шахматная игра — сложная задача, решаемая с помощью перебора ходов. Теоретически этот перебор является конечным, и поэтому игра в шахматы — конечная игра. Практически дерево перебора ходов столь неизмеримо велико, что шахматист не может обойтись без усечения дерева перебора ходов, после чего шахматы становятся, как правило, неточной задачей.

Уже отмечалось, что если решать задачу на базе включения в дерево перебора всех ходов, то дерево стремительно разрастается вширь (см. рис. 1) и задача становится безнадежной. Если же решать задачу, формируя узкое и глубокое дерево (как это делает шахматист), то есть надежда на успешное решение задачи о поиске хода.

Глубина усечения (предельная длина вариантов) является первым ограничением задачи. При глубоком дереве перебора изменение предельной длины будет тонким методом ограничения и, меняя глубину усечения, шахматист регулирует число ходов, включаемых в дерево перебора.

Формирование узкого и глубокого дерева и было принято как основное требование при составлении программы. Алгоритм этой шахматной программы, как это предполагается, подобен алгоритму игры мастера; сформированное дерево перебора должно быть узким и глубоким, оно должно содержать малое количество ходов.

С чего же надлежит начать работу?

По изложенной ранее общей теории работу следует начинать с установления цели неточной игры — так и будем действовать.

Каждый мастер знает, к чему он стремится, когда сидит за доской, — к выигрышу материала. Дилетант придет в ужас от этого утверждения и — напрасно! Все остальные факторы (в том числе и позиционные), которые учитывает мастер в своих расчетах или, точнее говоря, при проведении минимаксной процедуры на усеченном дереве перебора ходов, служат достижению этой прозаической цели.

Цель эта типовая, она проявляется в самой различной конкретной форме, но по типу — неизменна.

Установив цель неточной игры, мы сразу получаем первый критерий, позволяющий в начале формирования дерева перебора определить ходы, лишенные смысла. Когда мы еще ничего не знаем о позиции, когда только начинается ее изучение, ходы, не приводящие к возможному выигрышу материала, к нападению фигур одного цвета на фигуры другого цвета, исключаются из перебора!

Вместо примитивной процедуры включения всех ходов в перебор здесь все начинается с определения ходов, лишенных смысла. Это нельзя сделать, не установив разумной цели игры.

Далее будет показано, что игра в шахматы (когда в шахматы играет мастер) представляется как модель трехступенчатой системы управления. Например, первой ступенью является атакующая фигура с траекторией нападения на фигуру другого цвета; ступень эта имеет свою конкретную цель — уничтожение атакующей фигуры, выигрыш материала, по цене равной стоимости атакующей фигуры. Второй ступенью будет зона игры, например совокупность фигур, препятствующих действию и поддерживающих действие атакующей фигуры; зона также в самой различной форме имеет цель — выигрыш материала. И, наконец, третья ступень (математическое отображение позиции — МО), представляющая собой совокупность зон; для третьей ступени характерна множественность элементарных целей игры, ибо в МО атакуются разные фигуры разного цвета, но достижение всех этих целей должно привести к выигрышу материала.

Отметим, что в шахматах цель неточной игры одного типа с целью точной игры. Не следует забывать, что в шахматах король имеет бесконечно большую стоимость (в шахматных программах неудобно давать бесконечно большую цену королю; поэтому в программах цена короля принята обычно равной 200). Но независимо от того, что принято (бесконечно большое число или 200), цена короля — материал, так что точная и неточная цели в шахматах однотипны.



Уже отмечалось, что цель игры позволяет решать вопрос о том, к чему следует стремиться, а оценочная функция используется для оценки как достигнутых результатов, так и тех результатов, которые могут быть достигнуты.

Какова же оценочная функция для шахматной игры? Читатель уже знает, что оценочная функция должна иметь две составляющие. Первая позволяет судить о том, насколько успешно может быть достигнута цель неточной игры (выигрыш материала) в пределах минимаксной процедуры на усеченном дереве перебора. Попросту говоря, мы подсчитываем итоги по соотношению материала, выигранного сторонами во всех вариантах в пределах усеченного дерева перебора ходов. Это ясно и просто. Сложнее обстоит дело со второй составляющей: она должна прогнозировать выигрыш материала в пределах той неизвестной пока части дерева перебора, которая еще не сформирована и лежит за линией отсечения. Речь идет о позиционной оценке.

Оценочная функция по Шеннону состоит, казалось бы, из многих частей, а в действительности — также из двух. Первая — по материалу, вторая — позиционная, которая «начинена» самыми различными усредненными позиционными факторами.

Подобный метод позиционной оценки в шахматах ошибочен. Позиционный фактор, который в одной ситуации дает положительную оценку, в другой может дать отрицательную. Например, сдвоенные пешки — хорошо это или плохо? Все зависит от ситуации. Иногда сдвоенные пешки, поскольку одна не может защищать другую, являются удобным объектом атаки. Но бывают случаи, когда сдвоение пешек приводит к контролю полей, полей, через которые проходят важные коммуникации (траектории движения фигур), — тогда сдвоенные пешки весьма полезны. То же можно сказать и о других позиционных факторах, входящих в позиционную часть типовой оценочной функции, предложенной Шенноном.

В данной шахматной программе было принято принципиальное решение о позиционной составляющей оценочной функции, в основу позиционной оценки был положен контроль полей, из которых состоят траектории, входящие в МО. Сторона, контролирующая большее число полей, имеет позиционный перевес. Вопрос о позиционной оценке будет подробно рассмотрен далее. Заметим, однако, что контроль полей определяется результатом возможного размена на поле, т. е. соотношением по материалу, и позиционная оценка нужна лишь для того, чтобы прогнозировать выигрыш того же материала!

Цель неточной игры (материал) позволяет оборвать вариант перебора до того, как вариант достигнет предельной длины. Обозначим стоимость цели индексом  $m_{\text{ц}}$ , а  $m_{\text{т}}$  пусть будет текущей оценкой при спуске по варианту. Если рассматривать одну зону, то при условии  $|m_{\text{ц}}| > |m_{\text{т}}|$  вариант продолжается; при нарушении этого неравенства — прекращается.

В этом неравенстве выражаются принципы надежды и наибольшего выигрыша: пока есть надежда на достижение цели, пока потеряно (или выиграно) меньше стоимости цели — игра продолжается. Когда же потерян (выигран) материал, равный (или превышающий) стоимости цели, игра прекращается и вариант заканчивается (обрывается).

Суть дела здесь в том, что в шахматах (как и в других задачах) можно жертвовать материал только в том случае, если есть надежда выиграть больше пожертвованного; можно также жертвовать материал, если есть надежда избежать потери материала, превышающего пожертвованный.

Вариант может обрываться и по другим причинам, также связанным с возможным достижением (или невозможностью достижения) цели, например, когда исчезает цель или исчезает атакующая фигура — здесь мы можем наблюдать проявление принципов своевременности и надежды.

Когда же жертвуется материал — а жертвуется он лишь для того, чтобы в дальнейшем получить большую выгоду по сравнению с тем, когда он не жертвуется, — проявляются, как уже отмечалось, и принцип наибольшего выигрыша, и принцип надежды.

Вообще же вариант обрывается, когда прекращается игра в высшей ступени — совокупности всех ступеней. Но в невысших ступенях игра может прекращаться, когда вариант еще продолжается. Игра в этих ступенях прекращается, когда цель игры в ступени достигается или становится недостижимой, или когда теряется возможность вовремя поспеть к месту боя. К этому вопросу мы еще вернемся после того, как будет формализована многоступенчатая система — модель шахматной игры. В прекращении игры и обрыве варианта наглядно проявляется отличие данного шахматного алгоритма от прочих. Прекращение игры и обрыв варианта способствуют уменьшению числа ходов (узлов) в дереве перебора.

Отсечение ветвей (см. с. 50) также основано на принципах надежды и наибольшего выигрыша. Если сумма целей в узле меньше разности между текущей оценкой и оценкой ТОВ, то нет надежды на улучшение оценки ТОВ и надо прекратить дальнейший перебор ниже этого узла. Следует иметь в виду, что сумма целей может в узле менять свое значение при включении в игру зон нападения; при отсечении ветвей это должно быть учтено.

Таким образом, в этом алгоритме отсечение ветвей в узле никак не связано с методом граней и оценок; оно связано лишь с целью игры — когда улучшить результаты стремления к цели нельзя, ветви строить нечего.

## ГОРИЗОНТ

Нападений фигур одного цвета на фигуры другого цвета может быть неисчислимо множество. Цель неточной игры позволяет осмыслить действия фигур, но сама по себе не может ограничить их действия. Здесь следует ввести ограничение, основанное на уже упоминавшемся методе горизонта. В шахматной программе и введено ограничение — предельное время передвижения атакующей фигуры по своей траектории нападения. В методе горизонта проявляется принцип своевременности: нападения, которые не укладываются в предельное время (горизонт), исключаются из рассмотрения.

Известно, что в шахматах время измеряется в полуходах. Если взять предельный горизонт  $H_L$  равным четному числу полуходов, то предельное число передвижений атакующей фигуры по траектории нападения будет равно  $H_L/2$  (к траекториям отрицания это не относится).

Предельный горизонт  $H_L$  — грубое ограничение, ибо изменение горизонта приводит к резкому изменению числа узлов в дереве перебора. Когда фигур на доске много, игра носит открытый характер, обычно  $H_L$  равно 4 полуходам. В эндшпиле, когда фигур мало, да если еще дальнобойные фигуры хотя бы частично сняты с доски, при горизонте  $H_L=4$  число нападений резко сокращается и для получения хороших результатов при принятии решения (поиске хода) надо горизонт  $H_L$  увеличивать. Особенно велик он в пешечных концах, здесь предельный горизонт может быть равен и 10 полуходам.

## ШАХМАТЫ — ТРЕХСТУПЕНЧАТАЯ СИСТЕМА

В том, что играя в шахматы, сильный шахматист все свои расчеты начинает с нападения, ограниченного горизонтом, сомнений быть не может. Несомненно и то, что сильный шахматист видит не всю доску и не все фигуры, а лишь те, которые участвуют в расчетах и передвигаются лишь по некоторым полям доски. Совокупность этих фигур и их траекторий и образует МО — дерево траекторий, которое лежит в основе дерева перебора шахматных ходов.

Итак, две ступени (фигуры с их траекториями и МО) должны быть. А есть ли еще промежуточные ступени, не является ли шах-

матная игра трехступенчатой системой? Да, промежуточная ступень есть.

Все начинается с первой ступени — фигуры со своей траекторией, находящейся в пределах горизонта  $H_L$ ; у этой фигуры должна быть своя индивидуальная цель — тогда эта фигура нападающая. Если же фигура со своей траекторией, также находящейся в пределах  $H_L$ , принимает участие в игре, связанной лишь с борьбой за достижение какой-то индивидуальной цели другой фигуры, — тогда эта фигура не является нападающей. Фигуры, траектории которых определяются горизонтом  $H_L$ , были названы комлевыми.

Итак, комлевая и нападающая фигура стремится к завоеванию определенной цели. Комлевая, но не нападающая фигура стремится лишь изменить результат атаки нападающей фигуры на ее цель.

Комлевые фигуры не действуют в одиночку. У каждой из них есть своя «команда» из фигур того же цвета, которые ей помогают; при этом есть и неприятельская «команда» из фигур противоположного цвета, которые ей мешают. Совокупность комлевой фигуры и этих двух фигурных команд разного цвета образует зону игры — вторую ступень системы управления. Поскольку комлевые фигуры суть двух типов (нападения и связанные с нападением), то и зоны должны быть двух типов — нападения и связанные.

Таким образом, вторую ступень может образовывать как зона нападения, так и связанная зона. Связанными зонами были названы зоны контроля, блокады, деблокады, отступления (для исчезновения цели) и позиционная (для изменения знака размена на контролируемом поле). Вторая ступень (зона) всегда образуется одной стороной для улучшения результатов перебора (улучшения оценки ТОВ); если же эта вторая ступень является зоной нападения, то появляются еще новые цели (в этом также состоит возможность воздействия второй ступени этого типа на оценку ТОВ).

#### ПЕРВАЯ СТУПЕНЬ — ФИГУРА С ТРАЕКТОРИЕЙ

Условно первая ступень может быть двух типов. Во-первых, фигура и ее траектория могут быть комлевыми, лежать в основе зоны игры (второй ступени); предельная длина этой траектории ограничена предельным горизонтом  $H_L$ . Первая ступень такого типа формируется лишь тогда, когда образующаяся вторая ступень (зона) способна изменить оценку ТОВ в благоприятную сторону.

При втором типе первой ступени — фигуре с некомлевой траекторией, т. е. при фигуре, участвующей в команде (так называемой

отрицающей фигуре, действующей по траектории отрицания), дело обстоит сложнее. В соответствии с принципом своевременности предельный горизонт у этих фигур равен уже не постоянной, а переменной величине  $H_x$ . Эта отрицающая фигура связана в конечном итоге с борьбой за  $\alpha$ - или  $\beta$ -поле комлевой траектории зоны (как известно, на  $\alpha$ -поле фигура останавливается при передвижении по своей траектории,  $\beta$ -поле фигура проходит без остановки); чем номер этого  $\alpha$ -поля больше, т. е. чем оно дальше отстоит от начального  $\alpha_0$ -поля комлевой траектории, тем больше времени имеет отрицающая неприятельская фигура для того, чтобы успеть принять участие в борьбе за  $\alpha_i$ -поле, тем больше значение переменного горизонта  $H_x$ . Любопытно, что предельное значение  $H_x$  у отрицающей траектории контроля может даже превышать значение  $H_L$  на один полуход; это возможно при контроле последнего  $\alpha_k$ -поля комлевой траектории (при ходе контролирующей стороны), если комлевая траектория предельной длины. Сложность образования траекторий отрицания состоит в том, что предельное время  $H_x$  отводится для образования не только той траектории первого отрицания, которая непосредственно связана с  $\alpha$ -полем комлевой траектории, но и всех траекторий от первого до более высоких отрицаний. Траектория высокого отрицания привязывается к  $\alpha$ -полю траектории отрицания, меньшего на единицу, а в конечном итоге — к данному  $\alpha$ -полю комлевой траектории.

Главной особенностью образования траекторий отрицания, в отличие от образования комлевых траекторий, является то, что если в пределах  $H_x$  эти траектории могут существовать, они образуются безоговорочно. Единственным условием здесь является выполнение принципа своевременности.

Впоследствии будет установлено, что в соответствии с этим принципом своевременности происходит исключение и включение в МО уже сформированных траекторий отрицания.

Итак, если на формирование комлевых траекторий распространяются принципы как своевременности (горизонт  $H_L$ ) так и наибольшего выигрыша (улучшение оценки ТОВ), то при формировании траекторий отрицания учитывается лишь принцип своевременности. Да это и понятно: формирование зоны может происходить, когда еще нет данных об ее безусловной бесполезности — стало быть, траектории отрицания формировать надо.

Разумеется, принцип надежды действует при формировании и комлевых, и отрицающих траекторий — он действует всегда.

Следует учитывать, что не все фигуры со своими траекториями являются первыми ступенями системы управления. Так, комлевая фигура со своей траекторией нападения, несомненно, является первой ступенью многоступенчатой системы управления. У каждой ступени должна быть своя индивидуальная цель — здесь это требование удовлетворяется.



Переменный горизонт  $H_x$  определяет строго детерминированную структуру зоны: в зону включаются те отрицающие фигуры, которые в соответствии с временем, ограниченным  $H_x$ , успевают принять участие в игре. Как уже отмечалось, все траектории отрицания\*, удовлетворяющие условию  $H_x$ , безоговорочно включаются в игру. На рис. 6 изображены три зоны игры: нападения  $Z_0$  и связанные с ней зоны контроля и отступления ( $Z_1$  и  $Z_2$ ).

### ТРЕТЬЯ СТУПЕНЬ — СОВОКУПНОСТЬ ЗОН (МО)

Мы уже обсуждали условия формирования совокупности зон. В МО включаются лишь те зоны, с которыми связана надежда на улучшение оценки ТОВ.

Таким образом, объем МО зависит, в частности, и от первого хода в исходной позиции; при удачном выборе первого хода в исходной позиции дальнейшее включение зон может быть сведено к минимуму или вообще не потребоваться, что приводит к существенной экономии ресурсов. Этому должна способствовать система приоритетов.

Итак, мы предварительно ознакомились с трехступенчатой системой управления — игрой в шахматы. Теперь перейдем к ее анализу со всей тщательностью.

### ДЕРЕВО ПЕРЕБОРА И МИНИМАКСНАЯ ПРОЦЕДУРА

Метод построения дерева перебора общеизвестен. Вариант перебора продолжается от исходной позиции до конца (несущественно при этом, по какой причине вариант закончен — логически или из-за достижения предельной длины); затем начинается подъем по варианту. В первом же снизу узле варианта перебора необходимо проверить возможность формирования новых ветвей дерева. Если ее нет, то подъем продолжается; если она есть, формируется новая ветвь и идет спуск по дереву до конца нового сформированного варианта, и т. д.

Но мало дерево построить, надо его оценить, т. е. найти оптимальный вариант; делается это с помощью минимаксной процедуры на дереве перебора. В конце каждый вариант оценивается с помощью оценочной функции. При подъеме по варианту к ближайшему узлу дерева «подтягивается» и оценка варианта. Когда в узле собираются все оценки вариантов, примыкающих снизу к этому узлу, сторона, которой принадлежат ходы, отходящие от узла вниз (именно ей принадлежит здесь право выбора варианта по его оценке), выбирает максимальную для себя оценку, которая

---

\* Возможность получения всех траекторий отрицания подтвердилась в эксперименте при решении этюдов.

для другой стороны является минимальной. Вариант с этой оценкой в данном узле является ТОВ. Эта оценка и подтягивается выше — к следующему узлу дерева перебора.

В исходном, наиболее высоком узле дерева перебора ТОВ превращается в оптимальный вариант, по которому и решается вопрос о поиске хода в исходной позиции. (При равенстве оценок нескольких ходов в исходном узле программа, как и человек, бросает жребий.) В этом и состоит выбор хода по минимаксной процедуре на дереве перебора. Когда дерево перебора очень велико, а ресурсы памяти для его запоминания недостаточны, при минимаксной процедуре держат в памяти лишь два варианта, примыкающих к узлу дерева. Оценки этих вариантов сравниваются, вариант с меньшей оценкой стирается в памяти, и лишь тогда идет спуск по новому варианту. При подъеме по новому варианту к данному узлу подтягивается новая оценка, опять идет сравнение двух оценок (новой и старой) и т. д.

Этот метод сравнения двух оценок применяется, по-видимому, во всех играющих программах. Отметим, что этот метод, вынужденный, если дерево перебора велико, предопределяет слабые результаты решения задачи о поиске хода. Человек пользуется иным методом: он формирует небольшое дерево и запоминает его. При этом человек получает возможность найти хороший ход в исходной позиции.

При запоминании поддерева перебора, отходящего вниз от узла, в котором должен быть определен ТОВ, появляется возможность решить вопрос о формировании вариантов (примыкающих к данному узлу) с благоприятными оценками. Если есть надежда, что оценки вариантов (или одного из них) могут быть получены такими, что оценка ТОВ в этом узле может быть изменена в выгодном направлении для той стороны, которая стремится эту оценку изменить, то формируются дополнительные ветви поддерева перебора. Окончательное определение ТОВ и подъем его оценки к следующему узлу производятся лишь тогда, когда надежды на возможное изменение результатов минимаксной процедуры уже нет.

Теперь перейдем к технике формирования МО.

#### ТЕХНИКА ОПРЕДЕЛЕНИЯ ТРАЕКТОРИЙ

Кратчайшая траектория передвижения фигуры от одного поля доски до другого поля определяется с помощью массива  $15 \times 15$ . Было принято, что не кратчайшие траектории образуются не более, чем из двух кратчайших. Для дальнобойных фигур было принято также, что кратчайшая траектория состоит из двух передвижений (трех  $\alpha$ -полей). Эти меры существенно упрощают задачу определения траекторий и, вероятно, без заметного ухудшения качества решения задачи о поиске хода.



Шахматная игра отличается большим разнообразием характера передвижения фигур. Но с помощью массивов  $15 \times 15$  получение этих разнохарактерных траекторий оказывается достижимым. Можно предположить, что шахматный мастер находит траектории с помощью тех же массивов.

Решение задачи о формировании траекторий позволяет решить задачу о формировании первой ступени шахматной системы управления — фигуры, действующей по своей траектории; это оказалось сравнительно легким делом.

Необходимо отметить также, что когда определяется передвижение фигуры от одного поля к другому одна-единственная траектория возникает лишь тогда, если расстояние между полями доски покрывается за одно передвижение. При траектории, состоящей из большего числа передвижений, мы имеем дело с пучком траекторий, ибо фигура от одного поля до другого, как правило, может перемещаться по различным траекториям. Таким образом, когда мы говорим об определении траектории, то, по сути дела, речь идет об определении пучка траекторий.

#### ТЕХНИКА ФОРМИРОВАНИЯ ЗОНЫ

Формирование второй ступени игры в шахматы — уже дело тонкое. Зоны бывают двух типов: 1) когда, например, комлевая фигура с  $\alpha_0$ -поля стремится к конечному  $\alpha_k$ -полю по одной из возможных траекторий пучка комлевых траекторий и 2) когда фигура стремится покинуть  $\alpha_0$ -поле, на котором она расположена, и в одно передвижение переместиться на любое из ближайших  $\alpha_k$ -полей. В первом случае  $\alpha_k$ -поле является единственным, во втором —  $\alpha_k$ -полей множество. К первому случаю относятся зоны нападения, контроля и блокады, ко второму — зоны отступления и деблокады.

Структура зон отступления и деблокады элементарна. Структура зон нападения, контроля и блокады в общем случае достаточно сложна.

Как уже отмечалось ранее, в основу формирования зоны первого типа положен принцип своевременности. Условно принято, что из фигур (+)\* передвигаться может лишь комлевая фигура; другие фигуры (+) могут двигаться лишь при взятии фигур (—) в одно передвижение, т. е. они участвуют в игре, если они уже находятся в засаде. Фигуры (—) включаются в зону и действуют лишь тогда, когда успевают принять участие в игре; это формализуется с помощью переменного горизонта  $H_x$ .

---

\* Фигуры (+) имеют тот же цвет, что комлевая фигура; фигуры (—) имеют противоположный цвет.

Чем больше номер  $\alpha$ -поля комлевой траектории, т. е. чем больше полуходов надо затратить комлевой фигуре, чтобы достичь этого  $\alpha$ -поля, тем больше для этого поля значение  $H_x$  и тем больше полуходов дается на передвижение всем фигурам (—), связанным с борьбой за это  $\alpha$ -поле комлевой траектории. Аппарат, определяющий  $H_x$  и распределяющий это время в полуходах по траекториям отрицания (—), разработан Б. Штильманом.

Б. Штильман предложил также связать формирование зоны с формированием дерева перебора. Это весьма выгодно, ибо тогда зона, в общем случае, формируется не полностью, а лишь частично и до тех пор, пока это вызывается необходимостью; если перебор в частично сформированной зоне следует прекратить, дальнейшее формирование зоны также прекращается.

Задача определения траекторий отрицания (—) достаточно сложна. Эти траектории в пределах горизонта  $H_x$  должны быть определены полностью, без каких-либо исключений, ибо до получения результатов перебора нельзя сказать, какая из этих траекторий бесполезна.

Траектория отрицания определяется по двоянному методу. Когда, например, при игре в зоне какая-либо фигура перемещается на новое  $\alpha$ -поле, то в пределах горизонта  $H_x$  проверяется, может ли другая (другие) фигура напасть на передвинувшуюся фигуру? Если да, то определяется траектория отрицания. С помощью этого метода все же не представляется возможным определить все без исключения траектории отрицания, ибо в процессе перебора фигура, которая должна действовать по еще не определенной траектории отрицания, может сойти с  $\alpha$ -поля возможной траектории отрицания и эта траектория будет упущена.

Чтобы найти все упущенные траектории отрицания, был введен дополнительный метод. Суть его состоит в том, что для определения «пропавших» траекторий отрицания, для которых, в каком-то узле перебора какое-то  $\alpha$ -поле может быть конечным  $\alpha$ -полем, совершается новая процедура подъема по варианту (как бы по уже построенным «рельсам»); при этом фигуры, покинувшие начальные  $\alpha$ -поля возможных траекторий отрицания, будут возвращаться на эти поля и, поскольку горизонт  $H_x$  уже был определен, то вопрос о формировании всех рассмотренных траекторий отрицания может быть решен. Этот особый подъем «по рельсам» происходит до того узла, в котором было принято решение о формировании (включении в игру) данной зоны. Как только принято решение о включении в игру новой траектории отрицания, вступает в силу аппарат псевдоперебора; происходит возврат по варианту до узла дерева, в котором появилась новая траектория отрицания, и перебор продолжается.

При возврате по варианту (в процессе процедуры псевдоперебора) в данный узел дерева перебора (где была включена в игру

траектория отрицания) поддерево, начинающееся от этого узла, не должно быть стерто (стирается лишь одна ветка — рис. 7). Возможность избавления от этой дорогостоящей операции стирания поддерева связана с тем, что в отличие от обычного метода формирования дерева допускается повторение хода в узле — об этом будет сказано далее. Таким образом, пока происходит появление новых траекторий отрицания, действует процедура псевдоперебора, связанная с возвратом по варианту, стиранием одной ветви поддерева (в которой вариант не оценен) и, как следствие, с некоторым расходом ресурсов.

Подводя итог, повторим, что (как это следует из рис. 7) стереть надлежит лишь одну неоцененную ветвь  $nk$ ; все остальные варианты имеют конечную оценку и поэтому должны быть сохранены. Траектории отрицания включаются в игру безоговорочно, но, однако, как это будет объяснено далее, — по приоритету. Можно добавить, что ранее вопрос о стирании ветвей при псевдопереборе решался ошибочно — стиралось все поддерево ниже узла  $m$ . Это было вынужденным решением, поскольку еще не было введено в программу повторение хода в узле.

В 1960 г. в лекции, прочитанной в Университете им. Гумбольдта

в Берлине, автор этих строк интуитивно отметил, что шахматный мастер действует методом последовательных приближений. Процедура псевдоперебора — наглядная иллюстрация этого метода.

Итак, мы в общих чертах ознакомились с техникой формирования зоны — теперь можно двигаться дальше.

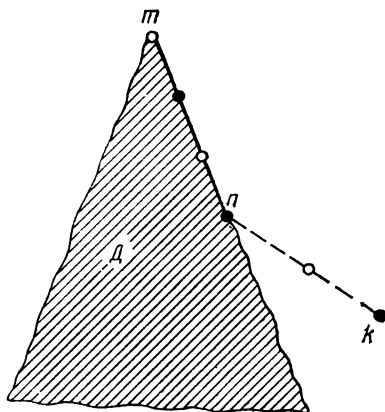


Рис. 7. Суть процедуры псевдоперебора:

$Д$  — сформированное поддерево;  $k$  — узел, в котором обнаружена траектория отрицания;  $n$  — узел, в котором надо стереть ветвь перебора от  $n$  до  $k$  (штриховая линия);  $m$  — узел, в который следует подняться для продолжения перебора (узел включения траектории отрицания)

## ПОЗИЦИОННАЯ СОСТАВЛЯЮЩАЯ ОЦЕНОЧНОЙ ФУНКЦИИ

Позиционная оценка не должна быть усредненной, она должна быть индивидуальной для данной ситуации. Если оценка усредненная, то она может быть справедливой, скажем, для 67% позиций, но ошибочной для 33%, в том числе и для рассматриваемой позиции. Надо отдать должное Капабланке, который, полемизируя с автором книги «Теория середины игры в шахматах» Е. Зноско-Боровским, указал, что основой позиционной оценки является

контроль полей; Капабланка при этом осудил такие отвлеченные позиционные понятия, как «пространство» и «время».

Теперь, когда в основном создан аппарат, формализующий методы игры, используемые шахматным мастером, можно формализовать и понятие «контроль полей».

Контроль полей — это не контроль полей всей доски, ибо имеет ценность контроль лишь тех полей, которые могут быть использованы в *предстоящей* игре. Поэтому следует стремиться к контролю полей тех траекторий, по которым фигуры могут двигаться, но еще не двигались.

В узле дерева перебора, где мы находимся, надо развернуть все пучки траекторий, которые еще не были развернуты, и определить, за какой же стороной контроль большинства полей, составляющих неиспользованные включенные в игру траектории. Это позволит прогнозировать результат игры, результат того перебора, от которого, в частности, пришлось отказаться в конечных узлах вариантов из-за недостатка ресурсов.

Можно принять, что так называемая позиционная жертва не должна превышать двух единиц материала (это будет уточнено в эксперименте). Поэтому, если одна из сторон в узле, где вариант оборван, потеряла больше двух единиц, то, казалось бы, позиционная оценка ничего изменить не может (вариант будет иметь отрицательную оценку). Это ошибочно, ибо воздействие позиционной оценки на минимакс остается.

Далее будет показано, что позиционная оценка позволяет решить вопрос о приоритете включения в перебор (в данном узле дерева) фигур и зон, включенных в игру. Поэтому позиционная оценка (с разворачиванием пучков траекторий) должна производиться в каждом узле дерева перебора. Можно утверждать, что находящиеся под контролем поля определяют полезную подвижность, маневренность фигур. А лучшая маневренность фигур частично и определяет позиционный перевес.

Было принято, что в контроле полей учитываются лишь те фигуры, которые находятся от контролируемого поля на расстоянии одного передвижения (а при блокаде поля фигура уже находится на этом поле). Знак размена фигур (на поле), участвующих в позиционной оценке, и определяет контроль поля той стороной, для которой этот размен благоприятен.

Итак, позиционная оценка подсчитывается в каждом узле дерева перебора. Процедура эта существенно более сложная, чем подсчет оценки по материалу. Все включенные в игру пучки траекторий, по которым игры еще не было (целиком или частично), учитываются при подсчете позиционной оценки. Здесь необходимо подчеркнуть одно различие между оценкой по материалу и позиционной. Оценка по материалу в узле всегда неизменна, она не зависит от того, как к этому узлу мы подошли — сверху или снизу.

Позиционная оценка может в одном и том же узле быть разной, поскольку при спуске или подъеме могут дополнительно включаться в игру зоны (траектории). А раз появилась вновь включенная зона (траектория), появился пучок траекторий, то и позиционная оценка может измениться. Таким образом, позиционная оценка имеет изменчивость, по аналогии близкую к сумме стоимостей целей (мишеней).

Отметим все же, что формирование позиционной оценки гораздо проще формирования зоны, ибо здесь  $H_x$  равно 1 или 0 как для белых, так и для черных. Позиционная оценка на поле траектории учитывается лишь до первого поля (исключительно), в котором оценка будет не положительной — это и определяет подвижность фигуры по траектории.

Все это представляется весьма логичным; поскольку вариант далее не просчитывается, то мы и смотрим, что в первом приближении можно получить на тех траекториях, по которым фигуры еще не успели переместиться!

Итак, основной фактор позиционной оценки пропорционален отношению  $K_w/K_b$ , где  $K_w$  и  $K_b$  — число полей, контролируемых белыми и черными соответственно.

#### ПРИОРИТЕТ ВКЛЮЧЕНИЯ В ПЕРЕБОР ВКЛЮЧЕННЫХ В ИГРУ ФИГУР И ЗОН

Прежде всего надлежит принять решение: включать или не включать зону в игру? Если принято решение включить зону в игру (как это происходит, будет объяснено позднее), то возникает вопрос: в каком порядке включенные в игру зоны (и фигуры) следует включать в перебор? Вопрос этот является важнейшим. От приоритета включения зависит размер дерева перебора. Если выбрать неудачный порядок включения, размер дерева будет больше. Нехватка ресурсов приведет к сокращению предельной длины варианта, т. е. к менее глубокому решению задачи.

Приоритет включения должен зависеть от позиционной оценки, т. е. от контроля полей. Введем два новых понятия.

*Уязвимая мишень* — если в траектории нападения все поля этой траектории, так же как и поля отступления мишени, находятся под контролем атакующей стороны, будем называть такую мишень уязвимой. Размер мишени будем считать равным (по материалу) наименьшему результату размена на каком-либо поле траектории, если этот результат не превышает размера мишени.

Если рассматривается не траектория нападения, а любая другая траектория, то эти понятия также применяются с той разницей, что стоимость мишени будет равна стоимости мишени

в зоне нападения, с которой эта проходима траектория любым образом связана.

*Поражаемая мишень* — то же, что уязвимая мишень (по комлевой траектории нападения), но траектория в одно передвижение и мишень может быть уничтожена при очереди хода за атакующей стороной; в этих условиях мишень может быть поражена. т. е. можно, не совершая взятия, точно прогнозировать его результаты.

Установим следующий приоритет включения траекторий в перебор (в зонах): если стоимость уязвимой мишени такова, что в этой зоне есть смысл играть, то прежде всего следует играть в этой зоне. Если есть другая уязвимая мишень такой же стоимости, то играть надо в той зоне, где цель достигается скорее. При различной стоимости уязвимых мишеней надлежит выбрать зону с большей стоимостью мишени.

Опыт шахматного мастера свидетельствует, что этот путь является оптимальным для получения хорошего решения при минимальной величине дерева перебора.

Понятие поражаемой мишени также может способствовать получению малых размеров дерева. Например, если белые могут поразить мишень такой стоимости, что вариант будет непременно оборван (или если этой стоимости равна разность поражаемых мишеней белых и черных), то даже не следует продолжать вариант, поражая эту мишень, — и так все уже ясно!

Или другой случай: если есть поражаемая и уязвимая мишени одного цвета, то, возможно, следует играть на уязвимую мишень — поражаемая пока подождет.

Не только использование уязвимых и поражаемых мишеней для комлевых траекторий способно сократить дерево перебора; этот же метод можно применить для любых траекторий. Например, после разворачивания пучка траекторий возникает тот же вопрос: какой траектории пучка следует отдать предпочтение? Ответ ясен — той траектории, в которой все поля под контролем, и той из них, которая короче. Если требуется решить вопрос, в каком пучке траекторий надо играть в первую очередь, то следует применить тот же метод.

«Вилочность» траекторий также определяет приоритет включения траекторий в перебор. Когда траектории совпадают в какой-то своей части, это означает, что фигура, которая станет двигаться по совпадающей (вилочной) части траектории, будет перемещаться с тем большей скоростью, чем больше траекторий совпадают. Это также дает приоритет включения, но меньший, чем для рассмотренных случаев.

Та позиционная оценка, которая была подготовлена, — соотношение суммарных чисел контролируемых полей траекторий (до первого неконтролируемого поля) белых и черных, говорит, как

это уже отмечалось ранее, об относительной подвижности фигур, о свободе маневра. Это, конечно, важно, но все же недостаточно для оценки позиции.

Должны быть еще другие составляющие оценки позиций. Такой составляющей, несомненно, должно быть соотношение сумм стоимостей уязвимых мишеней, так как оно также способно прогнозировать возможный успех ниже верхушки усеченного дерева. Но следует учитывать и время, которое необходимо потратить на продвижение атакующих фигур по уязвимым траекториям. Чем это суммарное время меньше, тем лучше. Таким образом, соотношение этих суммарных времен тоже необходимо учитывать. Скорей всего, оно должно учитываться лишь при равенстве (для белых и черных) сумм стоимостей уязвимых мишеней, но это можно будет решить лишь после проведения эксперимента.

Итак, три «классических» фактора позиции учтены в оценке: материал, пространство и время, но это совсем не те факторы, о которых пишут в шахматных учебниках...

#### ВКЛЮЧЕНИЕ ЗОН В ИГРУ

Вопрос о включении зоны в игру следует (как правило) решать в возможно более низком узле поддерева перебора. Это означает, что вариант в процессе перебора просчитывается до конца, а затем, при подъеме по варианту, в первом же узле снизу, где возник вопрос о включении зоны, этот вопрос и должен быть решен.

Таким образом, как правило, ниже узла, в котором решается вопрос о включении зоны, все ранее возникавшие проблемы о включении зон должны быть уже решены, ниже этого узла все должно быть «чисто».

В процессе перебора ходов в пределах горизонта  $H_L$  может возникнуть множество новых «видений», когда фигура одного цвета «видит» фигуру другого цвета (т. е. устанавливается факт существования траектории). Это приводит к образованию пучка комлевых траекторий, после чего неизбежно возникает вопрос: включать эту зону в игру, или не включать?

Не включать зону следует в тех случаях, когда ее включение не связано с надеждой на изменение оценки ТОВ. Можно сформулировать следующее правило. Зоны не следует включать при выполнении одного из двух условий: 1) фигура заведомо не успеет попасть на  $\alpha_k$ -поле своей комлевой траектории и 2) включение зоны в игру заведомо не может изменить оценку ТОВ.

Проверка этих условий производится следующим образом. Примем, что ниже узла, в котором следует решить вопрос о включении зоны, поддерево уже сформировано. Прежде всего следует убедиться в том, что в исследуемом варианте зона еще не вклю-

чалась, — этот вариант и должен быть обследован. Для этого введена процедура «светофор»: ход, относящийся лишь к одной зоне, имеет в дереве метку этой зоны, и при совпадении проверяемой зоны с этой меткой «зажигается» красный свет, а при несовпадении или ходе, относящемся к нескольким зонам, — зеленый.

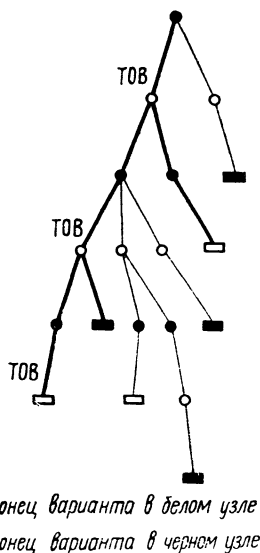


Рис. 8. Какие варианты улучшать? (выделенные утолщенной линией)

Уточним, что поскольку речь идет об улучшении оценки ТОВ, нам для ответа на первый вопрос (успевает ли фигура действовать) не нужно обследовать все варианты поддерева. Если мы обследуем поддерево, например, ниже белого узла и решаем воп-

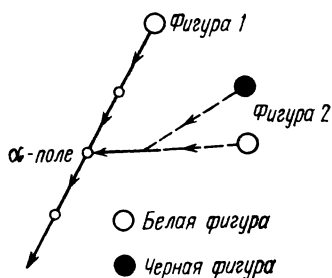


Рис. 9. К вопросу о включении зоны в игру

рос о включении белой зоны, следует учитывать видения и обследовать лишь те варианты, которые в черных узлах совпадают с ТОВ (рис. 8). Другие варианты обследовать не имеет смысла, ибо благоприятное изменение оценок этих вариантов не способно повлиять на результат минимакса в лучшую сторону. Условимся, что обследование этой части поддерева было произведено с положительным результатом — фигура может успеть к  $\alpha_k$ -полю. Теперь мы должны ответить на второй вопрос: способно ли включение зоны изменить итоги минимакса, т.е. оценку ТОВ.

Рассмотрим общий случай, изображенный на рис. 9. Фигура 1 по своей траектории проходит через  $\alpha$ -поле. В переборе эта фигура может пройти  $\alpha$ -поле или на этом поле может произойти размен. Когда фигура 1 попадает на  $\alpha$ -поле, фигура 2 (того же или другого цвета) видит фигуру 1 в пределах горизонта  $H_L$ ; обследование вариантов должно показать, успевает ли фигура 2 попасть на  $\alpha_k$ -поле своей траектории ( $\alpha$ -поле траектории фигу-



ры 1). Этот вопрос возникает, если фигуры 1 и 2 разного цвета; при одинаковом цвете фигур надлежит принять, что фигура 2 поспевает к  $\alpha$ -полю безусловно. Следует ли включать в игру зону с комлевой фигурой 2 (эта зона может быть зоной контроля, блокады или нападения)?

Если есть надежда, что максимальная выгода от включения зоны способна изменить оценку ТОВ, зону надо включать. Как же это определить?

Рассмотрим случай зоны контроля с черной фигурой 2. Возможны две ситуации: 1) когда на  $\alpha$ -поле был размен, в котором участвовала фигура 1, и 2) когда фигура 1 прошла  $\alpha$ -поле и участвовала в размене в дальнейшем. В первом случае происходит включение черной фигуры 2 в размен на  $\alpha$ -поле, что может улучшить результат размена (для черных) на  $\alpha$ -поле. Если выгода от этого улучшения размена в сумме с конечной оценкой варианта, где содержится этот размен, превосходит значение оценки ТОВ, то зону нужно включать в МО, ибо есть надежда на улучшение оценки ТОВ.

Во втором случае максимальная надежда состоит в том, что белая фигура 1 может быть выключена из того размена, который произошел ниже по варианту, после того, как фигура 1 прошла  $\alpha$ -поле. Это может привести к улучшению результата размена в пользу черных. Если эта выгода от изменения результата размена в сумме с конечной оценкой варианта превосходит оценку ТОВ, то и в этом случае есть смысл положительно решать вопрос о включении черной фигуры в МО. Таким же образом решается вопрос о включении в игру зон другого типа — здесь мы рассмотрели включение в МО лишь зоны контроля. Следует иметь в виду, что (после обследования поддерева) таких зон — кандидатов на включение в перебор — может быть множество, и для сокращения дерева перебора необходимо также установление приоритета.

Определение таких зон и установление приоритета их включения в перебор — один из самых тонких вопросов алгоритма.

Таким образом, при видении, которое возникает при спуске по варианту, следует, как правило, достроить поддерево ниже того узла, в котором решается вопрос о включении зоны, — иначе нельзя и решить этот вопрос. Однако в тех случаях, когда нет сомнений в том, что фигура 2 успевает к  $\alpha$ -полю и что новая оценка ТОВ может быть выше полученной (об этом можно судить, сравнивая стоимость новой мишени с суммой прежних), зону следует включать сразу, как только это выяснено.

#### О ПОВТОРЕНИИ ХОДА В УЗЛЕ

Уже отмечалось, что по общепринятым канонам формирования дерева перебора повторение хода в узле лишено смысла: действительно, к чему повторять в узле ход, который уже был рассмотрен

рен? Однако повторение хода в узле лишено смысла тогда, когда  $H_L < 3$ ; в этом случае траектории состоят из одного передвижения. Именно с таким горизонтом действуют все до сих пор известные шахматные программы. Если же  $H_L \geq 3$  и поэтому возможны траектории более чем в одно передвижение, повторение хода в узле становится неизбежным.

Повторение хода приобретает смысл, ибо суть дела не в ходе, а в движении по траектории. Если МО в узле изменилось и появилась новая траектория, у которой первый ход совпадает с ходом по траектории, включенной в МО ранее, и по которой фигура уже двигалась, то повторение хода не только приобретает смысл, но и становится необходимостью. При этом МО будет (в том же узле, где повторяется ход) отличаться от прежнего, а текущая оценка по материалу сохраняется.

#### ГЛУБИНА ПЕРЕБОРА И ВКЛЮЧЕНИЕ ЗОН В ПЕРЕБОР

Прежде всего, условимся, как устанавливается значение предельной глубины варианта  $D_L$ . В отличие от горизонта  $H_L$ , глубина может меняться в партии от хода к ходу. Это связано с тем, что изменение горизонта является весьма грубым воздействием на расход ресурсов, а изменение глубины — тонким (поскольку всегда действует соотношение  $D_L > H_L$ ).

Будем устанавливать глубину перебора по времени, затраченному на обдумывание в предшествующей стадии партии, т. е. по «накопленному» времени, а главное — по времени, затраченному на обдумывание предыдущего хода. В первом приближении будем действовать лишь в зависимости от времени, затраченного на предыдущий ход. Если на предыдущем ходу было затрачено более трех минут, то значение глубины  $D_L$  уменьшаем, если менее — увеличиваем. Значение этого увеличения (уменьшения) глубины будет уточнено в эксперименте.

Итак, глубина  $D_L$  в предстоящем поиске хода установлена. Теперь решим, как значение  $D_L$  будет использовано для отбора зон нападения, включаемых в перебор (из тех зон, которые включены в игру, т. е. в МО); это означает, что остальные зоны должны быть «заморожены».

Примем, что в перебор в данном узле включаются лишь те зоны нападения, мишени в которых могут быть уничтожены (по ком-левым траекториям) за время, оставшееся до достижения предельной длины. Например, если это время равно  $x$  полуходам, а какие-либо две мишени могут быть уничтожены за минимальное суммарное время  $t$  и  $t \leq x$ , то обе зоны с этими двумя мишенями могут быть включены в перебор. Если же  $t > x$ , в перебор может быть включена лишь одна из этих двух зон (или какая-либо другая, удовлетворяющая условию  $t \leq x$ ).

Это вполне логично. Мы стремимся выиграть материал в пределах усеченного дерева; за предельной длиной действует лишь позиционная оценка. Стало быть, в перебор и включаем лишь те зоны, в которых есть надежда, что мишени будут поражены в пределах усеченного дерева. Неплохая аналогия — борзая собака из нескольких зайцев выберет того, который не успеет доскакать до леса; за теми, которые близки к лесу, она не погонится...

Таким образом, когда в переборе будет достигнута глубина  $D_L$ , естественно,  $x$  станет равным нулю и формула  $t \leq x$  определит, что никакая зона далее в переборе анализироваться не должна (собственно говоря, это условие может быть выполнено и ранее — к этому мы еще вернемся). В этом случае, на глубине  $D_L$ , перебор должен продолжаться и далее при наличии уязвимой зоны, а в некоторых случаях, по-видимому (при нападении на короля, например), и при частично уязвимых зонах (когда зона отступления неуязвима или уязвима лишь частично).

Рассмотрим теперь тот случай, когда условие  $t \leq x$  не выполняется, а условная предельная глубина еще не достигнута. В этом случае нет надежд на выигрыш материала (в пределах усеченного дерева) при игре по комлевым траекториям (если зоны не являются уязвимыми — это уже отмечалось); поэтому играть по комлевым траекториям смысла не имеет.

В этом случае следует играть по траекториям отрицания (или комлевым траекториям связанных зон), которые удовлетворяют условию  $t \leq x$ . Хотя кажется, что материал в пределах усеченного дерева (с глубиной  $D_L$ ) все равно не выиграть, это представление может быть ошибочным, если игра по траекториям отрицания (или по комлевым траекториям связанных зон) приведет к тому, что какая-либо комлевая траектория станет уязвимой (с оговоркой, указанной ранее) — тогда ведь перебор может быть продолжен ниже предельной глубины.

Если уязвимых траекторий вообще нет, а для других траекторий условие  $t \leq x$  не выполняется, вариант должен быть оборван до предельной глубины.

По сути дела глубина  $D_L$  сама по себе не является предельной длиной варианта; эта величина лишь косвенно влияет на длину варианта. По-видимому, шахматный мастер так и действует — фиксированной длины варианта в явном виде у него нет...

Теперь можно ответить на естественный вопрос — как быть с обрывом варианта по материалу? Стоимости каких мишеней составляют сумму мишеней ( $\Sigma m$ ), входящих в формулу обрыва? Сомнений быть не должно: стоимости лишь тех мишеней, которые удовлетворяют условию  $t \leq x$  (по комлевым траекториям или по другим, входящим в зону, где идет игра). Это самый большой материал, который можно выиграть; при этом следует обязательно учитывать, что стоимости уязвимых мишеней (а также частич-

но уязвимых) включаются в сумму  $\Sigma m$  дополнительно. Отсюда следует, что обрыв происходит на основании трех факторов — по материалу, времени и контролю полей (уязвимости мишеней). Таким же образом следует уточнить и формулу отсечения ветвей (вариантов).

Теперь осталось еще рассмотреть вопрос о различии между практической партией и этюдом. При поиске хода в позиции из практической партии на значение оптимального варианта ограничений не накладывается. Поэтому соотношение по материалу, содержащееся в исходной позиции, не влияет на поиск хода. Иное дело — в этюде: здесь заранее устанавливается задание (ничья или выигрыш). Поэтому исходное соотношение по материалу необходимо учитывать в этюдах; это соотношение и влияет на стоимости мишеней, которые должны быть атакованы, позволяет исключить из рассмотрения некоторые цели и сократить перебор.

### ОТСЕЧЕНИЕ ВЕТВЕЙ

При подъеме по варианту в данный узел не следует безоговорочно продолжать перебор — формировать в этом узле новые ветви. Необходимо всегда выяснять: а нельзя еще несформированные ветви отсечь от уже сформированных ветвей в данном узле поддерева? Отсечение ветвей отнюдь не следует отождествлять с обрывом варианта. Обрыв варианта в общем случае (рис. 10) производится по формуле

$$-cm_{\tau} \geq \Sigma m_w + \Sigma m_b,$$

где  $\Sigma m_w$  и  $\Sigma m_b$  — сумма стоимостей мишеней белых и черных фигур соответственно;  $c = +1$  в белом узле,  $c = -1$  в черном узле;  $m_{\tau}$  — текущая оценка при спуске по варианту;  $m_{\tau} = M_w - M_b$ , где  $M_w$  и  $M_b$  — сумма стоимостей белых и черных фигур соответственно, стоящих на доске в текущей позиции.

Обрыв варианта происходит независимо от конечной оценки других вариантов, независимо от оценки ТОВ, т. е. от результатов минимакса. Обрыв варианта органически связан с целью игры, со стоимостями мишеней — и только.

Отсечение ветвей имеет иной смысл (хотя один случай отсечения внешне и имеет тот же вид, что обрыв варианта): оно производится лишь для того, чтобы не совершать напрасной работы при процедуре минимакса, т. е. отсечение органически связано с минимаксом. Формула отсечения в белом узле имеет вид

$$m_{\text{ТОВ}} \geq m_{\tau} + \Sigma m_b,$$

а в черном

$$-m_{\text{ТОВ}} \geq m_{\tau} + \Sigma m_w,$$

где  $m_{\text{тов}}$  — поднятая при минимаксе конечная оценка варианта, являющаяся текущей оптимальной. В этой формуле по принципу надежды сравнивается оценка уже достигнутого ТОВ с наибольшей оценкой, которая еще может быть получена. Если новый ТОВ не может превзойти старый — отсечение обязательно.

Отсечение возможно в любой момент формирования ветвей в узле. Поэтому может отсекается первая же ветвь в узле — тогда как бы отсекается весь узел (все ветви в узле) и внешне отсечение нельзя отличить от обрыва варианта. Отсечение может начинаться и с любой ветви в узле.

В левой части формулы отсечения стоит оценка ТОВ. Она должна быть выбрана наибольшей по варианту; таким образом, она может быть связана и с узлом, где происходит отсечение, и с узлом, расположенным выше по варианту.

Поясним все это на примере, изображенном на рис. 10. Из узла  $n$  идет спуск по варианту до узла  $l$ , в котором вариант обрывается (он обрывается по той же формуле, что и в узле  $p$ ). При подъеме в узел  $k$  выясняется, что по МО можно построить в этом узле другую ветвь. Происходит обращение к формуле отсечения; знак неравенства обратный, значит, ветвь строить надо. Спускаемся в узел  $p$ , где вариант обрывается, и поднимаемся в узел  $q$ ; и здесь формула отсечения дает отрицательный результат. Тогда спускаемся в узел  $r$ , в котором формула отсечения определяет прекращение формирования ветвей...

Остается определить лишь член  $\Sigma m$  в правой части формулы отсечения. Это не совсем простое дело. По-видимому, сюда могут входить стоимости мишеней (как из включенных, так и не включенных в игру, ибо включение возможно ниже по ветви) и поражаемых целей. Однако окончательный ответ на вопрос может быть получен лишь в эксперименте. Между прочим, то же можно сказать и о формуле обрыва. В правой части неравенства в  $\Sigma m_w + \Sigma m_b$  не должны быть все стоимости белых и черных мишеней, но и на этот вопрос ответ также должен быть получен в процессе эксперимента.

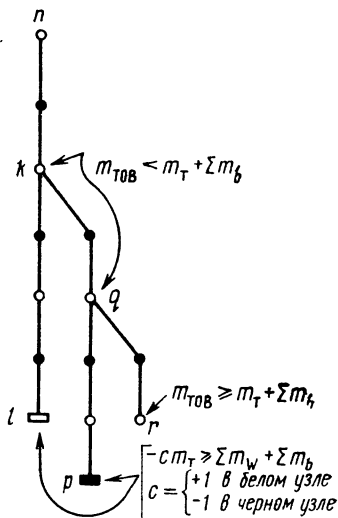


Рис. 10. К вопросу отсечения ветвей

В процессе перебора МО меняется. Появляются новые зоны игры, МО растет, растет объем перерабатываемой информации и это вызывает существенное замедление работы программы. Поэтому в МО зона должна включаться лишь тогда, когда она нужна, и выключаться, когда необходимости в ней уже нет. С этой точки зрения, зона в процессе своего «существования» может претерпевать следующие состояния.

1. Зона включена в МО. Она участвует во всех делах МО — в позиционной оценке, в определении приоритета, вилочности и пр.

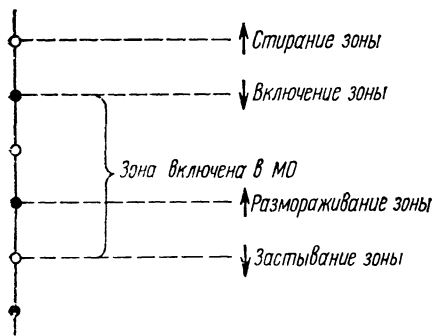


Рис. 11. Фазы состояния зоны:

↑ — подъем по варианту; ↓ — спуск по варианту

Информация о включенной в МО зоне собирается и перерабатывается.

2. Зона выключена из МО при спуске по варианту. Это может произойти по следующим причинам:  $\alpha_0$ -фигура уничтожена или сошла с  $\alpha_0$ -поля (не занимая  $\alpha_1$ -поле), то же случилось с  $\alpha_k$ -фигурой (или с  $\alpha_k$ -полем, если речь идет не о комлевой траектории нападения). В этих случаях было принято, что зона должна быть заморожена и вся информация о зоне должна быть отделена от информации, непосредственно связанной с МО.

Следует иметь в виду, что эта замороженная информация, не нуждающаяся в переработке в данный момент, при подъеме по варианту (в узле, который расположен непосредственно выше над узлом, где зона была заморожена) вновь должна быть разморожена в нужный момент и включена в МО.

3. При дальнейшем подъеме по варианту, когда мы миновали тот узел, в котором при спуске зона была впервые найдена, эта зона должна быть стерта в памяти ЭВМ, в общем случае зона уже не понадобится...

Таким образом, в МО находятся лишь те зоны, которые описаны в случае 1 (рис. 11). Это приводит к резкому сокращению МО; МО способно меняться от узла к узлу как при спуске, так и при подъеме. Подобное МО было названо «действующим». Следует указать, что все сказанное о замороженных зонах относится и к застывшим траекториям отрицания — они также должны храниться там же, где и подобные им зоны. С траекториями дело, однако, обстоит несколько сложнее: уже застывшие траектории

отрицания еще включенной зоны должны обследоваться каждый раз, когда решается вопрос о формировании новой траектории отрицания с тем, чтобы уже сформированная, но застывшая траектория отрицания не была сформирована вновь. Итак, запоминанию подлежат зоны, соответствующие случаям 1 и 2, но анализируется информация, лишь связанная со случаем 1.

Это действующее МО непрерывно меняется от узла к узлу в процессе перебора. Именно это МО используется при определении позиционной оценки и приоритетов включения траекторий в перебор. Особо важную роль играет действующее МО в узлах, где вариант заканчивается принудительно. К этому мы сейчас и перейдем.

### ЗАПОМИНАНИЕ МО В КОНЕЧНОМ УЗЛЕ

Когда в результате минимакса определяется оптимальный вариант, по которому и совершается ход в исходной позиции, то если и ответ противника соответствует тому же оптимальному варианту, не следует возобновлять перебор с новой исходной позиции. Если мы снова начнем перебор с вершины сохранившегося поддерева (другая, уже ненужная часть дерева стерта в памяти ЭВМ), то это будет неэкономным решением.

Прежде всего убедимся в том, что прежний оптимальный вариант (или варианты) остался оптимальным и в новой позиции — на это надежда есть, поскольку противник также сыграл по этому варианту. Для этого следует спуститься в конечный узел оптимального варианта и возобновить перебор в сохранившемся поддереве с этого узла.

Чтобы «оттолкнуться» в переборе именно от этого узла, нам надо знать МО, которое ранее было в этом узле. Очевидно, заранее надлежало сохранять в памяти математическое отображение в конечных узлах ТОВ. В процессе формирования дерева перебора должны сохраняться в памяти ЭВМ те МО, которые соответствуют конечным узлам оптимальных вариантов (если их несколько) или оптимальному варианту (если он один).

### ТЕХНИЧЕСКИЙ ВОПРОС

Рассмотрим один из технических вопросов, который также может влиять на формирование малого дерева: когда в варианте повторяется позиция, которая ранее уже была оценена в *другом* варианте, то возможно вариант этот продолжать не имеет смысла, ибо оценка его уже известна. Необходимо, однако, убедиться в том, что не только позиции, но и МО в этих узлах совпадают; определить это можно косвенным путем — по совпадению приоритетов очередности включения ходов в перебор.

Когда позиция повторяется в том же варианте, то по правилам игры его также продолжать нечего — он должен быть признан ничейным. Определить повторение позиции можно при формировании библиотеки текущих позиций. Эта библиотека невелика, ибо число узлов в дереве перебора мало.

## Б. ИСПОЛЬЗОВАНИЕ ОПЫТА ПРОШЛОГО

Когда шахматный мастер играет партию, он использует опыт прошлого (как чужой, так и собственный) четырьмя различными способами:

1. По способу «попугая» — это характерно для игры в начале партии. Дебютная теория в какой-то своей части не подлежит обсуждению, и мастер делает эти ходы в дебюте, не вдаваясь в суть дела, т. е. он действует так, как говорит попугай.

2. По справочному методу — играя партию, мастер ищет в своей библиотеке накопленных знаний точно такую же позицию, как в дереве перебора. Эта позиция из библиотеки имеет оценку; при совпадении позиций можно тут же оборвать вариант, так как оценка варианта перебора становится известной. Этот способ использования прошлого опыта характерен для эндшпиля, но может дать успех и в дебюте (при перестановке ходов).

3. По методу, основанному на стремлении к позициям, используемым при справочном методе — в этом случае мастер ищет в библиотеке позиции с благоприятной оценкой, *близкие* \* к позиции в переборе. Найдя такие позиции, мастер так формирует перебор, чтобы по возможности получать эти позиции в дереве перебора. Тогда вступает в действие способ 2. Таким образом, этот метод также характерен для эндшпиля и, возможно, для дебюта.

4. По ассоциативному методу, основанному на частичном сходстве позиции из дерева перебора с «позициями» в библиотеке; мастер ищет в библиотеке «фрагмент» позиции, т. е. небольшую группу фигур, действие которых в прошлом принесло успех. Если в позиции из дерева перебора в наличии та же группа фигур, что содержится во фрагменте (так же расположенных), то мастер включает в перебор в первую очередь фигуры из этого фрагмента, чтобы проверить, не может ли это и на сей раз принести успех. Если фрагмент в прошлом много раз приносил успех, то, вероятно, что он принесет успех снова. Этот метод определяет направление перебора, и в среднем он дает экономию ресурсов при формировании дерева перебора. Ассоциативный метод, по-видимому, является единственным способом использования опыта прошлого в середине игры и сложном эндшпиле.

---

\* См. словарь терминов (с. 143).



## БИБЛИОТЕКА ДЕБЮТОВ

У «Пионера» (так была названа шахматная программа для ЭВМ, создаваемая по данному алгоритму) библиотека дебютов невелика, и характерный для использования этой библиотеки метод «попугая», который применяется во всех шахматных программах, где имеется библиотека дебютов, к сути задачи отношение имеет слабое.

Длина дебютных вариантов ограничена 12 полуходами. Предполагается, что после этого начнется формирование дерева перебора. Список наименований дебютов сведен также к минимуму.

Предусмотрено пополнение библиотеки дебютов; в принципе возможно и самообучение «Пионера» — он сам будет пополнять библиотеку на основании опыта, полученного в сыгранных им партиях, а возможно — и чужих партий.

## БИБЛИОТЕКА МИТТЕЛЬШПИЛЯ

Ассоциативный метод, несомненно, самое сложное из того, что связано с использованием накопленных в прошлом знаний. Фрагмент занести в библиотеку надо так, чтобы это дало направление перебора. С этой целью фигуры во фрагменте разделены на две группы: фиксированные, которые должны уже стоять на определенных полях доски, и привязанные к определенным полям, т. е. расположенные на некотором расстоянии от этих полей. Фиксированные фигуры не намечены к участию в переборе — сторона, использующая фрагмент, в этом не заинтересована. Привязанные фигуры должны стремиться к тем полям, к которым они привязаны.

Одним фрагментом можно записать некоторое их множество, если этот фрагмент в разумных пределах перемещать по доске. При формировании библиотеки это должно быть учтено. Фрагмент определяет (если группа фигур в позиции из дерева перебора «совпадает» с позицией фигур во фрагменте) приоритет включения привязанных фигур в перебор, т. е. дает направление перебора.

Следует иметь в виду, что сложный эндшпиль, с точки зрения использования ассоциативного метода, ничем не отличается от миттельшпиля. Но для этого библиотека эндшпиля также должна быть записана во «фрагментной» форме. Пока это не сделано.

Библиотека миттельшпиля еще невелика — она насчитывает около 70 фрагментов, а с учетом перемещения фрагментов по доске — 630 позиций. И здесь, в принципе, возможно самопополнение библиотеки, т. е. использование принципа самообучения.

## БИБЛИОТЕКА ЭНДШПИЛЯ

Заполнение этой библиотеки с шахматной точки зрения — дело несложное, ибо теория эндшпиля в отличие от теории миттельшпиля хранит великое множество позиций. Но с точки зрения составления программы — сложное.

Для упрощения задачи было принято, что при использовании библиотекой эндшпиля будут применяться только второй и третий методы. Собственно говоря, справочный метод можно рассматривать как частный и наиболее простой случай, вытекающий из метода стремления.

Обычно — в представлении шахматистов — теория эндшпиля связывается с набором вариантов. Для программирования это вряд ли может быть использовано, да этим не пользуются и шахматные мастера. Было принято, что библиотека включает в себя позиции; при этом отмечается очередь хода, оценка (выигрыш, проигрыш, ничья) и, когда это необходимо, рекомендуемый ход.

Для компактности записи в библиотеку включался «силуэт» позиции, в которой белый король привязывался к какому-либо полю доски. Были введены в программу формулы, позволяющие высчитать оценку и рекомендуемый ход при изменении координат положения белого короля на доске в допустимых пределах — при этом взаимные координаты фигур в позиции остаются неизменными. Можно было и сразу записать все необходимые позиции в память ЭВМ; в этом случае формулы были бы не нужны, но был бы занят больший объем памяти.

Было записано в память около 700 силуэтов, которые преобразуются примерно в 7000 позиций. С учетом очереди хода — это уже 14000! Автор этих строк знает не более половины этих позиций. Однако если учесть вертикальную и горизонтальную симметрию, а также диагональные (при отсутствии пешек), то получается примерно 35000 позиций...

Эти позиции принадлежат к так называемому «техническому» эндшпилю, знание которого составляет неотъемлемую часть шахматной техники мастера. Условно было ограничено число фигур (8), позиции были разбиты на классы по соотношению материала (всего 31). После того, как библиотека была составлена, она была дана на проверку (по частям) десяткам кандидатов в мастера. Примерно в каждой десятой позиции был обнаружен дефект. После этого можно было считать, что число ошибочных позиций вряд ли существенно превышает 1%, что представляет удовлетворительным.

На стремление к позициям также были наложены ограничения: 1) фигуры в позиции из дерева перебора должны по наименованию совпадать с фигурами из позиции в библиотеке, 2) координаты могут отличаться только у двух фигур и 3) общее число передвижений у этих двух фигур (для того, чтобы позиции совпали) не должно превышать заданного числа.

Сторона, стремящаяся к позиции, для своих фигур формирует пучки комлевых траекторий (ростки зон) и, если нужно, зону: траектории чужих фигур принудительно формировать нельзя — они могут иметь место только в том случае, если они уже сфор-

мированы по другим причинам. Если это так, то, в принципе, стремление возможно. При этом, для неприятельских фигур эти траектории будут антивилочными в том смысле, что неприятельская сторона должна действовать по этим траекториям лишь в крайних случаях.

Библиотека эндшпиля должна содержать не только позиции, но и правила, например, правило квадрата. Однако установить, какие правила должны быть включены, можно лишь на основании длительного эксперимента.

Пополнение библиотеки позиций эндшпиля путем самообучения, несомненно, является напрашивающейся и перспективной задачей. Получив результаты в этом направлении, можно будет обратиться и к другим, более тонким сторонам проблемы самообучения. Работа в отношении самополнения библиотеки позиций уже начата.

Мастер, когда играет в шахматы, непрерывно обращается к опыту прошлого как в дебюте, так и в середине игры, и эндшпиле. Хотя он делает это, по-видимому, не в каждом узле дерева перебора, но все же из этого следует, что число узлов должно быть невелико, а программа пользования библиотекой должна быть организована так, чтобы это требовало малого расхода ресурсов ЭВМ.

Рассмотрим вопросы о приоритете, связанные с библиотеками миттельшпиля и эндшпиля. Очевидно, наличие библиотеки (если отвлечься от случая полного совпадения позиции из перебора с позицией из библиотеки эндшпиля, когда вариант обрывается) может привести к изменению принятого порядка приоритетов, а также к появлению в МО новых траекторий.

Действительно, наши приоритеты приняты на основе усредненного опыта прошлого — когда они применяются, в среднем дерево перебора будет наименьшим. Но бывают и исключения из правил: если в прошлом встречалась позиция, у которой дерево перебора можно существенно сократить, нарушая правила приоритетов, то такая позиция (фрагмент) должна быть включена в библиотеку (на этом должно основываться самополнение и простое пополнение библиотеки). При использовании этой библиотечной позиции (фрагмента) должен быть изменен порядок приоритетов подобно тому, как он должен был быть изменен в прошлом — все незатронутые этими изменениями приоритеты сохраняются. Каждая позиция (фрагмент) должна иметь оценку по стоимости мишени, чтобы можно было использовать эту информацию при определении приоритета.

При сравнении стоимостей мишеней (в том числе с библиотечными стоимостями) следует учитывать еще условную стоимость, которую назовем результирующей. Например, можно принять, что при позиционном равновесии различие по материалу в две

(или три) единицы уже определяет результат партии. Поэтому, если можно играть по уязвимой траектории нападения на мишень, меньшую, чем содержится в библиотеке, но такую, что может быть выиграна результирующая стоимость, то приоритет включения в перебор «по библиотеке» отодвигается.

## 5. ТРИ ЭТЮДА (ЭКСПЕРИМЕНТ)

Еще до того, как была начата работа над программой, мне довелось предсказать, что первая ее проверка будет на этюдах. Решение этюда программой — весьма выигрышный и в то же время не столь уж сложный для данной программы эксперимент. Одной из основ шахматной силы мастера является счет вариантов; после того, как эта способность мастера формализована в виде программы, напрашивается эксперимент на этюдах.

Было подготовлено 11 композиций, но оказалось, что для получения необходимых сведений достаточно решить лишь 3 этюда...

Предполагалось, что этюды будут легкой добычей «Пионера», ибо этюды лишены позиционных тонкостей и все варианты досчитываются до конца. Учитывалось также, что не потребуется библиотек дебютов, миттельшпиля и эндшпиля. В действительности все оказалось гораздо сложнее.

На первом же этюде Р. Рети (рис. 12) «Пионер» спотыкался. Дерево перебора разрасталось, варианты не обрывались, «Пионер» не знал, какой из ходов имеет приоритет. Между тем, шахматный мастер знает, когда надо оборвать вариант и какому ходу следует предоставить приоритет во включении в перебор — тогда дерево и оказывается небольшим!

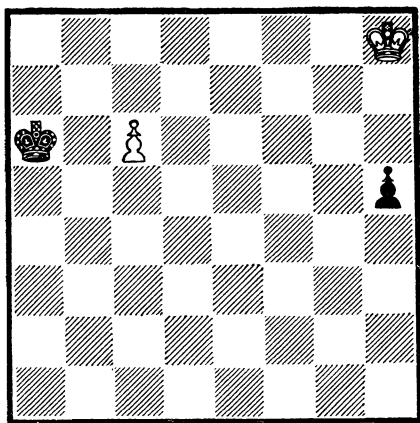


Рис. 12. Этюд Р. Рети. Ничья

Пришлось ввести в программу правила — в основном, они базировались на известном правиле квадрата. Тогда дерево перебора оказалось состоящим всего из 54 узлов. Если бы была библиотека текущих позиций, то тогда удалось бы избежать повторения вариантов и число узлов сократилось бы до 45 (рис. 13).

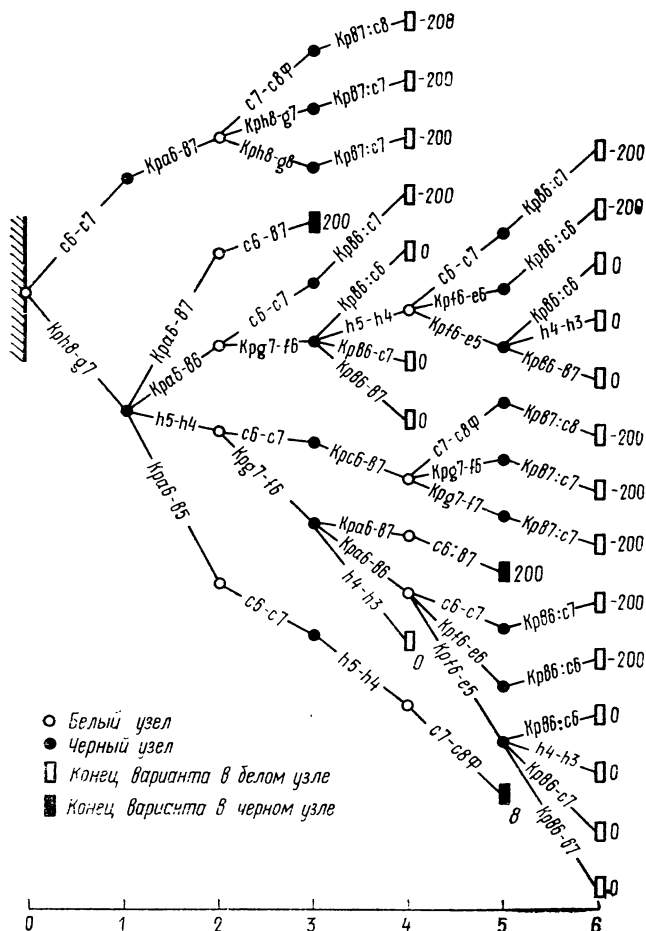


Рис. 13. Дерево перебора ходов в этюде Р. Рети

Некоторые варианты кажутся незавершенными, но в действительности они доведены до логического конца — они оценены в соответствии с включенными в программу модификациями правила квадрата.

Нелегкой задачей для «Пионера» оказался и этюд М. Ботвинника и С. Каминара\* (рис. 14), несмотря на то, что этюд весьма

\* С этим этюдом произошла забавная история. В 1925 г., когда мы с Сереей Каминером составляли этюд, я считал, что на поле g6 стоять должна пешка, а Сережа настаивал, чтобы предпочтение было отдано слону... Сережа меня уговорил, и этюд был опубликован в его редакции. В январе 1977 г., вспоминая эту композицию, я по ошибке поставил на поле g6 пешку, и «Пионер» решил этюд в первоначальном варианте...

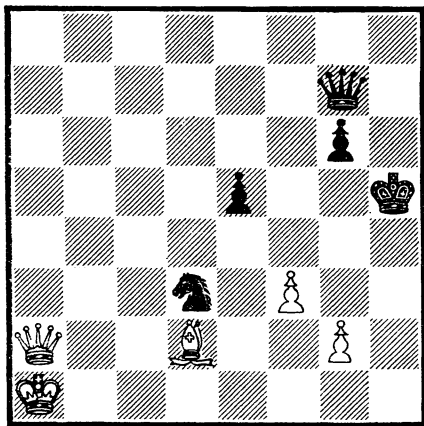


Рис. 14. Этюд М. Ботвинника и С. Каминера. Выигрыш

прост, он поистине детский — когда этюд был составлен, мне было 13, а Каминеру — 14 лет. Вновь выяснилось, что если без разбора включать в перебор ходы по траекториям МО, то дерево перебора «Пионера» существенно отличается от дерева перебора шахматного мастера. Пришлось запретить игру в зонах, где стоимость мишени меньше потерянного материала, пришлось ввести еще и некоторые правила, при выполнении которых вариант обрывался. По-видимому, эти затруднения были вызваны отсутствием позиционной оценки. Дерево перебора в конеч-

ном итоге состояло из 145 узлов (рис. 15).

Казалось, следовало бы прекратить дальнейшее решение до тех пор, пока в программу не будет введена позиционная оценка. Но это решение не было принято; суть дела в том, что в этюде Рети так мало фигур и они столь малоподвижны, что вполне можно было допустить, что этюд окажется под силу сильнейшим программам, основанным на методе полного перебора. В этюде Ботвинника и Каминера фигур больше, среди них есть и дальнобойные, но так много взятий и шахов (программы, основанные на полном переборе, продлевают варианты в случае взятий и шахов за пределы стандартной длины варианта), что и данный этюд мог быть решен этими программами. Поэтому была предпринята попытка решить этюд Г. Надареишвили, ибо не было сомнений как в сложности этой композиции, так и в том, что другие программы осилить его не смогут (рис. 17).

Основной вариант, найденный «Пионером»: 1. g6 Kpf6 2. g7 Ch7 3. e4!! Kf3 4. e5+K:e5 5. Kp:h7 Kf7 6. g8Ф Kg5+7. Ф:g5+ Kp:g5 8. h6 c4 9. Kpg7 c3 10. h7 c2 11. h8Ф c1Ф 12. Фh6+ Kpf5 13. Ф:c1.

Сначала решение требовало столь большой затраты машинного времени, что практически исключалась возможность получения решения. Тогда Б. Штильман был вынужден действовать, как аэронавт, воздушный шар которого угрожающе теряет высоту, — пришлось выкидывать не только балласт, но увы — и полезный груз. Так Б. Штильман отключил подпрограмму деблокады траекторий (оставив только деблокаду траектории пешки), и поэтому придиричивый читатель не найдет одного авторского вариан-

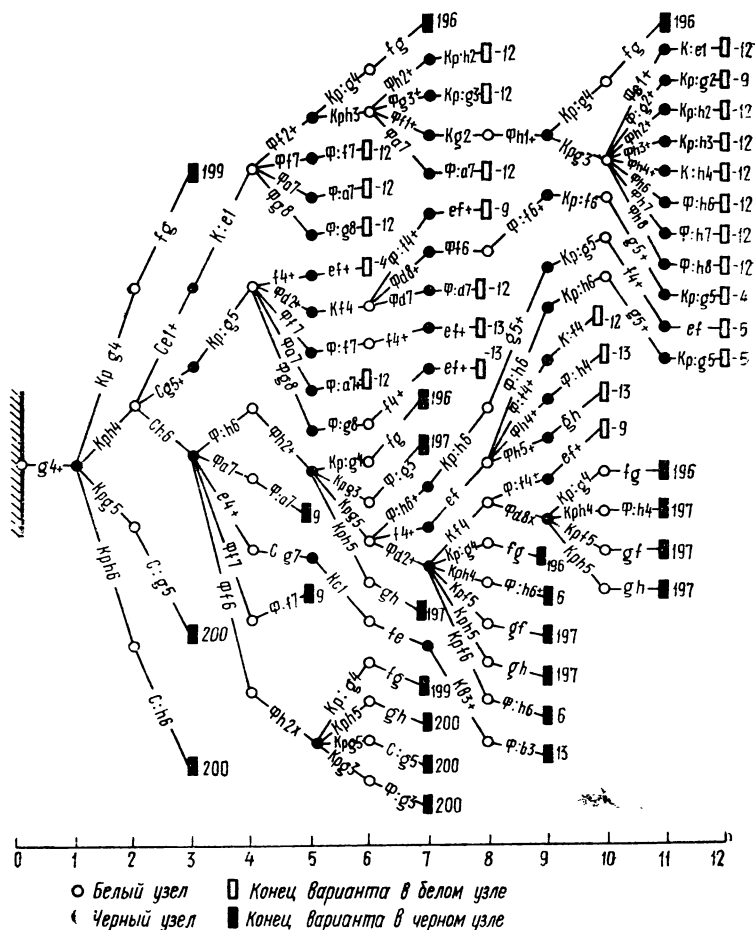


Рис. 15. Дерево перебора ходов в этюде М. Ботвинника и С. Каминара

та решения. Но ускорить решение удалось настолько, что время решения стало менее 4 часов — для игры в силу мастера «Пионеры» нужны ЭВМ с большим быстродействием!

Программу, конечно, не удалось «почистить» основательно — технических ошибок осталось немало — устранить их можно путем неоднократных экспериментов, для чего также нужно большое быстродействие... Поэтому в дерево перебора затесался и один весьма странный ход (Kd7 — g7). Были и другие ошибки... Но и этот черновой вариант дерева перебора (рис. 16) говорит о многом, ведь в дереве всего 200 узлов.

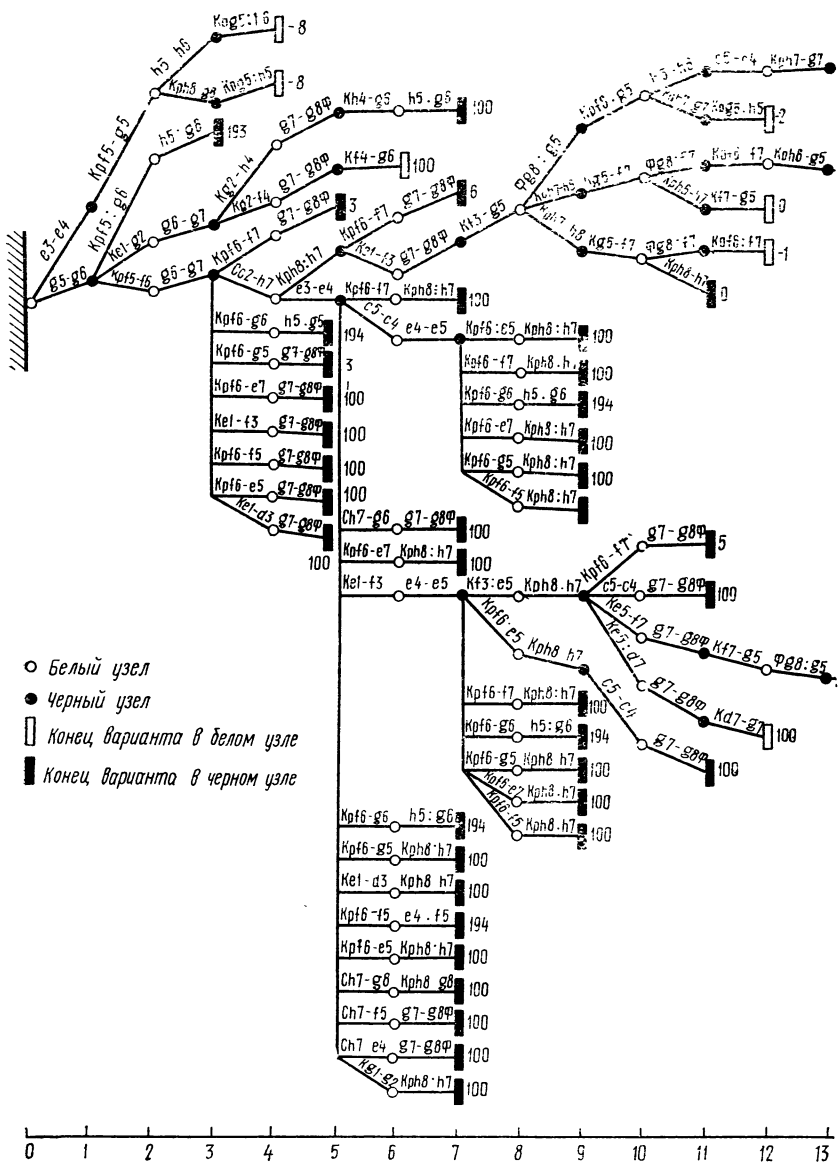


Рис. 16. Дерево перебора





В процессе отладки также пришлось вводить некоторые позиционные паллиативные правила, но на основании этого этюда были сделаны весьма важные выводы. Собственно, лишь после этого и удалось решить вопрос о приоритете включения зон в перебор.

«Логика» этюда Надареишвили, в основном, состоит в том, что черный король, когда он стоит на поле f6, контролирует поля отступления белого короля — это (и только это!) дает черным контригру. Вряд ли можно составить «человеческое» дерево, если этого не «понять». Тонкости игры белых построены, отчасти, на том, чтобы отогнать черного короля от поля f6. Рассмотренный ранее анализ части поддерева (с включением соответствующих вариантов) может в этом помочь! Сравнивая результаты анализа двух частей поддерева, можно увидеть, какую роль в оценке ТОВ играет позиция черного короля на f6. Формально это и будет «понимание».

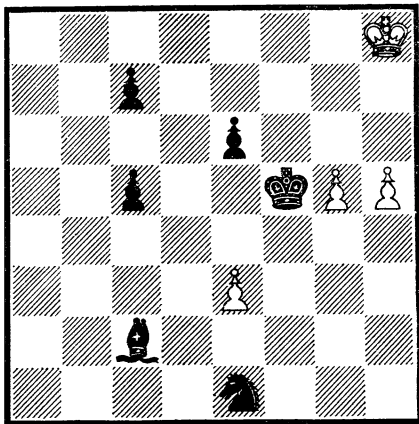


Рис. 17. Этюд Г. Надареишвили. Выигрыш

Можно отметить, в чем состоит главная «хитрость» этюда Надареишвили для шахматиста-практика. Мастер всегда ищет уязвимые и поражаемые мишени;

на третьем ходу такой поражаемой мишенью является слон h7. После того, как уничтожается слон h7, уязвимой мишенью становится превращение пешки g7 в ферзя... Нападение же пешки e3 на короля f6 контролируется черными на поле e5, да и черный король может отступать. Вот в этом и состоит психологический барьер, который нужно преодолеть, чтобы успешно найти решение — надо играть не в зонах с уязвимой и поражаемой мишенями, а в другой зоне!

Теперь, когда программа, в основном, закончена, дело за дальнейшими экспериментами.

## 6. ВТОРОЙ ЧЕМПИОНАТ МИРА

Первый чемпионат мира среди шахматных программ состоялся в 1974 г. в Стокгольме под эгидой конгресса ИФИП (Международной федерации по обработке информации); тогда победила советская программа «Каисса».

История шахматных турниров среди ЭВМ начинается с 1970 г., когда в Нью-Йорке был проведен первый чемпионат США, приуроченный к ежегодной конференции Эй-Си-Эм (Ассоциация вычислительной техники). С тех пор чемпионаты проводятся ежегодно.

Всемирные же чемпионаты среди ЭВМ проводятся один раз в три года (так же, как и среди людей!). Очередной, 2-ой чемпионат происходил в Торонто (Канада) в августе 1977 г.

Эти турниры требуют немалых финансовых затрат — на машинное время и каналы связи. Поэтому они обычно проводятся по швейцарской системе, в 4 тура. Контроль времени — 20 ходов в час.

Программа «Каисса» (авторы Г. Адельсон-Вельский, В. Арлазаров и М. Донской) и в первом и во втором чемпионатах основывалась на полном переборе. Ее главный конкурент — программа «Чесс 4.6» (США) — в 1974 г. пытался уклониться от полного перебора и потерпел неудачу. В 1977 г. американцы (Д. Слейт и Л. Аткин) вернулись к полному перебору и на сей раз стали чемпионами.

Обе эти программы, пожалуй, выделяются среди других (участников всего было 16). Они отличаются в деталях, но в основе — подобны. Обе считают все варианты (с учетом так называемого метода ветвей и границ, или  $\alpha - \beta$ -отсечения) до предельной длины; варианты продлеваются дальше, если есть взятия или шахи.

Показательная партия, сыгранная после чемпионата, дает ясное представление об игре сильнейших программ.

### *Каисса — Чесс 4.6*

1. e2 — e4 Kb8 — c6

Это малоизвестное начало, носящее имя А. Нимцовича, было запланировано черными, несомненно, для того, чтобы избежать неприятностей, возможных из-за дебютной библиотеки Каиссы.

2. Kg1 — f3...

Белые, очевидно, не были готовы к этому дебюту — теоретический ход здесь 2. d4.

2... e7 — e6

Черные продолжают избегать известных продолжений, которые возникали после 2...e5.

3. d2 — d4 d7 — d5 4. Cf1 — d3...

Ход, который вряд ли бы сделал квалифицированный шахматист. Продолжая 4...Kb4, черные могли теперь разменять слона d3, после чего дебютные трудности для них были бы позади. После 4. e5 черным было бы нелегко.

4... d5 : e4 5. Cd3 : e4 Cc8 — d7 6. 0—0...

Шаблонное развитие. Напрашивалось 6. Кe5 К : e5 7. de и позиционный перевес белых очевиден.

6... Kg8 — f6 7. Лf1 — e1 ...

Таким путем белые сохраняют более свободную игру, но, расставаясь с королевским слоном, снижают свои активные возможности. Напрашивалось 7. Cd3.

7... Кf6 : e4 8. Le1 : e4 Cf8 — e7 9. c2 — c4 f7 — f5

Вряд ли мотивированное ослабление полей по линии e и диагонали a2 — g8.

10. Le4 — e1 0 — 0 11. Kb1 — c3 f5 — f4

Этот ход производит крайне неблагоприятное впечатление. Теперь хотя бы простое 12. Ke2 g5 13. d5 ставило черных из-за ослабления позиции рокировки в критическое положение.

Достаточно было и 12. Ke4.

12. Фd1 — d3 Фd8 — e8 13. g2 — g3 ...

Если «Каисса», кроме небольших приключений в дебюте, до сих пор держалась с достоинством, то этот ход рассеивает все иллюзии. Инициатива черных по линии f принимает угрожающий характер; ослабление поля f3 особенно чувствительно. После 13. d5 задача черных была бы нелегкой из-за неудачного положения черного ферзя на поле e8.

13... f4 : g3 14. h2 : g3 Фе8 — f7 15. Cc1 — f4 g7 — g5

Это выглядит весьма опасным, ибо после 16. К : g5 С : g5 17. С : g5 Ф : f2+ 18. Kph1 К : d4 неизбежно 19... Cc6.

16. d4 — d5 ...

(см. диаграмму на с. 67).

16... e6 : d5

После 16 ... Kb4 17. Фе4 gf 18. de С : e6 19. Ф : e6 fg (или 19 ... Кc2 20. Ф : e7 Ф : e7 21. Л : e7 К : a1) черные добивались материального перевеса.

17. Кc3 : d5 g5 : f4 18. Kd5 : e7+ Кc6 : e7 19. Фd3 : d7 Ke7 — g6 20. Фd7 : f7+ Лf8 : f7

Осложнения кончились и наступил спокойный эндшпиль. До этого шла равная борьба, что заслуживает быть отмеченным особо: дело в том, что «Каисса» использовала ЭВМ с быстродействием  $3 \cdot 10^6$ , а «Чесс 4.6» —  $12 \cdot 10^6$  операций в секунду. Это позволяло «Каиссе» считать варианты со стандартной длиной 5 полуходов, а «Чесс 4.6» — 6 полуходов; минимальное различие, которое оказалось несущественным. Но с разменом фигур уменьшалось число возможных ходов в позиции, а стало быть, и в дереве перебора — значит можно увеличивать длину варианта («Каисса» поддерживала число узлов в дереве, равным 90 000, а «Чесс 4.6» — 400 000). В конечном итоге «Каисса» удлинила варианты до 9 полуходов, а «Чесс 4.6» — до 12! В эндшпиле

«Чесс 4.6» был сильнее ... «Тактическая» ошибка «Каиссы» состояла в том, что она не уклонялась от разменов!

21. g3—g4 ...

Хороший ход. Как объяснили авторы «Каиссы», программа стремилась сохранить у черных разобщенные пешки f и h.

21 ... Лf7—d7 22. Ла1—d1 Ла8—d8 23. Лd1 : d7 ...

Напрасно. После 23. Ле8+ Л : e8 24. Л : d7 Ле7 25. Л : e7 К : e7 26. Кpf1 с последующим Кpf1—e2—d3—e4 белые получали очевидный перевес в коневом окончании из-за слабости пешки f4.

23 ... Лd8 : d7 24. Кpg1—g2 Кpg8—g7 25. Кf3—g5 ...

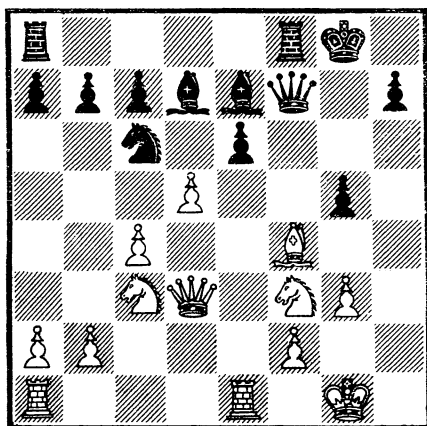


Рис. 18. Познция из партии Каисса — Чесс 4.6

Ниже всякой критики — нельзя было пропускать черную ладью на вторую горизонталь.

25 ... Лd7—d2 26. Лc1—b1 ...

Может быть, плохо было и 26. Ке6+ Кpf6 27. К : c7 Л : b2 и фигуры белых разобщены.

26 ... Лd2—c2 27. b2—b3 ...

Сомнительно и 27. Ке6+ Кpf6 28. К : c7 Л : c4.

27 ... Кg6—e5 28. Лb1—h1 Лc2 : a2 29. Лh1—h4 ...

Эндшпиль безнадежен, и после 29. Л : h7+Кpg6 30. Ле7 Кс6 белые без фигуры.

29 ... Ке5—d3 30. Кg5—h3 Ла2—b2 31. g4—g5 Кpg7—g8

Проще было 31 ... a5.

32. Kh3 : f4 ...

Ошибочная оценка пешечного конца. Последний шанс заключался в продолжении 32. Kpf3 Л:b3 33. К:f4 Ке5+ 34. Кре4 К:c4 35. Кd5. Остальное очевидно.

32 ... Лb2:f2+ 33. Kpg2—g3 Лf2:f4 34. Лh4:f4 Кd3:f4 35. Kpg3:f4 Kpg8—f7 36. b3—b4 Kpf7—e6 37. Kpf4—e4 a7—a6 38. Кре4—f4 Кре6—d6 39. Kpf4—e4 c7—c5 40. b4:c5+ Kpd6:c5 41. Кре4—d3 a6—a5 42. Kpd3—c3 a5—a4 43. Крс3—d3 Крс5—b4 44. Kpd3—c2 Kpd4:c4 и белые сдались.

Из других программ — участников чемпионата — заслуживает внимания «Острич» (Страус), автором которой является М. Ньюборн. «Острич» был единственным, который действовал без канала связи — компьютер «Супернова» является миникомпьютером, он во время игры находился на столе рядом со своим автором. По иронии судьбы в третьем туре компьютер вышел из строя; в выигранной позиции «Остричу» было зачтено поражение (в четвертом туре уже был мобилизован компьютер, находившийся в Монреале ...). Хотя «Острич» и использует небольшой компьютер, он играет на равных с другими программами. Идея Ньюборна состоит в том, чтобы на базе «Острича» определить те идеи, которые должны лечь в основу спектра программ, предназначенных для мини- и микрокомпьютеров.

Чемпионаты машин — зрелище прелюбопытное. За шахматными досками сидят авторы программ. С помощью терминалов по специальным каналам связи они сообщают своим компьютерам, находящимся на далеком расстоянии, очередной ход «противника». Ответный ход компьютера воспроизводится на дисплее и повторяется на шахматной доске. В ожидании ответа ученые дружелюбно беседуют, анализируют позицию, спорят, шутят и нередко критикуют игру своих программ. Да это и понятно: турнир компьютеров — спортивное соревнование лишь по форме, по существу же он преследует научные цели.

Когда закончился чемпионат, а за ним и показательная партия «Каисса — Чесс 4.6», Д. Каландер — консультант фирмы «Контрол дейта», машины которой неизменно использует американская программа, позвонил в Миннеаполис и дал задание компьютеру Сайбр-176 решить этюд Г. Надареишвили. Первые два хода за белых «Чесс 4.6» нашел, но после того, как был сообщен третий ход компьютера, Д. Каландер глянул на дисплей, рассмеялся и махнул рукой: компьютер рассмотрел около миллиона ходов, но правильного пути так и не нашел ...

Авторы программы «Чесс 4.6» Д. Слейт и Л. Аткин сообщили после турнира, что намерены отказаться от метода «брут форс» (грубая сила — так в США называют метод полного перебора), как бесперспективного, но будут совершенствовать свою программу «эволюционным» путем. Но возможно ли это? И сколь-

ко времени им потребуется, чтобы повторить путь, пройденный «Пионером»?

Если при методе «брут форс» большее быстроедействие ЭВМ дает лишь незначительное увеличение глубины расчета и довольно скромную прибавку в шахматной силе, то для «Пионера» быстроедействие компьютера примерно пропорционально глубине расчетов. Пока что «Пионер» играет медленно. Решение этюда Г. Надареишвили на машине такого же класса, как Сайбр-176, заняло бы менее десяти минут. Далеко не каждый шахматист даже самой высокой квалификации уложился бы в это время.

Третий чемпионат мира среди шахматных компьютеров намерено провести в 1980 году. Сначала в Японии состоится отборочный турнир, после чего лучшие две программы присоединятся к «Чесс 4.6» и «Каиссе» и отправятся в Австралию, где и определится третий чемпион мира. Можно рассчитывать, что этот чемпион будет уже хорошим шахматным мастером.

По окончании чемпионата в Торонто состоялось совещание участников — авторов программ, на котором голландский программист Б. Светс призвал к организации ИССА (Международной ассоциации машинных шахмат), что и было в принципе встречено одобрительно — несомненно, знамение времени!

#### ЗАКЛЮЧЕНИЕ

С тех пор, как автор этих строк пытался получить поддержку своему алгоритму, пришлось выслушать (с конца 1964 г.!) немало критических замечаний. Нелишним будет их перечислить. Утверждали, что все это является фантазией; это противоречит общепринятым канонам; потребует еще большего расхода ресурсов, чем решение, основанное на полном переборе; на это уйдут десятилетия; для составления программы потребуются коллектив, состоящий не менее, чем из 20 математиков; ресурсы современных ЭВМ явно недостаточны для реализации подобного алгоритма и т. д., и т. п. ...

Однако прошло 14 лет (а работа над программой началась лишь в 1972 г.), и мы на пороге решения большой научной проблемы. Алгоритм, моделирующий действия шахматного мастера, оказался реальным (а общепринятые методы все больше вызывают сомнения); работа продвигалась вперед при малом числе программистов; ресурсы современных мощных ЭВМ оказались вполне достаточными.

Если Архимеду, который создал теорию рычага, приписывают изречение «Дайте точку опоры и я переверну земной шар», то, прочитав эту книгу, читатель, вероятно, согласится с тем, что уместно ныне перефразировать великого ученого древности: «Дайте «Пионеру» быстроедействующий компьютер — и теория решения неточных задач поможет лучше управлять» ...

## ЗОНЫ ИГРЫ

Б. М. Штильман

Высшей ступенью системы управления, моделирующей шахматную игру по алгоритму М. Ботвинника, является математическое отображение (МО) — совокупность зон. В процессе перебора ходов МО непрерывно меняется путем включения — выключения зон, направляя тем самым перебор. В настоящем приложении рассмотрены вопросы формирования включенных зон и перебор ходов в совокупности включенных зон в математическом отображении, неизменном по составу включенных зон. Формирование и постоянные перестройки МО здесь не рассматриваются.

Все приводимые алгоритмы описаны содержательно. Особое внимание уделено вопросам их реализации в шахматной программе «Пионер». Заметим, что многие из рассмотренных здесь процедур были детализированы и уточнены в процессе создания программы «Пионер» и экспериментов с ней.

## 1. ФОРМИРОВАНИЕ ЗОНЫ И ПЕРЕБОР В НЕЙ

Фигура при движении по траектории стремится занять  $\alpha_k$ -поле этой траектории. Пусть при этом необходимо уничтожить неприятельскую фигуру, стоящую на этом поле. Подсистема второй степени должна быть организована так, чтобы в процессе передвижения атакующей фигуры к своей мишени неприятельские фигуры противодействовали нападению, а свои — поддерживали. Поскольку все они действуют по своим траекториям, такого согласования можно добиться, *подчинив* игру по отдельным траекториям общей цели совокупности траекторий. Совокупность траекторий и фигур-участниц «местного боя», разделенных на два противоборствующих лагеря, составляет зону игры.

**1.1. Понятие зоны.** Предположим, что на полях  $\alpha_0$  и  $\alpha_k$  шахматной доски стоят фигуры разного цвета (+) и (—), называемые  $\alpha_0$ -фигурой и  $\alpha_k$ -фигурой. Утверждение, что  $\alpha_0$ -фигура напала на  $\alpha_k$ -фигуру, означает, что существует траектория  $\alpha_0$ -фигуры, ведущая с поля  $\alpha_0$  на поле  $\alpha_k$ , причем такая, что время (в полуходах), затрачиваемое на передвижение  $\alpha_0$ -фигуры по этой траектории, не превышает некоторого числа  $N_L$ , называемого предельным горизонтом нападения. Рассмотренная траектория фигуры стороны (+) называется комлевой траекторией зоны, а  $\alpha_0$ -фигура — комлевой фигурой.

Рассмотрим совокупность полей комлевой траектории: это будут поля остановки ( $\alpha$ -поля) и поля, проходимые без остановки ( $\beta$ -поля). Множество траекторий фигур (—) и (+), заканчиваю-



щихся на этих полях, назовем траекториями первого отрицания. Совокупность траекторий фигур (—) и (+), заканчивающихся на полях траекторий первого отрицания, назовем траекториями второго отрицания и т. д.

Такую совокупность белых и черных фигур и определенным образом выбранных траекторий назовем зоной. (Цвет зоны — цвет коневой фигуры этой зоны.) Точный смысл понятия зоны будет установлен при описании способа вычисления траекторий зоны.

Покажем, как формируется зона игры, на примере специально подобранной искусственной позиции (рис. 19). Сначала, однако, необходимо рассмотреть, как программа «Пионер» производит перебор ходов.

**1.2. Процедура перебора ходов при поиске хода в оригинальной позиции.** Деревья перебора изображают корнем вверх, т. е. варианты перебора, ведущие из исходной позиции, изображаются ветвями дерева, растущими вниз. Из исходной позиции происходит спуск по дереву — последовательность ходов один за другим, в соответствии с некоторыми критериями вариант обрывается и полученная конечная позиция оценивается с помощью выбранной оценочной функции. Далее мы делаем ход «назад» (подъем по дереву) и, «поднимая» оценку снизу, оцениваем получившуюся позицию. Затем делаем новый ход из этой позиции, т. е. снова начинаем спуск по дереву, но уже по другой ветви. Когда вариант обрывается, опять производим ход назад и поднимаем оценку, делаем новый последний ход в варианте и т. д. После того, как проверены все возможные в рамках модели последние ходы, делаем новый предпоследний и т. д. Полученные внизу оценки переносятся вверх, причем для позиций с ходом черных выбирается минимальная, а для позиций с ходом белых — максимальная из оценок расположенных ниже позиций.

Блок-схема программы поиска хода в оригинальной позиции представлена на рис. 20.

Движение вниз по дереву перебора управляется процедурами той части блок-схемы, которая расположена справа на рис. 20. Движение вверх по дереву, необходимое для минимакса, а также для продолжения ветвления, управляется процедурами левой части

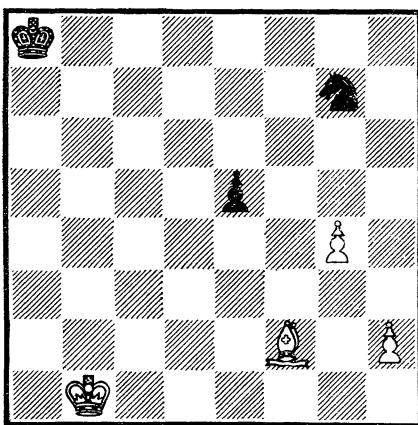


Рис. 19. Искусственная позиция

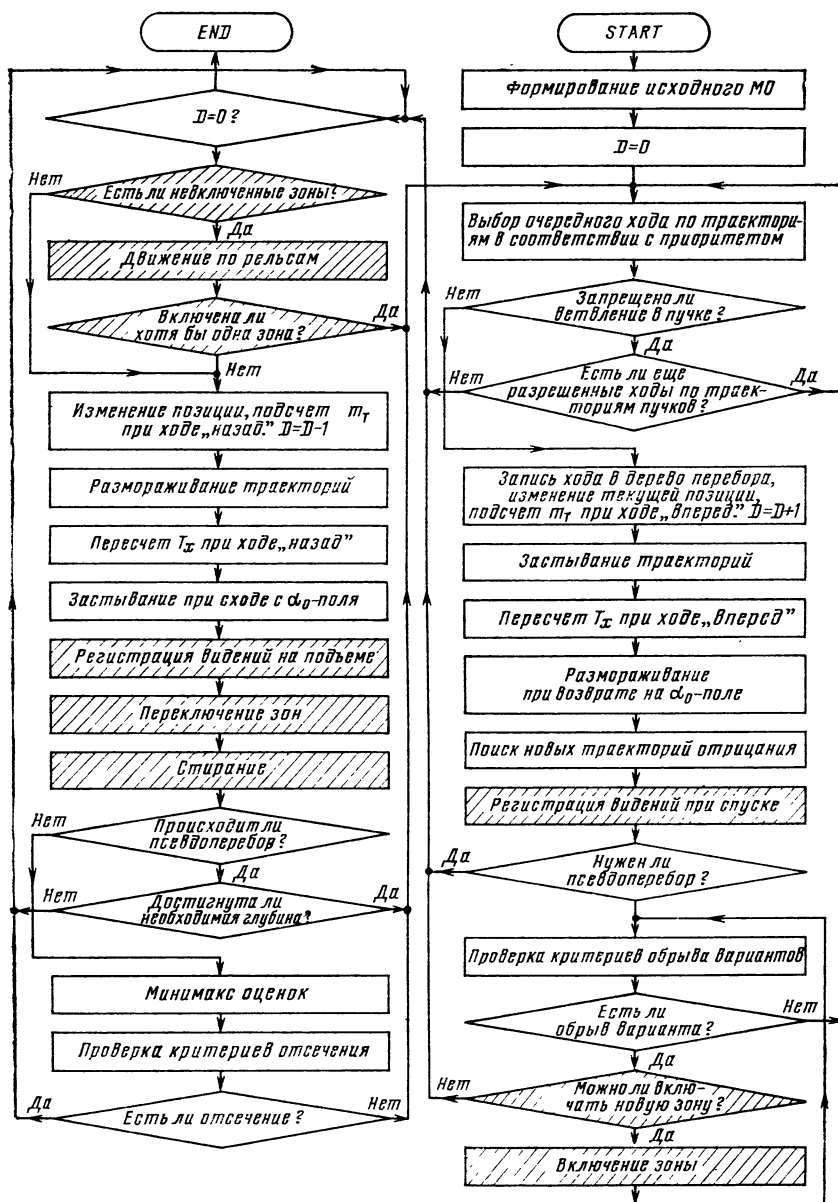


Рис. 20. Блок-схема программы поиска хода в оригинальной позиции

блок-схемы. *D* обозначает текущую глубину (в полуходах) в дереве перебора.

В рассматриваемом ниже примере в демонстрационных целях активизированы лишь незаштрихованные процедуры блок-схемы. Они управляют формированием зон в процессе перебора и подробно описаны несколько ниже. Остальные процедуры формируют математическое отображение.

**1.3. Пример формирования зоны в процессе перебора.** Итак, обратимся к позиции, изображенной на рис. 19. В этой позиции мы задали программе небольшой горизонт, равный двум передвижениям. Поэтому она может построить здесь только одну белую зону нападения слона f2 на пешку e5. Остальные возможные зоны не попадают в заданный горизонт, и программа, выяснив это, их не строит. Таким образом, в исходной позиции вычисляются траектории слона f2—g3—e5 и f2—d4—e5, траектории так называемого комлевого пучка. После этого программа начинает перебор ходов (рис. 21). Точные описания всех процедур, которые здесь поясняются на примере, даны в следующих пунктах.

Поскольку в позиции ход черных, а траекторий у них пока нет, т. е. нет и ходов, делается пропуск хода. Теперь очередь хода за белыми: они делают ход по комлевой траектории 2.f2—g3. После этого программа начинает поиск возможных траекторий противодействия за черных и поддержки за белых, т. е. траекторий контроля поля g3, причем для черных ищутся траектории, длина которых позволила бы черным успеть принять участие в борьбе за поле g3 (здесь два передвижения). В белой зоне все траектории поддержки белых фигур имеют длину в одно передвижение. Программа находит пучок траекторий коня g7—f5—g3 и g7—h5—g3, а также пешки h2—g3. Поскольку найдены новые траектории коня, движение по которым в данный момент уже не имеет смысла (оно запоздало), продолжение текущего варианта также теряет смысл. Надо вернуться назад по варианту на столько полуходов, чтобы найденные траектории коня включились в игру. В нашем случае надо подняться по дереву в исходную позицию. Таким образом, программа обрывает вариант без оценки и совершает все ходы в нем в обратном порядке, причем число 500 правее хода означает, что совершается подъем по варианту без оценки.

Заметим, что в машинной распечатке дерева перебора (рис. 21) расстояние от левого края до записи хода пропорционально глубине этого хода в дереве перебора, т. е. количеству полуходов в варианте, ведущем из исходной позиции к данной.

Итак, программа вернулась в исходную позицию. При возврате по варианту без оценки все построенное ранее дерево перебора было стерто в памяти ЭВМ. Программа снова начинает спуск по дереву, однако у черных уже появились ходы по траекториям. Делается ход... Kg7—f5 по траектории контроля. Здесь же программа находит траекторию пешки g4—f5. Поскольку игра по этой траектории в данный момент перебора имеет смысл, возврата по варианту не происходит. Прежде всего в варианте программа делает наимыгоднейшие взятия, поэтому следующий ход 2.g4:f5. Здесь вариант обрывается, так как черные потеряли три единицы материала (стоимость коня) — больше, чем они надеялись защитить (одну единицу — стоимость пешки e5). Соотношение белого и черного материала, снятого с доски в данном варианте, равно трем — это и есть оценка варианта. Программа начинает подъем по дереву с оценкой (напечатана справа от записи хода на рис. 21).

В позиции, возникающей после хода назад, соответствующего взятию 2.g4:f5, у белых есть еще не изученные ходы по траекториям (ходы слоном). Программа выясняет: поднимаемая оценка 3 настолько плоха для черных, что предыдущий свой ход (1...Kg7—f5) они все равно отбросят по минимуму. Поэтому исследовать другие ходы белых в данной позиции не имеет смысла, и

WHITE \*KB1,BF2,PG4,PH2,  
BLACK \*KA8,NG7,PE5,  
BLACK TO PLAY

```

*****
      BF2-G3
      BF2-G3   500
***** 500
NG7-F5
      PG4*F5
      PG4*F5   3
NG7-F5   3
NG7-H5
      *G4*H5
      PG4*H5   3
NG7-H5   3
*****
      BF2-G3
          *****
              B63*E5
              B63*E5   500
          ***** 500
      BF2-G3   500
***** 500
NG7-H5
      PG4*H5
      PG4*H5   3
NG7-H5   3
NG7-F5
      PG4*F5
      PG4*F5   3
NG7-F5   3
NG7-E6
      BF2-G3
          NE6-F4
              B63*F4
                  PE5*F4
                  PE5*F4   0
              B63*F4   0
          NE6-F4   0
      BF2-G3   0
      BF2-D4
          PE5*D4
          PE5*D4   -3
      BF2-D4   -3
NG7-E6   0

      SIZE OF THE TREE 13
      SIZE OF THE MAP 26
      TOTAL NUMBER OF MOVES CONSIDERED IS 18
      CPU TIME IS 25 SECONDS

```

Рис. 21. Машинная распечатка перебора ходов, выполненного программой «Пионер» при формировании зоны:

Здесь в записи хода первая буква — наименование фигуры, далее — обозначения полей, с которого и на которое выполняется ход. Буква К соответствует Кр, В — С, N — К, Р — п, \*\*\*\*\* — пропуск хода

в данном случае необходим возврат к исходной позиции. Происходит возврат по варианту без оценки, все ходы назад без оценки сопровождаются числом 500 (рис. 21). При этом подъеме все ранее построенное дерево перебора стирается в памяти ЭВМ.

Лотя, вернувшись в исходную позицию, программа не нашла еще удовлетворительной защиты, ее знания о позиции значительно расширились: сформирова-

программа продолжает подъем по варианту. В результате приходим в исходную позицию.

Начинается новый вариант с первым ходом 1 ... Kg7—f5 по траектории g7—h5—g3. Совершенно аналогично мы снова попадаем в исходную позицию. Итак, в исходной позиции все ходы черных, противодействующие нападению слона на пешку, получили оценки в пользу белых, т. е. пока удовлетворительной защиты не найдено. Перебор продолжается: черные пропускают ход, белые отвечают 2. Cf2—g3. Новых траекторий не найдено, однако есть старые траектории коня с g7 на g3. Программа выясняет, что время, отпущенное для движения коня по этим траекториям, исчерпано, поэтому такое движение не имеет смысла: конь не успеет принять участие в контроле поля g3.

Итак, у черных нет разрешенных ходов — они снова пропускают ход, белые отвечают 3. Cg3:e5. В полученной позиции программа находит траектории блокады конем g7 поля f4 (g7—h5—f4 и g7—e6—f4). Поле f4 является так называемым β-полем траектории слона f2—g3—e5, т. е. полем, проходным без остановки. Поиск траекторий контроля черными поля e5 не приводит к успеху: программа не находит ни одной такой траектории, длиной не превышающей три передвижения. (Заметим, что в позиции, отличающейся от изображенной на рис. 19 положением черного короля, он стоял на b7, в этот момент программа находила траектории короля с поля b7 на e5 и позднее включала их в игру.)

Поскольку найдены новые траектории блокады β-поля, необходимо возврат по варианту на столько полуходов, чтобы движение фигуры по ним приобрело смысл. Программа выясняет, что

лись траектории зоны, связанные с траекторией слона  $f2 - g3 - e5$ . Перебор ходов, по сути дела, начинается заново, однако с включением в игру всех найденных траекторий. Аналогично тому, как было раньше, изучаются ходы  $1...Kg7 - h5$  и  $1...Kg7 - f5$ ; меняется лишь порядок их рассмотрения, поскольку ход  $1...Kg7 - h5$  имеет более высокий приоритет как ход по «вилочной» траектории (через поле  $h5$  проходят две траектории коня  $g7 - h5 - g3$  и  $g7 - h5 - f4$ ).

Наконец, не найдя ничего нового, программа переходит к изучению хода  $1...Kg7 - e6$  по траектории блокады  $g7 - e6 - f4$ . Белые продолжают  $2.Cf2 - g3$ , черные блокируют  $2...Ke6 - f4$ . Здесь программа находит траектории слона  $g3 - f4$  и пешки  $e5 - f4$ . Для включения их в игру возврат по варианту не нужен. Как выгодные взятия ходы  $3.Cg3 : f4$   $e5 : f4$  рассматриваются в первую очередь. На этом вариант обрывается, так как потеряна основная (комлевая) фигура нашей зоны — слон, и размен закончен. Вариант оценивается нулем, поскольку нулю равно соотношение снятого с доски материала. Начинается подъем по варианту с оценкой.

Так как нулевая оценка хода  $2.Cf2 - g3$  белых не устраивает, программа заменяет этот ход на  $2.Cf2 - d4$  по траектории  $f2 - d4 - e5$ . Заметим, кстати, что в этот момент черный конь стоит на  $e6$ . После хода слонем на  $d4$  программа получает траектории пешки  $e5 - d4$  и коня  $e6 - d4$ , которые сразу же включаются в игру. Однако приоритет имеет взятие пешкой как более выгодное. Вариант обрывается с оценкой  $-3$ , так как потеряна комлевая фигура (слон). Происходит подъем по дереву с оценкой, причем оценивая ход  $1...Kg7 - e6$ , программа производит минимакс оценок — выбирается наибольшая (выгоднейшая для белых) оценка из двух  $0$  и  $-3$ .

В результате перебора программа сформировала в памяти ЭВМ зону игры. Проведенный программой перебор ходов нельзя рассматривать как полный анализ исходной позиции. Дело в том, что и оценки вариантов, и сформированное программой МО достаточно условны, поскольку включена в игру и построена единственная зона. Таким образом, данный пример лишь иллюстрирует алгоритм формирования зоны.

**1.4. Формирование зоны.** Построив в исходной позиции лишь комлевую траекторию, будем формировать зону в процессе перебора ходов и в дальнейшем осуществлять сам перебор, передвигая фигуры по построенным к данному моменту траекториям зоны. Принято, что траектория произвольной фигуры от поля  $a_0$  до поля  $a_k$  может быть построена лишь в том случае, если на поле  $a_k$  в процессе перебора появится неприятельская фигура. Таким образом, если в процессе перебора фигура не побывает на каком-либо участке траектории, то этот участок не будет атакован и на нем не возникнут траектории более высоких степеней отрицания.

Предложение о том, чтобы связать основные операции алгоритма М. Ботвинника с перебором ходов и, в частности, таким способом формировать зону в процессе перебора, внесено автором настоящего приложения. Это привело к уменьшению объема информации, характеризующей модель, а также к значительному сокращению дерева перебора.

Некоторые этапы формирования зоны проиллюстрированы на рис. 22,а—в, причем штрихами обозначены участки траекторий, на которых в процессе перебора не побывали соответствующие этим траекториям фигуры. Зону формирует процедура поиска новых

траекторий отрицания (рис. 20), рассмотренная ниже.

Заметим, что употребляя термин «траектория», мы имеем в виду «пучок траекторий», так как операция определения траектории некоторой фигуры с одного поля на другое количеством передвижений (длиной), не превышающим фиксированного числа, дает в общем случае не одну, а несколько траекторий [3], содержащих большое количество информации, особенно если длина траекторий превышает два передвижения на свободной от фигур доске.

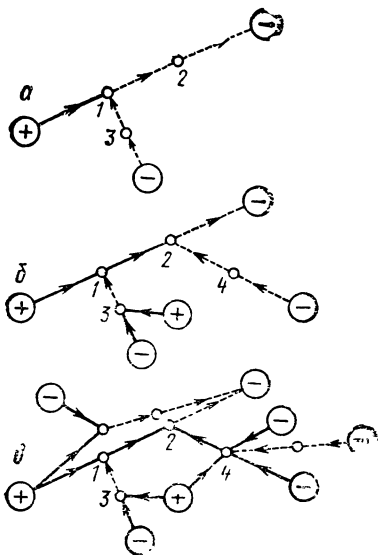


Рис. 22. Некоторые этапы формирования зоны игры

1	Номер фигуры
2	$\alpha_k$ -поле пучка
3	Длина кратчайшей траектории пучка
4	Блокада или контроль
5	Номер атакованной фигуры
6	Глубина застывания
7	$\alpha_i$
8	$\alpha_{i+1}$
9	$\alpha_0$ -поле пучка
10	0-траектория отрицания
11	Адрес роста зоны
12	Параметр запрета ветвления в пучке
13	Время, отпущенное для игры в зоне ( $T_x$ )

Рис. 23. Структура стандартной ячейки цепного списка (следа пучка)

**1.5. Размещение информации о пучках траекторий в памяти ЭВМ. Цепные списки.** Прежде чем воспользоваться этой информацией, следует решить вопрос о ее хранении. Сведения о пучках траекторий нужны на протяжении всей работы программы, так как от них зависит формирование зон. Какие же из этих сведений необходимо хранить? Решено хранить в памяти не сам пучок (т. е. списки полей всех траекторий), а лишь информацию о существовании этого пучка и его типе, а именно: номер \* фигуры, траектории которой содержатся в этом пучке,  $\alpha_k$ -поле пучка, длину кратчайшей траектории пучка, тип пучка (блокады, контроля), пара-

\* Номером фигуры назовем натуральное число, однозначно характеризующее данную фигуру.

метры, связывающие этот пучок с другими пучками, а также с более высокими ступенями системы управления. Полностью список параметров представлен на рис. 23. Здесь описаны упомянутые выше четыре параметра. Остальные параметры описаны в следующих пунктах.

Итак, пучок характеризуется списком параметров. Эта информация должна быть «привязана» к полю шахматной доски, на котором стояла фигура, когда этот пучок был впервые найден, т. е. к  $\alpha_0$ -полю. По первым четырем параметрам из перечисленного списка пучок однозначно восстанавливается с помощью подпрограммы получения пучка траекторий [3].

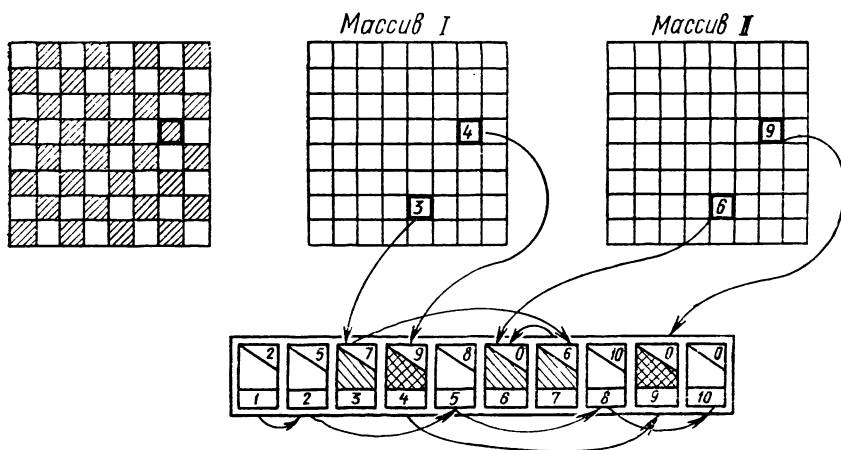


Рис. 24. Размещение множества пучков в оперативной памяти ЭВМ

Сведения о пучках можно было бы хранить в виде многомерно-го массива, однако учитывая, что каждый из параметров является двузначным десятичным числом (в общем случае), линейная размерность массива составляла бы  $10^{26}$  ( $100^{13}$ ), а это, конечно, неприемлемо. Поэтому «Пионер» хранит информацию в виде цепного списка (рис. 24). Память, отводимая для размещения множества пучков, разбивается на «ячейки». На рисунке показаны 10 таких ячеек. Каждая ячейка предназначена для хранения списка параметров, а также адреса сцепленной с ней ячейки (указан в правом верхнем углу ячейки). Кроме того, отводим память для двух массивов  $8 \times 8$ , нумерация элементов в которых соответствует нумерации полей на шахматной доске.

Информация обо всех пучках, проходящих через  $\alpha$ -поле, привязывается к  $\alpha$ -полям этих массивов  $8 \times 8$ . Это делается следующим образом: на  $\alpha$ -поле массива I записывается адрес ячейки, со-

держашей информацию о первом пучке, проходящем через  $\alpha$ -поле, затем все ячейки с информацией о пучках, относящихся к  $\alpha$ -полю, связываются в цепочку, первым звеном которой является упомянутая выше ячейка. Адрес последнего звена цепочки записывается на  $\alpha$ -поле массива II. Все ячейки, в которых в данный момент не содержится информации о пучках, также связаны в цепочку.

Подобная система обеспечивает возможность стирания из некоторых ячеек информации, ставшей ненужной в процессе перебора или после того, как фигура сделает ход в игре, и использования освободившейся памяти для новой информации. Это происходит присоединением освободившихся ячеек к списку пустых.

Если в процессе перебора фигура побывает на  $\alpha$ -поле, то информация обо всех пучках этой фигуры, проходящих через  $\alpha$ -поле, будет привязана к данному полю. Если при продолжении перебора фигура снова появится на  $\alpha$ -поле (речь идет о другом варианте перебора), описанная система размещения информации дает возможность восстановить все пучки траекторий этой фигуры, связанные с данным полем и найденные в процессе предшествующего перебора. Это происходит при последовательном просмотре цепного списка ячеек, связанного с данным  $\alpha$ -полем, поиске нужных ячеек, считывании из них параметров пучков и применении подпрограммы получения пучков траекторий с этими параметрами. Эта операция называется *разворачиванием пучков*. Ячейка цепного списка, привязанного к данному полю, содержащая информацию о некотором пучке, называется *следом пучка на данном поле*.

**1.6. Информация о траекториях зоны.** Информация о траектории некоторой фигуры в зоне связывается с теми  $\alpha$ -полями этой траектории, т. е. с полями шахматной доски (массивом  $8 \times 8$ ), на которых в процессе перебора успевает побывать эта фигура (см. п. 1.5). Таким образом, с каждым полем массива  $8 \times 8$  связан цепной список ячеек, в каждой из которых записана информация об одном из пучков траекторий, проходящих через это поле. Нередко списки, связанные с некоторыми полями массива  $8 \times 8$ , могут оказаться пустыми. На рис. 23 показана структура стандартной ячейки цепного списка (след пучка). В ней записаны, в частности, глубина в дереве перебора, на которой при подъеме по дереву этот пучок должен быть разморожен, если при спуске по дереву он застыл (см. ниже). Кроме того, если данный пучок является пучком траекторий блокады  $\beta$ -поля траектории атакующей фигуры на участке  $\beta \in (\alpha_i, \alpha_{i+1})$ , то концы интервала также записываются в ячейку.

Одно из чисел, записанных в ячейке, — адрес ростка зоны — содержит информацию о зоне, которой принадлежит данный пучок. *Ростком зоны* называется след комлевого пучка зоны на  $\alpha_0$ -поле этого пучка. Адрес этого следа в цепном списке и является тем параметром, который однозначно характеризует зону игры. Если след



некоторого пучка сам является ростком зоны, то позиция, отводящая в нем под адрес ростка, используется для другого параметра — глубины включения этой зоны в дерево перебора (вопросы включения зон в настоящем приложении не рассматриваются).

**1.7. Параметр  $T_x$ .** Зона формируется по следующему принципу. Сторона (+) может передвигать лишь  $\alpha$ -фигуру по комлевой траектории, а другие фигуры (+) включаются в зону, если находятся на полях, где они готовы к взятию; сторона (—) включает в зону лишь те свои фигуры, которые успевают принять участие в игре. Это последнее определяет переменный горизонт  $H_x$  — время (в полуходах), отведенное фигурам (—) для того, чтобы принять участие в борьбе на  $\alpha$ - или  $\beta$ -поле комлевой траектории [3].

Непосредственно пользоваться величиной  $H_x$  для поиска траекторий отрицания неудобно. Вместо величины  $H_x$  введен вспомогательный параметр  $T_x$ . Он имеет ту же размерность, что и длина траектории, т. е. количество передвижений. Параметр  $T_x$ , синхронно изменяющийся с каждым ходом фигур по траекториям зоны (см. ниже), дает разрешение или запрещение на передвижение данной фигуры. Кроме того, параметр  $T_x$  необходим для вычисления максимальной длины траекторий фигур (—), которые могут быть включены в зону.

**1.8. Вычисление траекторий отрицания.** Пусть в переборе некоторая фигура делает ход  $X \rightarrow Y$ . Определим траектории всех фигур, стоящих на доске в позиции после хода  $X \rightarrow Y$ , ведущие на поле  $Y$ . Если фигура, стоящая на поле  $Z$ , имеет траекторию, ведущую на поле  $Y$ , то мы говорим, что произошло *видение с поля  $Z$  на поле  $Y$* . Для регистрации видений необходимо знать максимальную длину ( $A_{\max}$ ) искомым траекторий.

Кроме того, чтобы сохранить информацию о новой траектории фигуры с поля  $Z$  на поле  $Y$ , надо для нее заполнить след-ячейку, которая затем присоединится к цепному списку ячеек, связанному с полем  $Z$  массива  $8 \times 8$ . Наибольшую сложность для определения в списке параметров следа представляет параметр  $T_x$ .

Итак, надо заранее определить два числа:  $A_{\max}$  и  $T_x$ . Эти числа зависят от того, поиск каких траекторий производится.

Если вычисляется комлевая траектория зоны, то  $A_{\max} = \text{ENT}[(H_L + 1)/2]$ , где  $H_L$  — горизонт (в полуходах);  $T_x = 1$ .

Если же мы ищем траекторию отрицания, то  $A_{\max}$ , а также исходное значение  $T_x$  зависят от того, с какой траекторией связана искомая, т. е. по какой траектории зоны сделан ход  $X \rightarrow Y$ . Чтобы определить это, изучают следы траектории на полях  $X$  и  $Y$ . Из этих же следов мы можем извлечь значение  $T_x$  (обозначим  $T$ ) для траектории фигуры, сделавшей ход, а также  $A$  — длину оставшегося участка этой траектории.

Рассмотрим все возможные случаи.

1. Фигура сделала ход по комлевой траектории. Тогда длина  $A_{\max}$  искомой траектории отрицания не превышает  $T$  (см. ниже изменение  $T_x$  при движении комлевой фигуры):  $A_{\max} \leq T$ . Параметру  $T_x$  искомой траектории присваиваем значение 1.

2. Фигура сделала ход по траектории отрицания:

а) на конечное поле траектории; в этом случае  $A_{\max} = T - 1$ ,  $T_x = T$ ;

б) не на конечное поле траектории, причем на оставшемся участке кратчайший путь по траекториям пучка имеет длину  $A$ ; тогда  $A_{\max} = T - A$ ,  $T_x = A_{\max} + 1$ .

Если в случаях 1 и 2 атакуется  $\beta$ -поле (т. е. мы ищем траектории с поля  $Z$  на поля, принадлежащие интервалу  $(X, Y)$ ), то значения  $A_{\max}$  и  $T_x$  вычисляются так же, а затем из них вычитается единица. В рассмотренных случаях, вычисляя максимальную длину траекторий отрицания ( $A_{\max}$ ), мы имели в виду траектории стороны  $(-)$ . Принято, что траектории стороны  $(+)$  в зоне (за исключением комлевой) могут иметь длину  $A_{\max} \leq 1$ .

Таким образом, в зоне игры из фигур  $(+)$  может перемещаться только комлевая фигура (если нет взятия); контролирующие фигуры  $(+)$  могут находиться лишь в засаде, т. е. их траектории имеют длину в одно передвижение. Предельная длина траекторий отрицающих фигур  $(-)$  определяется параметром  $T_x$ , следовательно, фигуры  $(-)$  включаются в зону лишь тогда, когда они успевают принять участие в игре.

**1.9. Коррекция метода видения.** Поиск траекторий отрицания методом прямого видения по сравнению с другими методами обеспечивает значительное сокращение информации, хранящейся в памяти ЭВМ. Речь идет, например, о методе априорного построения полной зоны. При методе видения как отдельные траектории, так и целые зоны часто оказываются недостроенными. Так произошло и в рассмотренном выше примере формирования зоны игры. Зона формируется лишь в той части, где реально в процессе перебора побывали фигуры, где шла «борьба». Те же участки, на которые фигуры не дошли, не обрастают новыми траекториями, и, таким образом, память ЭВМ не засоряется лишней информацией, которая, в свою очередь, может привести к увеличению перебора ходов.

Однако метод видения предполагает, что к моменту видения после хода  $X - Y$  на поле  $Z$ , с которого мы ищем траекторию на поле  $Y$ , должна стоять соответствующая фигура, что вовсе необязательно для существования траектории. Предположим, что некоторая фигура в процессе игры в зоне побывала на поле  $Z$ , причем имея она траекторию с этого поля на поле  $Y$ , она успела бы принять участие в борьбе за поле  $Y$ . Но к моменту видения эта фигура уже сошла с поля  $Z$ , и соответствующая траектория не была найдена. Таким образом, при методе видения мы теряем некоторые

траектории. Рассмотренный недостаток метода видения был замечен лишь в процессе экспериментов с программой «Пионер».

Компенсируем этот недостаток коррекцией метода видения. Пусть фигура сделала ход  $X \rightarrow Y$  по траектории некоторой зоны. Тогда можно определить глубину включения этой зоны в дерево перебора. Это число записано в ростке зоны (см. п. 1.6). Получая управление, процедура поиска новых траекторий отрицания (см. рис. 20) производит возврат по текущему варианту перебора, причем подъем по дереву осуществляется до глубины, равной глубине включения данной зоны. Во время этого подъема синхронно с каждым ходом «назад» выполняются лишь две операции: изменяются позиции фигур и пересчитывается  $T_x$  траектории, проходящей через поля  $X$  и  $Y$  (см. ниже). В позиции, где подъем прекратился, процедура методом видения определяет все траектории отрицания, ведущие на поле  $Y$  (а также траектории блокады участка  $(X, Y)$ ), причем  $A_{\max}$  и  $T_x$  новых траекторий определяются так же, как рассмотрено выше.

Процедура производит спуск по той ветви дерева, где только что произошел подъем. С каждым ходом  $X_1 \rightarrow Y_1$  при спуске вычисляются новые траектории фигуры, сделавшей этот ход, т. е. с поля  $Y_1$  на поле  $Y$  (и на участок  $(X, Y)$ ). Так процедура выполняет спуск до хода  $X \rightarrow Y$ . Описанный подъем по ветви дерева перебора с последующим возвратом исправляет упомянутый выше недостаток метода видения.

**1.10. Псевдоперебор.** В тот момент, когда найдена новая траектория отрицания длиной более чем в одно передвижение (если это траектория блокады, то длина произвольная), движение фигуры по ней уже не имеет смысла в данном варианте перебора. Это происходит потому, что новая траектория является либо траекторией блокады некоторого  $\beta$ -поля атакованной фигуры, и тогда последняя уже прошла мимо этого поля, либо траекторией контроля  $\alpha$ -поля, причем атакованная фигура уже стоит на этом  $\alpha$ -поле, и контроль запоздал. Из-за невозможности включить в игру новые траектории, найденные «слишком поздно», продолжение текущего варианта теряет смысл. Поэтому необходимо прекратить спуск и подняться по ветви на такую высоту в дереве перебора, чтобы новые траектории включились в игру. При этом поддерево, образованное ветвями, исходящими из узлов текущего варианта, которые мы проходим при подъеме, должно быть стерто в памяти ЭВМ, а соответствующий этому поддереву перебор ходов объявлен предварительным.

Итак, назовем *псевдоперебором* подъем по ветви дерева (соответствующей текущему варианту перебора), предпринятый только для включения в игру новых траекторий.

**1.11. Глубина подъема по ветви дерева при псевдопереборе.** Покажем, как определяется глубина подъема при псевдопереборе.

Предположим, что в некоторый момент перебора найдены новые траектории нескольких фигур. Определим, какая из этих фигур дольше всего простояла на своем  $\alpha_0$ -поле в текущем варианте. Кажется бы, наивысший узел в дереве, где эта фигура еще стоит на  $\alpha_0$ -поле (проследивая назад текущий вариант), можно принять за глубину подъема; однако здесь надо учесть, что зона, которой принадлежат новые траектории, при подъеме по дереву на такую глубину может перестать существовать (т. е. мы поднимемся выше предельной глубины включения данной зоны). Таким образом,

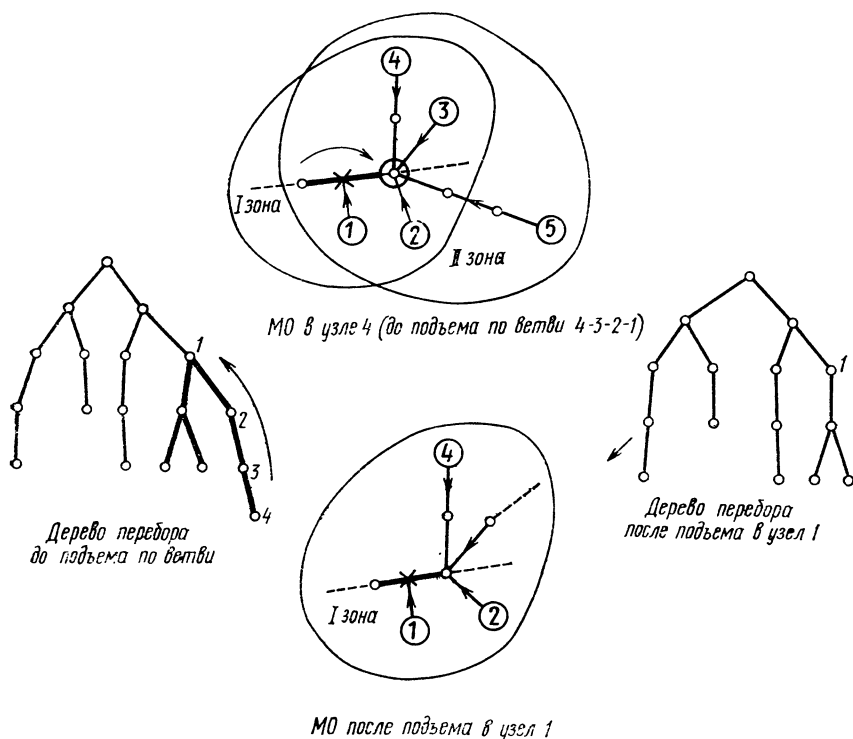


Рис. 25. Процедура псевдоперебора

сравнивая два числа: глубину подъема, на которой фигура еще стоит на  $\alpha_0$ -поле, и глубину включения данной зоны, надо наибольшее из этих чисел принять за глубину подъема до того узла, где и включаются в игру новые траектории данной зоны.

Отметим, что все глубины в исходной позиции равны нулю и на протяжении вариантов возрастают. Далее, примем в расчет, что перебор, в общем случае, происходит не в одной зоне, а траектории или их участки принадлежат сразу нескольким зонам. Мы

показали, как определяется глубина подъема для каждой из этих зон. Вычислив глубины, получим несколько чисел — наименьшее из них и есть окончательная глубина подъема при псевдопереборе. Процедура псевдоперебора проиллюстрирована на рис. 25. Изображенная здесь зона II оказывается слишком «короткоживущей» и при подъеме по дереву перестает существовать; все принадлежащие ей траектории, включая вновь найденные, стираются в памяти ЭВМ.

**1.12. Застывание траекторий из-за недостаточного значения параметра  $T_x$ .** Рассмотрим теперь, как с каждым ходом в зоне пересчитывается параметр  $T_x$  для каждой фигуры (—) этой зоны. Этот параметр подсчитывается при спуске по дереву процедурой пересчета  $T_x$  при ходе «вперед», а при подъеме — процедурой пересчета  $T_x$  при ходе «назад» (см. рис. 20). Если делается ход комлевой фигурой по комлевой траектории, то ее собственный параметр  $T_x$  увеличивается на единицу, а на траекториях отрицания уменьшается на единицу. Параметр  $T_x$  комлевой траектории служит лишь для определения максимальной длины вновь вычисляемых траекторий первого отрицания. Если в переборе делается ход по траектории отрицания некоторой зоны, то параметр  $T_x$  увеличивается на единицу на траекториях более высокой степени отрицания, связанных с траекторией, по которой сделан ход. На остальных траекториях зоны при таком ходе параметр  $T_x$  не изменяется.

Предположим, что некоторая фигура стороны (—) стоит на поле  $\alpha_i$  траектории отрицания зоны, причем кратчайший путь с поля  $\alpha_i$  до конечного поля траектории равен  $A$  (передвижений), а время, отпущенное для игры в зоне по этой траектории, определяется параметром  $T_x$ . Правила игры в зоне таковы, что данной фигуре разрешается сделать ход  $\alpha_i \rightarrow \alpha_{i+1}$ , если 1) этот ход является взятием либо 2)  $A \leq T_x$  и эта траектория не застыла по другим критериям (см. ниже). Таким образом, при  $T_x < A$  траектория *застывает из-за недостаточного значения параметра  $T_x$* .

При подъеме по дереву параметр  $T_x$  изменяется аналогично; только, если данный ход при спуске по дереву привел к тому, что значение  $T_x$  некоторой траектории увеличилось на единицу, при подъеме оно уменьшается на единицу и наоборот; если же при спуске значение  $T_x$  не менялось, то и при подъеме оно не изменится. Таким образом, спускаясь из некоторого узла дерева перебора, при возвращении в него (в процессе продолжения перебора и минимакса) мы обеспечиваем восстановление параметра  $T_x$  для всех траекторий зоны.

На рис. 26 показан пример подсчета параметра  $T_x$  для различных траекторий отрицания. Значения параметра  $T_x$  в исходном положении фигур получены следующим образом (для простоты рассмотрим лишь фигуру 4). В соответствии с упомянутыми выше

(см. п. 1.8) правилами заполнения следа комлевой пучка в исходном положении для комлевой фигуры имеем  $T_x=1$ . Допустим, что траектории отрицания еще не построены. При движении (в процессе перебора) комлевой фигуры значение ее  $T_x$  с каждым ходом увеличивается на единицу. После хода 3—4, когда она оказывается на  $\alpha$ -поле 4, ее  $T_x=4$ . Находим траекторию фигуры 4, длиной не превышающую четырех передвижений. Для нее  $T_x=1$  (см. п. 1.8). Далее управление передается процедуре псевдоперебора, происходит подъем по дереву к исходному положению ком-

Ход	Значение параметра $T_x$				
	Фигура				
	1	2	3	4	5
Исходное положение фигур	3	4	3	4	2
1 11-10	3	4	3	4	2
1-2	2	3	2	3	①
1 10-9	2	3	2	3	1
2-3	1	2	①	2	0
2 12-8	2	2	2	2	1
3-4	1	1	1	①	0

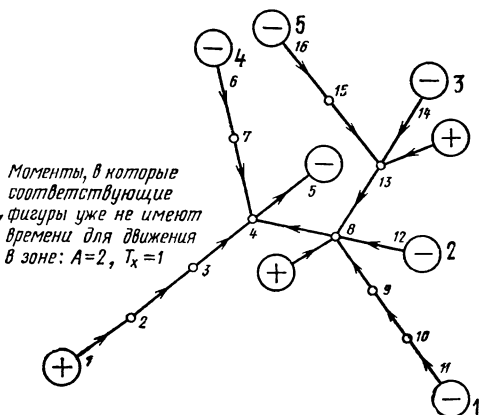


Рис. 26. Синхронный с перебором подсчет параметра  $T_x$

левой фигуры. При этом с каждым ходом назад комлевой фигуры параметр  $T_x$  фигуры 4 увеличивается на единицу, поэтому в исходном положении  $T_x=4$ . Отметим здесь: если траектория в некотором узле существует, однако из-за отсутствия видения в этом узле была впервые определена в другом узле ниже по дереву, как на рис. 26, то начальное значение  $T_x$  для нее (из п. 1.8) таково, что изменение  $T_x$  на подъеме (при псевдопереборе) обеспечит при возврате в верхний узел такой  $T_x$ , как если бы эта траектория была определена в этом узле и ранее.

**1.13. Пересчет  $T_x$  на траекториях отрицания.** Уточним, как работает алгоритм пересчета параметра  $T_x$  при ходе фигурой по траектории отрицания. В соответствии с рассмотренным правилом надо среди всех траекторий зоны выделить траектории более высоких степеней отрицания, связанные с траекторией, по которой сделан ход, и на них увеличить значение  $T_x$  на единицу.

Для этого берем произвольную траекторию зоны, затем переходим к траектории, к которой она привязана, затем к траектории, к которой привязана эта последняя, и т. д. Так, переходя от тра-

ектории к траектории (причем степень отрицания этих траекторий понижается), мы в результате можем прийти к комлевой траектории, и тогда у исходной траектории параметр  $T_x$  не меняется. В этом случае происходит переход от траектории к траектории по другому варианту, и если мы придем к траектории, по которой был сделан ход, то на исходной траектории изменяем  $T_x$ . Эта процедура повторяется для каждой траектории отрицания зоны.

**1.14. Застывание траекторий при потере связи с действующей траекторией.** Процедуре застывания траекторий при спуске по дереву соответствует процедура размораживания траекторий, выполняющаяся при подъеме (см. рис. 20). Эти процедуры реализуют следующий процесс.

При движении фигуры по траекториям пучка с каждым ходом появляются участки траекторий и целые траектории пучка, на которые эта фигура уже не попадет при продолжении вариантов перебора. Поэтому траектории, связанные с этими участками, должны быть выключены из игры. Ту же операцию надо проделывать с траекториями более высоких степеней отрицания, связанных с выключенными. Назовем эту операцию **застыванием при потере связи с действующей траекторией**.

На рис. 27 иллюстрируется процедура застывания при ходе 1—2 фигуры 1. Застывают следующие участки и траектории: 1—2, 3—5—6, 14—9—5, 13—3. При подъеме по варианту соответствующие траектории должны быть снова включены в игру и использованы в других вариантах перебора; эта операция называется размораживанием. Такое выключение из игры некоторых траекторий в данном узле дерева перебора при спуске по дереву, а затем включение их при возврате в этот узел на подъеме называется *процедурой застывания — размораживания при потере (восстановлении) связи с действующими траекториями*. Покажем, как работает эта процедура.

**1.15. Критерии застывания.** Рассмотрим  $\alpha$ -поле, принадлежащее нескольким пучкам траекторий данной фигуры. Назовем  $\alpha$ -поле полностью застывшим, если застыли все траектории этих пучков, проходящие через это поле. Примером полностью застывших полей могут служить поля 1, 3, 5 для фигуры 1, поле 5 для фигуры 4 (рис. 27). Траектория контроля некоторого  $\alpha$ -поля застывает лишь

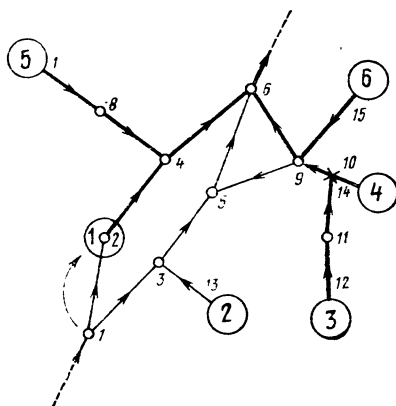


Рис. 27. Процедура застывания при потере связи с действующей траекторией

в том случае, если это  $\alpha$ -поле является полностью застывшим. Например, траектории 13—3 фигуры 2, 14—9—5 фигуры 4 застывают. Траектория блокады  $\beta$ -поля на участке  $\beta \in (\alpha_i, \alpha_{i+1})$  застывает лишь в том случае, если один из концов участка  $\alpha_i$  или  $\alpha_{i+1}$  является полностью застывшим  $\alpha$ -полем. Траектория 12—11—10 фигуры 3 (рис. 27) не застывает, так как концы участка (9—14) не являются полностью застывшими  $\alpha$ -полями. Как известно, для траектории блокады  $\beta$ -поля концы интервала  $(\alpha_i, \alpha_{i+1})$  входят в список сведений о траектории, хранящийся в памяти ЭВМ (см. рис. 23).

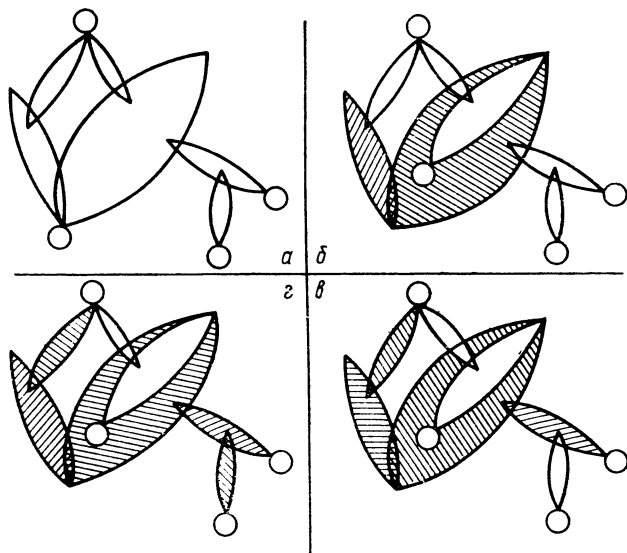


Рис. 28. Последовательность операций при застывании пучков траекторий

Таким образом, рассмотрены критерии застывания траекторий более высоких степеней отрицания, связанных с данной траекторией.

В момент застывания  $\alpha$ -поля некоторого пучка в соответствующем следе записывается текущая глубина в дереве перебора. При подъеме по ветви и возврате в этот узел дерева перебора все пучки с данной глубиной застывания размораживаются, а глубина застывания приравнивается нулю (на нулевой глубине ни один пучок не является застывшим в указанном смысле).

**1.16. Алгоритм застывания.** Предположим, что в процессе перебора сделан ход  $X \rightarrow Y$ . Разберем процедуру застывания (рис. 28). Прежде всего анализируется, по траекториям каких пучков сделан



ход. При этом в специальном массиве  $8 \times 8$  помечаются поля, принадлежащие траекториям, по которым сделан ход, причем рассматриваются только те  $\alpha$ - и  $\beta$ -поля, до которых фигура еще не дошла. Информация обо всех указанных пучках записывается в соответствующих ячейках цепного списка, привязанного к полю  $Y$ . После этого управление передается процедуре застывания. По очереди на всех полях доски, кроме отмеченных в специальном массиве  $8 \times 8$ , происходит поиск информации о незастывших пучках данной фигуры, проходящих через поле  $X$ . Если на некотором поле  $Z$  найдена информация о таком пучке, т. е. след пучка с числом нуль в шестой позиции, то  $\alpha$ -поля  $X$  и  $Z$  этого пучка застывают. Так происходит застывание траекторий данной фигуры (рис. 28,б). (На рис. 28,а—г застывшие пучки заштрихованы.)

Теперь необходимо, чтобы произошло застывание пучков более высоких степеней отрицания. Начинается новый просмотр всех полей массива  $8 \times 8$ , при этом осуществляется поиск информации о незастывших пучках любой фигуры, кроме той, которая сделала ход. Если на некотором поле  $V$  найдена информация о таком пучке, то из соответствующей ячейки извлекается информация о пучке более низкой степени отрицания, с которым данный пучок связан. Проверяются рассмотренные выше критерии застывания пучков более высоких степеней отрицания. При выполнении одного из них  $\alpha$ -поле  $V$  данного пучка застывает, а просмотр массива  $8 \times 8$  продолжается. После первого просмотра застынут пучки, связанные с застывшими траекториями фигуры, сделавшей ход (рис. 28,в).

Обозначим множество этих пучков через  $\{F\}_1$ . Вторично просмотрим массив  $8 \times 8$ , выполняя те же операции, что и при первом просмотре. В результате получим  $\{F\}_2$  — множество застывших пучков, связанных с пучками из  $\{F\}_1$ . Продолжая этот процесс, после  $n$ -го просмотра получим, что ни один новый пучок не застыл, т. е.  $\{F\}_n$  пусто. (Для случая, изображенного на рис. 28,  $n=3$ .) На этом процедура застывания заканчивается.

**1.17. Застывание при сходе с  $\alpha_0$ -поля на подъеме по дереву.** Процедура застывания при сходе с  $\alpha_0$ -поля при подъеме по дереву соответствует процедуре размораживания при возврате на  $\alpha_0$ -поле, которая выполняется при спуске по дереву (рис. 20).

В процессе перебора при подъеме по дереву фигура может, вообще говоря, сойти с  $\alpha_0$ -поля своей траектории. Это происходит потому, что многие траектории фигур строятся не в исходной позиции, а в глубине дерева перебора, когда эти фигуры сходят со своих исходных полей. Поэтому при подъеме по дереву в некоторый момент может оказаться, что среди прочих есть незастывшая траектория фигуры, в то время как эта фигура не стоит ни на одном из полей своей траектории. Если с  $\alpha_0$ -поля сходит комлевая фигура зоны, то все траектории зоны стираются в памяти ЭВМ. При некомлевой фигуре выгоднее, чтобы такая траектория засты-

ла. Однако это застывание другого типа по сравнению с уже рассмотренными. Во-первых, оно происходит при подъеме по дереву, а размораживание — при спуске (при вступлении фигуры на  $\alpha_0$ -поле). Во-вторых, застывают всегда целые пучки, а не отдельные участки траекторий.

Рассмотрим процедуру застывания (рис. 29). При подъеме по дереву после хода назад  $Y \rightarrow X$  просматривается весь массив  $8 \times 8$  с целью поиска пучков, имеющих  $Y$  своим  $\alpha_0$ -полем. Это возможно, так как информация об  $\alpha_0$ -поле пучка содержится в соответствующей ячейке цепного списка (см. рис. 23).

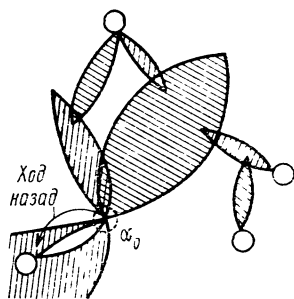


Рис. 29. Застывание пучков при сходе фигуры с  $\alpha_0$ -поля в процессе подъема по ветви дерева перебора (застывшие пучки траекторий заштрихованы)

застывших в смысле «99», имеющих данное поле своим  $\alpha_0$ -полем. Размораживаются эти пучки, а затем и пучки более высоких степеней отрицания.

Размораживание траекторий из-за прихода фигуры на  $\alpha_0$ -поле при спуске по дереву еще не определяет включения этих траекторий в перебор. Включение в перебор определяется по сути дела всеми тремя типами застываний, т. е., в частности, переменным горизонтом  $H_x$ ; в программе же этим управляет вспомогательный параметр  $T_x$ . Даже когда траектории заморожены, идет непрерывный пересчет параметра  $T_x$  для всех полей (на которых побывали фигуры) траекторий. Поэтому, когда фигура при спуске по дереву возвращается на  $\alpha_0$ -поле траектории, по значению параметра  $T_x$ , приписанного к этому полю, можно решить вопрос о включении траекторий в перебор.

**1.18. Траектория и зона.** В настоящем разделе мы рассмотрели взаимодействие двух низших ступеней системы управления: траектории и зоны. Эти ступени находятся в состоянии вертикальной соподчиненности. В процессе управления системой, т. е. при переборе ходов, фигуры как элементы системы могут двигаться лишь по построенным к данному моменту траекториям. Другие ходы, разре-

В следах каждого из найденных пучков в разделе «глубина застывания» записывается условное фиксированное число 99. Для застывания пучков более высоких степеней отрицания, связанных с уже застывшими, используется с небольшими поправками процедура, аналогичная застыванию при потере связи с действующей траекторией (см. п. 1.16).

При спуске по дереву, если фигура приходит на  $\alpha_0$ -поле пучка, застывшего в смысле «99», то он и все связанные с ним пучки должны быть разморожены. Соответствующая процедура, аналогичная процедуре застывания при сходе с  $\alpha_0$ -поля, производит поиск пучков,

шенные шахматными правилами, в данной модели игры для них как бы не существуют. Таким образом, движение фигуры подчинено траектории — первой ступени системы управления.

Зону игры образует совокупность траекторий и фигур, объединенных в два лагеря (+) и (—) (по цвету) в борьбе за некоторую фигуру, являющуюся целью игры в этой зоне (или поле, косвенно связанное с атакой на некоторую фигуру). Зона как вторая ступень системы управления оказывает непрерывное воздействие на передвижение фигур по своим траекториям. Это воздействие сводится, в основном, к трем различным процедурам, запрещающим (разрешающим) передвижение фигур по траекториям зоны. По сути это три различных типа застывания траекторий:

а) из-за нехватки времени, отпущенного фигуре для игры в зоне (определяется параметром  $T_x$ );

б) из-за потери связи с действующей траекторией;

в) при сходе фигуры с  $\alpha_0$ -поля траектории при подъеме по дереву.

Организуя и действуя на передвижение фигур, зона вмешивается в функционирование подчиненной ей ступени — траектории. Такое вмешательство объясняется тем, что цель игры в зоне в данный момент перебора не обязательно совпадает с целью игры данной фигуры по некоторой траектории зоны; например, для некоторой траектории отрицания — это контроль  $\alpha_k$ -поля траектории с целью взятия контролируемой фигуры из засады.

Тип траектории присваивается ей более высокой ступенью системы управления. Цель игры по траектории зависит от типа последней. Для комлевой траектории зоны нападения цель — взять  $\alpha_k$ -фигуру. Для траектории контроля (отрицающей или комлевой траектории зоны контроля) цель — уничтожить контролируемую фигуру, поджидая ее в засаде на  $\alpha_{k-1}$ -поле. Траектория блокады не имеет своей цели игры, она полностью подчинена цели игры в зоне, где играет блокируемая фигура.

Итак, иерархия ступеней рассматриваемой системы управления сопровождается иерархией соответствующих целей игры. Отметим здесь, что и сама зона испытывает на себе управляющее воздействие третьей ступени системы управления — МО, которое опять-таки объясняется несовпадением целей игры.

В процессе перебора ходов, действуя на подчиненную ступень системы управления, зона игры реагирует на это воздействие. Управляя перебором ходов, зона сама формируется в процессе перебора — вычисляются траектории отрицания зоны. Таким образом, обратная связь при взаимодействии первой и второй ступеней системы меняет саму структуру системы управления. Такое изменение структуры (формирование зон), направляя перебор, и должно в конечном итоге привести к оптимальному варианту управления этой системой.

## 2. ПЕРЕБОР ХОДОВ В СОВОКУПНОСТИ ЗОН (МО)

Рассмотрим, как происходит перебор ходов, управляемый совокупностью зон. Как известно, цель игры в шахматах для каждой стороны — мат неприятельскому королю. В настоящей модели эта цель заменена новой — выигрышем материала. Каждая из сторон, стремясь к достижению этой цели, ищет лучшую для себя стратегию — вариант игры. Совокупность всех стратегий образует дерево перебора ходов. Таким образом, выбор по минимаксу оптимального варианта в дереве перебора есть выбор оптимальной стратегии в системе управления. В этом разделе мы рассмотрим вопросы формирования дерева перебора, критерии обрыва и оценки вариантов перебора, отсечения ветвей при минимаксе, а также технику реализации перебора в программе «Пионер».

**2.1. Цель игры и обрыв вариантов перебора.** При создании шахматных программ одним из важнейших является вопрос о критериях обрыва и оценки вариантов перебора. Как правило, в существующих программах все варианты перебора обрываются на фиксированной глубине, если они заканчиваются так называемой «спокойной» позицией (не содержащей взятий, шахов и ответов на шахи); иначе варианты продолжают до возникновения спокойной позиции. Заключительные позиции вариантов оцениваются с помощью линейной функции, учитывающей соотношение материала на всей доске и многочисленные позиционные факторы. Такой подход к формированию дерева не позволяет в результате выбрать хороший ход в исходной позиции. Данная методика обрыва вариантов и их оценки не подтверждается и практикой шахматных мастеров.

В настоящей модели, а следовательно, и в программе «Пионер», обрывом вариантов управляет цель игры. Общий критерий обрыва варианта есть критерий достижения цели игры или потери надежды достигнуть эту цель при продолжении варианта. Поскольку рассматриваемая система управления является многоступенчатой, причем ступени имеют разные цели игры, критерии обрыва вариантов, т. е. прекращения игры, формулируются по-разному для каждой ступени.

**2.2. Критерии обрыва вариантов перебора.** Целью игры в зоне является взятие  $\alpha_k$ -фигуры, т. е. фигуры, которая атакована комлевой фигурой по комлевой траектории. Стоимость  $\alpha_k$ -фигуры и есть тот выигрыш, к которому стремится сторона (+) в процессе игры в зоне. Сторона (—) противодействует этому. Ее цель игры — по возможности не проиграть ничего или, по крайней мере, меньше стоимости  $\alpha_k$ -фигуры. В соответствии с этим критерий обрыва варианта есть, в сущности, критерий надежды для каждой из сторон (+) и (—) достичь при продолжении варианта своей цели игры. Итак, вариант игры в зоне обрывается, если выполнено одно из условий:

1)  $\alpha_0$ -фигура уничтожена или заблокирована так, что снятие блокады невозможно;

2)  $\alpha_k$ -фигура уничтожена или отступила с  $\alpha_k$ -поля;

3) потеряно материала ( $m_T$ ) больше, чем можно выиграть с точки зрения стороны (+) или защитить с точки зрения стороны (—), т. е. если  $-cm_T \geq m_K$ , где  $m_K$  — стоимость  $\alpha_k$ -фигуры;  $c=1$ , если в данной позиции ход черных, и  $c=-1$ , если ход белых.

Рассмотрим теперь МО, состоящее из нескольких включенных в игру зон нападения. В МО несколько  $\alpha_0$ - и  $\alpha_k$ -фигур, причем они могут быть разного цвета. Чтобы сформулировать критерий обрыва варианта в МО, надо вычислить суммы стоимостей белых и черных  $\alpha_k$ -фигур, входящих в совокупность зон:  $\Sigma m_w$  и  $\Sigma m_b$ . Критерий надежды здесь имеет вид

4)  $-cm_T \geq \Sigma m_w + \Sigma m_b$  (где  $c$  такое же, как в условии 3).

Итак, при переборе в МО в данном варианте перебора игра прекращается в тех зонах, входящих в МО, в которых выполнены условия 1) или 2), и продолжается в остальных зонах. Во всей совокупности зон вариант обрывается, если выполнено условие 4). Потеря надежды на достижение цели высшей ступенью системы управления диктует прекращение игры всем низшим ступеням, независимо от того, потеряли они надежду достигнуть своих местных целей или нет. Условие 3) (для отдельных зон) не проверяется.

Таким образом, проверка условий обрыва варианта, а следовательно, и вычисление величин из условия 4) должны производиться после каждого хода в процессе перебора вариантов в математическом отображении.

**2.3. Проверка критериев обрыва вариантов.** Выполнение условий 1) и 2) обеспечивается процедурами застывания (размораживания) (см. п. 1.14). Если в некоторой зоне из совокупности зон оказывается выполненным одно из условий (1) или 2)), траектории комлевого пучка застывают, поэтому застывают и траектории отрицания, т. е. перебор в этой зоне прекращается и продолжается в незастывших зонах.

Покажем теперь, как вычисляются суммы  $\Sigma m_w$  и  $\Sigma m_b$  из условия 4). Рассмотрим некоторую комлевою фигуру, входящую в совокупность зон, например фигуру 1 (рис. 30). Пусть, кроме того, просматривая информацию о комлевых пучках траекторий, связанную с полем доски, на котором стоит фигура 1, мы найдем, что ее нападения на фигуры 3, 4 и 5 не застыли. С помощью процедуры вычисления траекторий [3] строим траектории этих пучков. Рассмотрим одну из полученных траекторий, например, 1—2—4—5—7. Просмотрев по очереди ее  $\alpha$ - и  $\beta$ -поля, обнаружим, что на  $\alpha$ -поле 4 стоит фигура 3 стороны (—), причем существует незастывшая траектория нападения на нее фигуры 1 (1—2—4). Следовательно, при движении фигуры 1 по комлевой траектории 1—2—

4—5—7 максимальный выигрыш равен  $m_3 + m_5$ , где  $m_3$  и  $m_5$  — стоимости фигур 3 и 5. Переходя к другим траекториям комлевых пучков фигуры 1, получим, что для траектории 1—2—3—5—7 выигрыш равен  $m_5$ , для 1—2—4—5—6—7 равен  $m_3 + m_5$ , для 1—2—3—5—6—7 равен  $m_4$ . Таким образом, наибольший возможный выигрыш в зонах с комлевой фигурой 1 равен  $\max(m_3 + m_5, m_5, m_3 + m_4, m_4)$ . Предположим, что этот максимум достигается на траектории 1—2—4—5—7 и равен  $m_3 + m_5$ . В специальном массиве  $8 \times 8$  пометим поля, на которых стоят фигуры 3 и 5.

Перейдем к следующей комлевой фигуре совокупности зон — фигуре 2 (рис. 30). Совершив аналогичные операции, получим, что



Рис. 30. Комлевые траектории совокупности зон

максимальный выигрыш в зонах с комлевой фигурой 2 равен  $\max(m_5, m_6)$ . Однако, учитывая, что поле, на котором стоит фигура 5, отмечено в специальном массиве  $8 \times 8$  и, следовательно, фигура 5 исключается из рассмотрения, находим, что максимум равен просто  $m_6$ . Итак, в рассмотренном примере совокупности зон (рис. 30)  $\Sigma m_{(-)} = (m_3 + m_5) + m_6$ ,  $\Sigma m_{(+)} = 0$ .

Текущее значение  $m_T$  также вычисляется при каждом ходе в процессе перебора и равно  $m_T = (M_b - M_w) + m_0$ , где  $M_w(M_b)$  — сумма стоимостей всех белых (черных) фигур, снятых с доски в данном варианте перебора, считая от исходной позиции к текущему ходу в варианте

те;  $m_0$  — соотношение материала в исходной позиции.

Если вариант обрывается, то значение  $m_T$  в заключительной позиции варианта является оценкой этого варианта  $m_K$ . Таким образом, обсуждаемая здесь модель характеризуется отсутствием статической оценочной функции, функция  $m_K$  определена лишь в некоторых узлах дерева перебора, а именно, в заключительных позициях вариантов (критерии определения таких позиций рассмотрены выше). Следовательно, одной из основных задач в данной модели является формирование области определения оценочной функции  $m_K$ . Еще раз подчеркнем, что оценочная функция здесь является не просто оценкой заключительной позиции варианта, она оценивает сам вариант перебора — стратегию в системе управления, т. е. степень достижения цели игры при данной стратегии. Будучи оторванной от варианта, т. е. как соотношение материала в данной позиции, функция не имеет смысла.

Заметим здесь, что оценочная функция программы «Пионер»

имеет и позиционную составляющую, тесно связанную с МО, однако обсуждение этих вопросов выходит за рамки данного приложения.

**2.4. Отсечение ветвей при минимаксе. О методе граней и оценок.** По установившейся терминологии построение узкого дерева перебора, т. е. априорное отбрасывание некоторых ходов в каждой позиции (здесь — ходов не по траекториям), называется *отсечением при движении вперед*. Однако в процессе подъема по дереву при минимаксе в некоторых узлах может возникнуть такая ситуация, что дальнейший спуск из этого узла, т. е. дальнейшее ветвление в нем не изменит результата минимакса. В таком случае ветвление в узле можно прекратить и продолжить подъем по дереву. Тем самым мы отбрасываем некоторые ходы в позиции, соответствующей данному узлу. Такое отсечение ветвей при минимаксе называется *отсечением при движении назад*.

Покажем, как производится отсечение ветвей при минимаксе в данной модели игры. Предположим, что перебор происходит в совокупности зон. Итак, если при минимаксе в некоторый белый узел поднимается концевая оценка варианта  $m_k$ , столь высокая, что противная сторона все равно отсечет ветвь, ведущую сверху к этому узлу, то другие ветви от узла считать не имеет смысла.

Рассмотрим два метода отсечения ветвей, реализованные в данной модели.

Первый из них основывается на следующем соображении. Пусть выше данного белого узла  $N$  по дереву уже было ветвление и минимакс, а следовательно, в некоторых узлах уже есть текущие оценки. Тогда после очередного подъема в данный белый узел и определения в нем текущей оценки  $m_1$  надо последовательно проверить во всех расположенных выше по текущей ветви черных узлах, в которых уже есть текущие оценки, нет ли среди них таких, чтобы выполнялось условие  $m_2 \leq m_1$ . Если есть — ветви из узла  $N$  более не формируются.

Это соображение лежит в основе метода «граней и оценок». Этот метод, называемый иногда  $\alpha$ — $\beta$ -отсечением, используется в большинстве шахматных программ, а также при решении других задач переборного типа. Очевидно, при использовании отсечений большое значение приобретает очередность рассмотрения ходов в переборе. Доказано [8], что при теоретически наилучшей очередности перебора и размере дерева  $a^n$  ( $a$  — фиксированное количество ходов в каждой позиции,  $n$  — глубина перебора в полуходах) метод ветвей и границ уменьшает перебор до величины, оцениваемой как  $a^{(n+1)/2}$ . Реально, в существующих шахматных программах полного перебора на фиксированную глубину дерево перебора намного превышает теоретический минимум, достигая  $10^5$  полуходов при глубине  $n=5$  полуходам. Практически с увеличением глубины на один полуход дерево перебора вырастает в семь раз (в середине пар-

тии). Программы с таким деревом перебора предъявляют повышенные требования к быстродействию ЭВМ. Причем удовлетворение этих запросов за счет сверхмощных машин вовсе не ведет к существенному улучшению игры, поскольку мастерского уровня игры нельзя достигнуть, увеличив глубину перебора *всех* вариантов на 1—2 полухода. Здесь мы отвлекаемся от качества оценочной функции и цели игры. В переборных задачах, где в каждой ситуации количество возможностей  $A$  много больше числа ходов в шахматной позиции ( $A \gg a$ ), практическая ценность полного перебора на фиксированную глубину даже с использованием метода граней и оценок чрезвычайно мала из-за астрономических размеров дерева перебора.

Вернемся к описанию модели игры. Метод граней и оценок как таковой здесь не используется, так как установление границ изменения оценочной функции до начала перебора повлекло бы за собой искажения при формировании МО. Однако изложенное выше соображение, лежащее в основе метода, применяется и здесь.

В процессе перебора в памяти ЭВМ хранится текущая (наивысшая в дереве) глубина узла, уже имеющего текущую оценку. При каждом подъеме в некоторый узел и определении текущей оценки программа производит подъем по дереву и сравнение оценки лишь до упомянутой текущей глубины.

Перейдем к рассмотрению второго метода отсечения ветвей, характерного лишь для данной модели.

## 2.5. Отсечение ветвей по критерию наихудшего результата.

В каждом узле мы можем определить два числа — «цену» цели и «цену» варианта; сравнение этих цен дает ответ на вопрос о необходимости формирования или отсечения других ветвей, отходящих от данного узла.

Поскольку мы рассматриваем совокупность зон, цена цели может быть разной у белых и черных, поэтому и решение о формировании новых ветвей может быть разным в зависимости от того, какая сторона выбирает ход в данном узле. Пусть в данном узле ход выбирают белые. Введем понятие наихудшего результата: назовем наихудшим результатом для данного узла такое  $m_k$ , когда одна сторона в текущем оптимальном варианте игры в совокупности зон ничего не выиграла и проиграла все, что могла. Для черных это будет проигрыш всех конечных стоимостей  $\alpha_k$ -фигур, т. е.  $\Sigma m_b$ . Итак, когда в белом узле появляется концевая оценка  $m_k$ :

$$m_k \geq m_T + \Sigma m_b,$$

другие ветви от белого узла считать нечего, так как минимум все равно приведет к исключению этого узла из оптимального варианта перебора. Аналогично, для черного узла условие отсечения:

$$-m_k \geq m_T + \Sigma m_w.$$



Проверка условий отсечения не представляет сложности. Для этого после каждого хода назад при подъеме по варианту надо воспользоваться процедурой вычисления  $\Sigma m_w$  и  $\Sigma m_b$  в совокупности зон, которая уже применялась при проверке критериев обрыва вариантов.

Интересно отметить, что существование такого отсечения ветвей при минимаксе в практике шахматных мастеров и необходимость его реализации в данной модели были выявлены лишь в процессе эксперимента с работающей программой перебора ходов в зоне — первым вариантом программы «Пионер».

Модель содержит и другие отсечения при «движении назад», необязательно приводящие к прекращению ветвления в данном узле, однако значительно сокращающие количество рассматриваемых ходов. Речь идет об отсечении зон и о прекращении ветвления в пучке (см. п. 2.10).

**2.6. Приоритет ходов в переборе.** Выигрыш от использования отсечения ветвей при минимаксе, т. е. значительное сокращение дерева перебора, можно получить лишь в том случае, если при спуске по дереву заранее определить, какую ветвь в данном узле формировать в первую очередь с тем, чтобы после подъема в этот узел решить вопрос об отсечении остальных ветвей. В соответствии с этим установлена следующая очередность рассматриваемых ходов во включенных в игру зонах:

1. Взятия соответствуют основной цели игры в данной модели, поэтому они имеют первый по порядку приоритет. Чем выгоднее взятие, тем отсечение более вероятно, поэтому в первую очередь рассматриваются взятия, при которых стоимость взятой фигуры, а при равенстве — разность между стоимостью взятой фигуры и стоимостью той фигуры, которая бьет, будет наибольшей.

Для реализации рассмотренного принципа необходимо быстрое распознавание взятий среди прочих ходов по траекториям МО. Отметим, что взятиям всегда соответствуют некоторые траектории зон. Упомянутое распознавание осуществляется с помощью процедуры «хода не по МО», формирующей (по шахматным правилам) все ходы данной фигуры, а не только ходы по траекториям МО. Основой этой процедуры является операция наложения массива  $8 \times 8$  на соответствующий массив  $15 \times 15$  [3] и определения полей в массиве  $8 \times 8$ , в которых запишется единица.

Опишем процедуру определения выгоднейших взятий. Рассмотрим некоторую фигуру той стороны, которая выбирает ход в данном узле дерева перебора. Пусть фигура принадлежит зоне, в которой происходит перебор. С помощью процедуры хода не по МО определяем все ее взятия. Выделим одно из них, т. е. пару фигур со стоимостями  $m_1$  (взятая) и  $m_2$  (атакующая) и координаты этих фигур. Затем выясняем, совершался ли уже этот ход (взятие) раньше в данном узле в процессе перебора (см. ниже). Если совершался, переходим к следующему взятию, если нет, то переменным  $X_1$  и  $Y_1$  присваиваем координаты атакующей и взятой фигур соответственно, а переменной  $D_{\max}$  разность  $300m_1 - m_2$ . Переходим к следующему взятию, определяем новые  $m_1$  и  $m_2$ . Если  $300m_1 - m_2 > D_{\max}$ , запоминаем новые значения  $X_1$ ,  $Y_1$  и  $D_{\max}$ . Если нет, переходим к следующему взятию и т. д. В результате, просмотрев все взятия в зоне, мы найдем наилучшее взятие — ход  $X_1 - Y_1$ .

2. Формировать зону, т. е. строить новые траектории отрицания, выгодно целыми областями, так как если здесь будет получен ответ о результатах борьбы

в зоне, то другие возможные области зоны не будут сформированы (в результате обрыва вариантов и отсечения при минимаксе). Таким образом, перебор и МО будут меньше. Итак, второй приоритет имеют ходы по таким траекториям, которые в последний момент включены в МО.

Этот принцип можно реализовать, если специальным образом пометить вновь построенные пучки траекторий и после возврата по варианту при псевдопереборе воспользоваться этими метками при решении вопроса о том, по каким пучкам траекторий двигаться в первую очередь. После того, как найдены новые траектории, метки на старых должны быть стерты.

3. Предположим, что псевдоперебор (см. п. 1.10) привел к определению новых траекторий отрицания. Происходит возврат по дереву к некоторому узлу с попутным стиранием ветвей дерева перебора. После этого перебор должен начаться вновь из данного узла с включением в игру новых траекторий.

Примем, что передвижения той фигуры, которая была «атакована» в смысле псевдоперебора, должны быть такими же, как и в предыдущем варианте перебора, том варианте, который уже стал предварительным и стерт в памяти ЭВМ. Поэтому возвратное движение атакующей фигуры запоминается в соответствующих следах, привязанных к  $\alpha$ -полям, которые фигура проходит при возврате по варианту. Таким образом, при переборе (вперед) появляется возможность повторить это движение, так как ходы данной фигуры на отмеченные  $\alpha$ -поля имеют приоритет.

4. Шахматное время (в полуходах) играет важную роль в обсуждаемой модели игры, поэтому быстрее достижение цели может быть обеспечено передвижением фигуры по наиболее короткой траектории. Ходы по таким траекториям имеют четвертый по порядку приоритет. При равенстве длин предпочтение отдается вилочной траектории, т. е. траектории, часть которой является общей для нескольких траекторий. Примером вилочных траекторий служат траектории 1—2—3—5—7 и 1—2—4—5—6 на рис. 30.

На данном этапе работы этот принцип в общем виде программно не реализован, однако запрограммирован некоторый частный случай. Если программа принимает решение о движении фигуры по некоторому пучку траекторий, то в первую очередь движение происходит по кратчайшим (вилочным) траекториям данного пучка (см. ниже). Сравнение вилочности траекторий различных фигур будет выполняться в следующем варианте программы «Пионер».

В результате эксперимента с программой «Пионер» выяснилось, что рассмотренная система приоритетов требует уточнения. Прежде всего в ней не учтена, если можно так выразиться, «степень важности» зон. Речь идет о стоимости цели игры в данной зоне, расстоянии по траектории от атакующей фигуры до атакующей, а также о проходимости этой траектории. Говоря о проходимости, мы имеем в виду следующее. Рассмотрим все  $\alpha$ -поля некоторой траектории, определим фигуры, находящиеся на расстоянии одного передвижения от данного  $\alpha$ -поля. Для каждого  $\alpha$ -поля определим результат оптимального размена на нем для этих фигур с учетом того, что фигура, двигаясь по траектории, пришла на это поле.

$\alpha$ -поля, на которых размен оказался в пользу стороны, которой принадлежит данная траектория, называются проходимыми. Зона с комлевой траекторией, у которой все  $\alpha$ -поля проходимы, называется уязвимой зоной. Естественно, игра в таких зонах должна иметь высокий приоритет. При этом взятия потеряют наивысший приоритет.

Эти принципы будут учтены в следующем варианте программы (см. также приложение 2).

**2.7. Анализ траекторий пучка для включения в перебор.** Как уже отмечалось, если в процессе перебора вариантов программа для некоторой фигуры выбирает пучок траекторий для включения в перебор, то в первую очередь рассматриваются ходы по кратчай-

шим (вилочным) траекториям пучка. Для этого из соответствующего следа, привязанного к полю, на котором стоит фигура, извлекается информация о данном пучке. Выясняется, не застыл ли этот пучок. Если нет, то с помощью процедуры вычисления пучка траекторий он разворачивается (см. п. 1.5), т. е. все траектории пучка строятся в явном виде на свободной от фигур доске. Далее каждая из траекторий по очереди переводится на реальную доску, строясь все ее  $\alpha$ - и  $\beta$ -поля, вычисляется ее реальная длина. Здесь же решается вопрос о том, есть ли время для передвижения данной фигуры по этой траектории зоны. Для этого надо лишь срав-

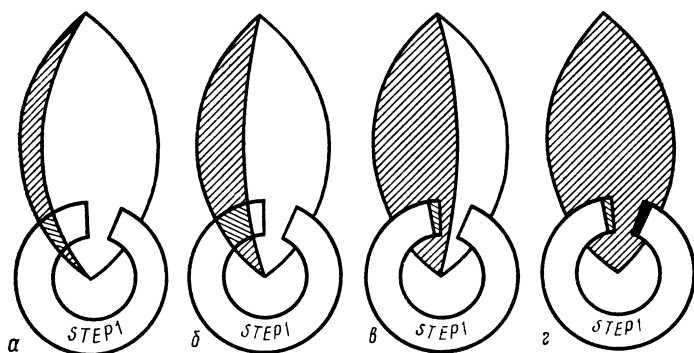


Рис. 31. Анализ траекторий пучка по признаку их длины

нить текущее значение параметра  $T_x$  для данного пучка (см. п. 1.12) с реальной длиной траектории. Если траектория заблокирована своей фигурой, выясняется возможность и необходимость ее деблокады. Если же заблокирован участок траектории  $[\alpha_0, \alpha_1]$ , то, естественно, эта траектория в перебор не поступает, однако с полем, на котором стоит блокирующая фигура, связывается информация о необходимости деблокады.

На рис. 31 изображен массив полей доски STEP1, на которые фигура может попасть за один ход. Массив STEP1 изображен в виде разомкнутого кольца — это означает, что некоторые ходы данной фигуры в текущем узле уже делались в процессе перебора вариантов. Вопрос определения таких ходов рассмотрен ниже. Предположим, что мы проанализировали некоторую траекторию пучка и выяснили, что ее по всем параметрам можно включить в перебор. В таком случае отмечаем ее  $\alpha$ -поле в массиве STEP1 (очевидно, оно там содержится). Длину этой траектории ( $A$ ) запоминаем. Предположим, что несколько следующих по порядку траекторий также можно включить в перебор и они имеют такую же длину.

Отмечаем их  $\alpha$ -поля в массиве STEP1. Это иллюстрируется рис. 31, а, б.

Так, переходя от одной траектории к другой, мы можем прийти к траектории, длина которой на реальной доске меньше  $A$ . Тогда стираем в массиве STEP1 все ранее отмеченные  $\alpha_1$ -поля и отмечаем  $\alpha_1$ -поле новой траектории (рис. 31, в). Переменной  $A$  присваиваем новое значение длины текущей кратчайшей траектории. Продолжаем этот процесс, причем, если очередная траектория имеет длину больше  $A$ , ее  $\alpha_1$ -поле в массиве STEP1 не отмечается, если — меньше, происходит рассмотренная выше очистка массива STEP1

и изменение текущего значения  $A$ . В результате за один просмотр пучка в массиве STEP1 окажутся отмеченными  $\alpha_1$ -поля всех кратчайших траекторий пучка, имеющих одинаковую длину (31, з). Остается выбрать те из них, которые принадлежат вилочным траекториям.

**2.8. Анализ траекторий пучка по признаку вилочности.** Обратимся к анализу остальных пучков данной фигуры, входящих в зону. Однако в них мы будем рассматривать лишь те траектории,  $\alpha_1$ -поля которых отмечены уже в массиве STEP1 после анализа основного пучка, выбранного для включения в перебор. Итак, пусть после анализа основного пучка на отмеченных полях массива STEP1 записаны единицы. Если некоторая траектория другого пучка дан-

ной фигуры имеет  $\alpha_1$ -поле, отмеченное в массиве STEP1, то этот элемент массива STEP1 увеличиваем на единицу. После того, как мы просмотрим все остальные пучки данной фигуры, на полях массива STEP1 будут записаны различные натуральные числа, каждое из которых показывает количество пучков траекторий данной фигуры, проходящих через данное поле, т. е. степень вилочности этого  $\alpha_1$ -поля основного пучка. Остается просмотреть все элементы массива STEP1 и найти наибольший. Его номер и будет координатой самого вилочного  $\alpha_1$ -поля кратчайших траекторий основного пучка. Следовательно, в переборе будет сделан ход  $\alpha_0$ — $\alpha_1$ . Эта процедура иллюстрируется рис. 32.

**2.9. Отступление и деблокада.** Везде в этом приложении мы называли пучком совокупность траекторий некоторой фигуры, имеющих по крайней мере два общих поля — начальное и конечное. В данной модели определен еще один тип пучка. Назовем *пучком траекторий отступления* (деблокады) совокупность всех траекторий в одно передвижение, имеющих общее начальное поле  $\alpha_0$ , а конечные поля — все поля доски, куда данная фигура может по пра-

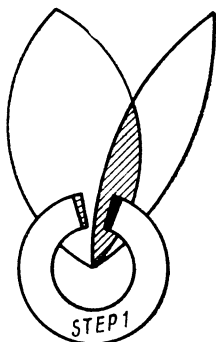


Рис. 32. Анализ траекторий пучка по признаку вилочности

вилам попасть за один ход с поля  $\alpha_0$ . Критерии включения в перебор таких пучков, а также формирование соответствующих зон в настоящем приложении не рассматриваются.

Список параметров пучка отступления несколько отличается от стандартного списка. Обратимся к рис. 33. След пучка отступления с поля  $\alpha_0$  привязывается к этому полю. Во второй позиции следа вместо  $\alpha_0$ -поля записывается нуль. В третьей позиции — единица. В четвертой — тип пучка: отступления или деблокады. В пятой позиции, используемой, в основном, в процедурах застывания —

1	Номер фигуры	
2	0	
3	1	
4	Отступление или деблокада	
5	Номер атакующей (или деблокируемой) фигуры	
6	Глубина застывания	
7	0	Поле, ближай- шее к блокированному
8	0	Поля, запре- щенные для деблокады

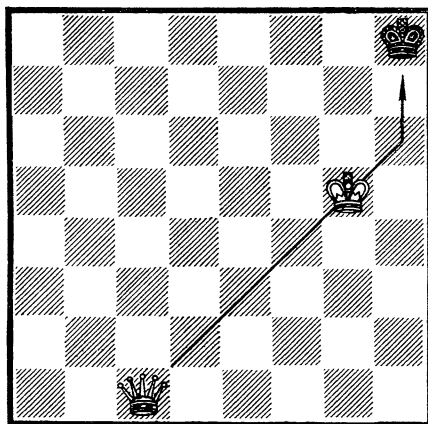


Рис. 33. След пучка отступления (де-блокады)

Рис. 34. Пример блокированной траектории

размораживания и при пересчете  $T_x$ , мы указываем номер фигуры, с траекториями которой был связан данный пучок в момент его появления. В седьмой и восьмой позициях следа пучка отступления записаны нули. Для пучка деблокады в седьмой позиции записана координата ближайшего (по траектории) к блокированному полю  $\alpha$ -поля, блокированной траектории, т. е.  $\alpha$ -поле, попадая на которое, фигура как бы упирается в блокаду.

Например, на рис. 34 траектория  $Fc1-h6-h8$  блокирована королем на g5. Для пучка деблокады поля Z(g5) в седьмой позиции (см. рис. 33) записана координата поля c1. При деблокаде существенно уйти с траектории так, чтобы снова ее не заблокировать. Поэтому в пучке деблокады, в отличие от пучка отступления, разрешены не все ходы; например, на рис. 34 надо запретить траектории  $Krg5-f4$  и  $Krg5-h6$  в соответствующем пучке деблокады. Для этого в восьмой позиции следа записываются координаты по-

лей  $f_4$  и  $h_6$  — соседних с блокированным  $\alpha(\beta)$ -полем траектории. Остальные позиции следа не отличаются от общего случая.

Пучки отступления (деблокады) могут быть лишь комлевыми пучками зоны, поэтому все процедуры в зоне, рассмотренные в первом разделе настоящего приложения, совершенно аналогично выполняются для зон с комлевой траекторией отступления (деблокады). Для выполнения хода по траекториям таких пучков в переборе, естественно, не нужно пользоваться процедурой вычисления пучков траекторий [3]. Достаточно обратиться к процедуре хода не по МО, которая упоминалась при описании быстрого определения выгодных взятий (с. 95).

Заметим, что при выполнении деблокады перед использованием процедуры хода не по МО программа временно «ставит» на поля доски, записанные в восьмой позиции следа (см. рис. 33), некие фигуры своего цвета, а после применения процедуры «убирает» их. Это приводит к тому, что процедура хода не по МО трактует соответствующие лучи как заставленные своими фигурами, а ходы в этих направлениях — как запрещенные правилами игры (на рис. 34 это лучи  $g_5-h_6$  и  $g_5-c_1$ ). Идея такого использования хода не по МО принадлежит А. И. Резницкому. Вилочность, а следовательно, и приоритет включения траекторий пучка отступления (деблокады) учитывается аналогично тому, как описано выше.

Как уже отмечалось, вопрос о необходимости и возможности деблокады решается при анализе пучков траекторий для включения в перебор. Каждая текущая кратчайшая блокированная траектория анализируется для выяснения возможности ее деблокады. В этом анализе также участвует процедура хода не по МО. Если выясняется, что деблокада невозможна, блокированная траектория не считается текущей кратчайшей, и процедура переходит к анализу следующей траектории пучка. В противном случае запоминается информация, необходимая для деблокады (параметры следа пучка деблокады). Присоединение этого следа к цепному списку ячеек, связанному с блокированным полем, произойдет по окончании анализа траектории пучка, если выяснится, что данная траектория действительно является кратчайшей и выбрана для исполнения хода (в переборе).

Итак, множество пучков траекторий вместе с пучками отступления (деблокады) полностью охватывает совокупность подсистем низшего уровня — первую ступень системы управления.

**2.10. Отсечение при ветвлении в пучке траекторий.** Отсечения, рассмотренные выше в этом разделе, основаны на анализе достижения целей игры во второй и третьей ступенях системы управления (в зоне и во всем МО). Здесь мы рассмотрим отсечение, которое основано на анализе достижения цели в первой ступени системы управления (пучке траекторий). Соответственно и результат этого отсечения влияет лишь на перебор ходов в данном пучке тра-

екторий. Речь идет о прекращении рассмотрения (в данном узле дерева перебора) ходов по траекториям данного пучка, а не о полном прекращении ветвления в узле и подъеме из него.

Итак, предположим, что в процессе перебора мы оказались в некоторой позиции. Модель выбирает в некотором пучке траекторию, чтобы сделать по ней следующий ход. Пусть, кроме того, в переборе уже рассматривался выходящий из данной позиции вариант, в котором фигура двигалась по одной из траекторий данного пучка. Если при этом фигура дошла до  $\alpha_k$ -поля пучка, либо сошла с данной траектории (уйдя в другую зону), либо же простояла на этой траектории до конца варианта (не будучи заблокированной), то движение по другим траекториям пучка в данной позиции не имеет смысла, и модель запрещает это движение. Если же фигура была потеряна на траектории пучка либо заблокирована, то движение по другой траектории разрешено. Таким образом, запрет на ветвление в пучке зависит от того, достигнуто ли  $\alpha_k$ -поле пучка и, если нет, есть ли надежда на его достижение.

Для пучков деблокады критерий упрощается: если в данной позиции была хотя бы одна деблокада и деблокирующая фигура на протяжении соответствующего варианта не была потеряна на поле деблокады, другие деблокады в этой позиции мы не рассматриваем.

Для пучков отступления ветвление разрешено всегда в связи с необходимостью отыскания безопасных полей отступления и возможностью продолжения атаки.

Критерий отсеечения в программе «Пионер» проверяется посредством анализа последней из построенных ветвей дерева ниже данного узла и изучения следов данной фигуры на полях доски, где она побывала в соответствующем варианте. Трудность здесь состоит в следующем: в момент анализа пучок содержится во включенной\* в игру зоне, однако возможна такая ситуация, что это включение произошло «только-что», и в изучаемом варианте фигура могла двинуться по траекториям выключенной зоны, выключным с какими-то траекториями включенных зон. В этом случае считается, что движения по траекториям пучка еще не было, и критерий разрешает ветвление. Для выделения таких ситуаций в списке параметров следа пучка отведена двенадцатая позиция (см. рис. 23). Если фигура сделала в переборе ход  $X—Y$  по траектории некоторого пучка во включенной зоне, с полем  $Y$  связывается след этого пучка, причем в 12-й позиции следа записывается адрес узла дерева перебора, соответствующего позиции, в которой был сделан ход  $X—Y$  (см. п. 2.11). Если же ход в пучке сделан по «вилочности», а сам пучок принадлежит выключенной зоне, в 12-й позиции следа записывается нуль.

---

\* Включение — выключение зон здесь не рассматривается.

Необходимость в отсечении при ветвлении в пучке была выявлена в процессе эксперимента с программой «Пионер» (при решении этюдов).

**2.11. Структура дерева перебора.** Информация о дереве перебора используется на протяжении всего перебора и минимакса. «Пионер», в отличие от большинства созданных к настоящему времени шахматных программ, хранит в оперативной памяти ЭВМ все дерево, в том числе ветви, соответствующие текущим оптимальным вариантам. Это оказалось необременительным для машины, поскольку ни в одном из экспериментов размер дерева не превысил 200 полуходов. Структура дерева должна обеспечивать спуск и подъем по дереву, т.е. восстановление при подъеме в каждом узле соответствующей позиции, пересчет текущего соотношения материала  $m_t$ , минимакс оценок вариантов, стирание некоторых ветвей при возврате по варианту при псевдопереборе, ликвидацию части дерева при исполнении хода на доске, «движение по рельсам».

Всем этим требованиям удовлетворяет структура цепного списка. Узлы дерева нумеруются последовательно в порядке их формирования в процессе перебора вариантов. Номер узла задает адрес, по которому записана информация о дереве перебора, связанная с этим узлом (рис. 35,а). Номер узла совпадает с номером столбца в двумерном массиве, в котором записана информация об узле. Структура стандартного столбца изобразжена на рис. 35,б. В нем записаны, в частности, адреса предшествующего «родительского» узла, а также соседнего узла — «сына того же родителя». Данный узел получается из родительского после хода  $X \rightarrow Y$ . Если ход сопровождается взятием или превращением пешки, информация об этом записывается в соответствующий столбец.

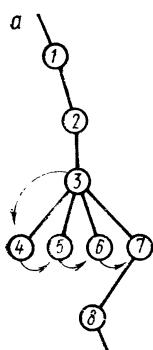
Рассмотренная структура дерева перебора позволяет для каждого узла в любой момент перебора установить, какие ветви, исходящие из данного узла, уже построены, т.е. какие ходы в данной позиции уже делались в процессе перебора. Для этого достаточно, зная номер данного узла, например 3 (см. рис. 35,а), увеличить его на единицу — это даст нам адрес «сына». По этому адресу считываем информацию о ходе, приводящем в этот узел (ход  $X \rightarrow Y$ ). Записываем этот ход в специальный массив. В том же столбце находим адрес «брата» (в примере на рис. 35,а это узел 5). Переписываем из соответствующего столбца информацию о ходе в специальный массив и т.д. Таким образом доходим до узла 7, у которого нет «брата». В специальном массиве мы получим список ходов, которые уже делались в данной позиции в процессе перебора.

При формировании дерева перебора позиции, соответствующие построенным узлам дерева, не хранятся в памяти ЭВМ. Имеется лишь текущая позиция, соответствующая узлу, который изучается



в данный момент. Поэтому при возврате в процессе перебора и минимакса в некоторый узел необходимо восстановить соответствующую ему позицию. Это выполняется процедурой, синхронно с каждым ходом (вперед или назад) изменяющей текущую позицию при переборе вариантов. Эта же процедура пересчитывает текущее соотношение материала  $m_t$ . Во время работы она использует информацию, привязанную к узлам дерева перебора.

Как уже отмечалось, в процессе работы программы необходимы частые перестройки дерева перебора. Они состоят в удалении из



б

Адрес „дврата“
Адрес „родителя“
X
Y
Номер взятой фигуры
Признак превращения пешки и взятия на проходе
Метка зоны
Оценка

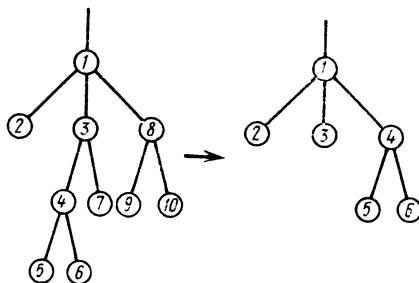


Рис. 35. Структура дерева перебора      Рис. 36. Перестройка дерева перебора

дерева уже построенных отдельных ветвей или целых поддеревьев. Принятая структура дерева позволяет производить такие перестройки путем перенумерации узлов дерева. Пример перестройки показан на рис. 36. Здесь необходимо удалить поддерево, содержащее узлы 4, 5, 6, 7. В сущности, перестройка дерева состоит в том, что программа переписывает содержимое некоторых столбцов упомянутого выше двумерного массива в другие столбцы.

**2.12. Новое содержание известных процедур.** Процедуры, рассмотренные в первом разделе настоящего приложения, являются новыми по сравнению с другими известными моделями шахматной игры. Во втором разделе собраны описания формально более или менее стандартных операций, выполняемых в большинстве шахматных программ и, вообще, при решении многих задач переборного типа. Мы имеем в виду перебор ходов, обрыв и оценку вариантов, минимакс с отсечениями и приоритет ходов в переборе. Но в данной модели, рассматривающей перебор как поиск оптимальной стратегии в трехступенчатой многоцелевой системе управления, упомянутые операции наполнены совершенно новым содер-

жанием. Это подтверждается, в частности, размерами деревьев перебора, полученными в эксперименте (порядка  $10^2$  полуходов), да и самим фактом нахождения решения в этюдах, где длина решающего варианта достигает 25 полуходов.

Применение таких же по форме операций в известных шахматных программах не дает соответствующих результатов. Это связано с тем, что ни одна из этих программ, насколько известно, не моделировала игру как многоступенчатую систему управления, с неточной целью игры, а в одноступенчатом виде, задаваемом правилами игры (цель игры — мат), эта система трудноуправляема из-за астрономических размеров дерева вариантов при поиске оптимальной стратегии. Причем эффект от применения всей совокупности вышеупомянутых процедур в рамках одноступенчатой системы при отсутствии неточной цели игры вряд ли оправдывает усилия, потраченные на оптимальное программирование переборных алгоритмов.

В обсуждаемой здесь модели роль процедур, описанных во втором разделе, не ограничивается лишь уменьшением размеров дерева перебора. Воздействуя на перебор, они оказывают влияние на взаимодействие ступеней системы управления — формирование зон (раздел 1), включение зон в перебор, — обеспечивая тем самым обратную связь подсистем. В частности, обрыв вариантов (по критериям) и их оценка с помощью оценочной функции являются одним из важнейших видов обратной связи системы управления с перебором: цель игры отвечает на вопрос — к чему мы стремимся, оценочная функция способна свидетельствовать, насколько успешно цель игры реализуется.

## ПРИЛОЖЕНИЕ 2

### ПОЗИЦИОННАЯ ОЦЕНКА И ПРИОРИТЕТЫ

**М. А. Цфасман, Б. М. Штильман**

В шахматных программах понятие позиционной оценки играет важную роль. В программах полного перебора на фиксированную глубину с отсеечениями по методу ветвей и границ позиционная составляющая оценочной функции — наиболее сложная ее часть. Как известно, оценочная функция в таких программах — это многочлен первой степени от нескольких переменных, первый член которого пропорционален соотношению материала в текущей позиции, а остальные члены, зависящие от таких факторов, как владение центром, пешечная структура и т.п., образуют в сумме позиционную составляющую. Совершенствование известных программ во многом

определяется совершенствованием позиционной составляющей оценочной функции путем включения в нее новых шахматных факторов.

Такой подход к позиционной оценке является статистическим: факторы, положительные для большинства позиций, в некоторых случаях оказываются отрицательными, и эти позиции получают неверные оценки. Предложенная М. Ботвинником позиционная оценка, основанная на контроле полей траекторий, с одной стороны, универсальна для всех позиций, с другой, — индивидуальна для каждой из них, поскольку вычисляется лишь на полях, охваченных математическим отображением (МО), а оно, в свою очередь, индивидуально для каждого узла в дереве перебора. Итак, в настоящей модели позиционная оценка пропорциональна отношению  $K_w/K_b$ , где  $K_w$  и  $K_b$  — число  $\alpha$ -полей незастывших траекторий включенных в игру зон, контролируемых белыми и черными соответственно. Далее уточняется, как эти  $\alpha$ -поля выбираются.

Вычисление позиционной оценки связано, в основном, с решением двух задач. Прежде всего надо определить незастывшие траектории фигур и отметить их  $\alpha$ -поля. Затем для каждой траектории и для каждого ее  $\alpha$ -поля определить список фигур, находящихся на расстоянии одного передвижения от этого  $\alpha$ -поля, и вычислить результат оптимального размена этих фигур на данном поле. Если данное  $\alpha$ -поле проходимо, т. е. результат размена в пользу той стороны, которой принадлежит данная траектория, например, белых, то  $K_w$  увеличивается на единицу. Аналогично для черных. Анализ  $\alpha$ -полей траекторий производится до первого непроходимого  $\alpha$ -поля.

Поскольку в программе «Пионер» информация о траекториях хранится в упакованном виде, точнее, в виде следов пучков траекторий (см. приложение 1), для определения  $\alpha$ -полей пучки необходимо «развернуть», т. е. извлечь информацию о пучках из их следов, а затем применить процедуру вычисления траекторий, воспользовавшись этой информацией как входными параметрами. Разворачивание всех незастывших пучков траекторий включенных зон — операция, требующая значительного времени. Казалось бы, можно избежать ее выполнения в каждом узле дерева перебора, если учесть, что позиция в варианте перебора с каждым ходом меняется «вяло», а следовательно, и позиционную оценку можно не вычислять заново в каждом узле дерева, а лишь корректировать от хода к ходу в варианте.

Однако на самом деле позиционная оценка имеет гораздо большую изменчивость, поскольку такая изменчивость характеризует МО, которое может существенно меняться не только от узла к узлу в дереве перебора, но и в одном и том же узле в разные моменты перебора (например, за счет включения новых зон). Это, в ча-

стности, подчеркивает индивидуальность принятой позиционной оценки для каждой позиции.

Следовательно, избежать операции разворачивания пучков в каждом узле дерева перебора не удастся. В таком случае было бы желательно совместить вычисление позиционной оценки с другими операциями, выполняемыми программой в каждом узле. Воспользуемся для этого процедурой выбора хода в соответствии с приоритетом; рассмотрим эту процедуру.

В прежнем варианте программы «Пионер» данная процедура состояла из многократных (по числу приоритетов) разворачиваний пучков траекторий для поиска траектории с данными признаками (см. приложение 1). Настоящий вариант программы «Пионер» предусматривает однократное разворачивание всех пучков, вычисление приоритета каждой траектории и затем выбор траектории с наивысшим приоритетом. Имеется в виду следующее. Для данной фигуры определяются все пучки, принадлежащие включенным в игру зонам. Каждый пучок по очереди разворачивается, определяется приоритет (см. ниже) каждой его траектории (по которой в данной позиции фигуры еще не двигались). Число, характеризующее этот приоритет, записывается на  $\alpha_1$ -поле данной траектории в специальном массиве  $8 \times 8$ . Если при разворачивании следующего пучка данной фигуры выясняется, что  $\alpha_1$ -поле некоторой траектории совпадает с полем, помеченным в упомянутом выше специальном массиве  $8 \times 8$ , это значит, что данная траектория — вилочная с некоторой траекторией предыдущего пучка. Поэтому их приоритеты увеличиваются, т.е. в массиве  $8 \times 8$  записывается наибольший из приоритетов, увеличенный на единицу. После того, как мы по очереди развернули все пучки данной фигуры, в массиве  $8 \times 8$  оказались отмеченными некоторые поля (на них записаны натуральные числа). Выбираем из этих полей то, на котором записано наибольшее число. Запоминаем три числа:  $X$  — координату данной фигуры,  $Y$  — координату выбранного поля,  $V$  — число, записанное на этом поле.

Очищаем (присваивая нули) массив  $8 \times 8$  и повторяем рассмотренные выше операции для следующей фигуры с тем же массивом  $8 \times 8$ . Если число, записанное на выбранном поле массива, больше  $V$ , запоминаем новые  $X$ ,  $Y$ ,  $V$ ; если — нет, просто переходим к следующей фигуре. В результате, после обхода всех фигур данного цвета, получим три числа:  $X$ ,  $Y$ ,  $V$ , задающие поля, с которого и на которое должен быть сделан ход, имеющий в данной позиции наивысший приоритет ( $V$ ). Такая система позволяет проводить взаимное сравнение приоритетов траекторий как для одной фигуры, так и для различных фигур, что было невозможно в предыдущих версиях программы «Пионер». Кроме того, при вычислении приоритета траектории можно учесть различные признаки: значимость зоны, которой принадлежит эта траектория (уязви-

мость мишени, ее стоимость), уязвимость самой траектории, длину траектории, ее вилочность.

Вернемся к вычислению позиционной оценки и укажем, как эта процедура включается в рассмотренный выше механизм выбора хода по приоритету. После того, как вычислена текущая траектория пучка со всеми ее  $\alpha$ - и  $\beta$ -полями на реальной шахматной доске, нужно перебрать ее  $\alpha$ -поля, начиная с  $\alpha_0$ -поля, и применить для каждого из них процедуру оптимального размена на поле вплоть до первого непроходимого  $\alpha$ -поля. Легко понять, что во всей совокупности описанных здесь процедур процедура оптимального размена является самым внутренним циклом. Это, естественно, предъявляет повышенные требования к ее эффективности с точки зрения использования ресурсов времени.

Процедура вычисления результата оптимального размена на заданном поле (траектории данной фигуры) организована следующим образом (фигуры одного цвета с данной назовем «своими», противоположного цвета — фигурами «противника»).

Прежде всего находятся все фигуры, которые могут принять участие в размене на этом поле. Фигура, принимающая участие в размене, должна иметь траекторию длиной в одно передвижение к данному полю. При этом траектория должна быть свободна или же заблокирована фигурами, также принимающими участие в размене. Заметим, что такие блокады могут быть только следующих типов: слоны и ферзи или же ладьи и ферзи могут блокировать друг друга, а пешка может блокировать слонов и ферзей.

Поэтому есть возможность, просматривая диагонали, вертикали и горизонтали, а также поля, на которые может пойти конь с данного поля размена, определить сразу все фигуры, принимающие участие в размене. Одновременно определяются и все фигуры, блокирующие любую из перечисленных. Естественно, просмотр каждой из диагоналей, вертикалей и горизонталей обрывается, как только на них встретилась фигура, не принимающая участия в размене.

Существенно, что нигде в процедуре размена не приходится вычислять траектории — это должно резко экономить время работы программы.

Итак, все фигуры, которые могут принять участие в размене, найдены. Теперь используется только список этих фигур, а про реальную позицию на доске можно «забыть».

Нас интересует порядок, в котором фигуры могут вступать в размен. Оптимальным с точки зрения результата размена является расположение фигур каждого цвета по возрастанию стоимости, с учетом того, что данная фигура может принять участие в размене только после тех фигур, которые участвуют в размене и ее блокируют.

Докажем, что не существует другого расположения фигур, улучшающего упомянутый результат размена. Рассмотрим какое-либо иное расположение фигур. Пусть оно оптимально с точки зрения размена и отлично от расположения по возрастанию. Выберем, например, две соседние по списку одноцветные фигуры, такие, что стоимость второй из них меньше стоимости первой. Теперь сравним этот вариант размена с таким, в котором эти фигуры переставлены.

Итак, пусть вначале произведено взятие фигурой с большей стоимостью. Противник может оборвать размен, тогда результат тот же, как если бы мы брали фигурой с меньшей стоимостью. Если противник продолжит размен, мы можем оборвать его — при этом проигрывается разность стоимостей переставляемых фигур по сравнению с таким же обрывом при первоначальном расположении. Если мы продолжим размен, то получим тот же результат, как и при расположении по возрастанию (если противник тоже продолжит), и потеряем разность стоимостей (при обрыве с его стороны).

Тем самым доказано, что рассматриваемый вариант размена не улучшает первоначальный оптимальный вариант.

Вообще говоря, рассмотренная ранее блокада может изменить этот принцип. Однако принятая шкала стоимостей фигур такова, что блокада траектории фигуры с меньшей стоимостью фигурой с большей стоимостью встречается, только если блокирующая фигура — ферзь; среди фигур — участниц размена стоимость ферзя плюс стоимость любой другой фигуры не ниже стоимости двух любых других фигур. Это обстоятельство позволяет распространить данный принцип на случай заблокированных фигур (король участия в блокаде при размене принимать не может).

Упорядочим по этому принципу отдельно свои фигуры и фигуры противника, вычислим собственно результат размена. Для этого проводится минимум по всевозможным моментам обрыва размена обеими сторонами. При этом считается, что противник обязан произвести хотя бы одно взятие, если он имеет такую возможность. В случае, если у противника вообще нет фигур, которые могут принять участие в размене на данном поле, в качестве результата размена вырабатывается признак, показывающий, что поле не атаковано (проходимо).

При подсчете минимакса, естественно, учтены следующие особенности шахматных правил. Во-первых, король не принимает участия в размене, если он не последняя фигура в размене. Во-вторых, поле размена может являться полем превращения одной из пешек, в результате чего изменяется ее стоимость. Наконец, в-третьих, первым ходом размена может являться взятие на проходе, что также необходимо учитывать.

Заметим, что результат размена определяется не полем доски, на котором он производится, а полем траектории.

Процедура вычисления результата оптимального размена на  $\alpha$ -поле траектории используется не только для позиционной оценки, но и для определения приоритета траекторий. Приоритет — это линейная функция нескольких переменных. Одна из этих переменных соответствует уязвимости траектории; это означает, что все поля этой траектории проходимы, т. е. контролируются

стороной, которой принадлежит эта траектория. Очевидно, при таком включении в программу процедуры размена, вычисляя позиционную оценку, мы одновременно можем определить уязвимость траектории,  $\alpha$ -поля которой анализируются. Имеется в виду уязвимость траектории нападения без учета уязвимости полей отступления атакующей фигуры.

Для вычисления приоритета траектории, помимо ее уязвимости, имеет значение еще уязвимость комлевой траектории зоны, с которой связана данная траектория, например, она является в этой зоне траекторией отрицания. В момент определения приоритета данной траектории выяснять уязвимость соответствующей комлевой траектории довольно сложно и неэффективно, так как для этого нужно разворачивать пучок другой фигуры. Гораздо удобнее выделить определение уязвимых зон в специальную процедуру. Таким образом, мы вначале определяем приоритет зоны, а затем приоритеты траекторий в этой зоне.

Итак, перед тем как перейти к процедуре выбора хода в соответствии с приоритетом, передается управление процедуре определения уязвимых зон. Эта процедура разворачивает лишь пучки комлевых траекторий зон нападения и выясняет уязвимость этих траекторий, используя процедуру оптимального размена на поле. Информация о приоритете комлевой траектории зоны нападения записывается в ростке зоны, т. е. в следе комлевого пучка зоны на  $\omega$ -поле пучка (см. приложение 1). Эта же процедура определяет количество уязвимых зон и суммарную длину их комлевых траекторий — эти числа необходимы для вычисления позиционной оценки.

Таким образом, в процедуре выбора хода по приоритету (при определении приоритета некоторой траектории) выяснение вопроса о приоритете зоны, с которой связана эта траектория, не представляет сложности. Для этого надо лишь обратиться к ростку соответствующей зоны и извлечь из него информацию о приоритете этой зоны.

Рассмотренные в настоящем приложении процедуры выполняют в программе «Пионер», в основном, две функции: вычисление позиционной оценки и определение приоритета ходов для включения в перебор. Все они основаны на определении результата оптимального размена на поле. Обращает на себя внимание, что и позиционная оценка, и приоритет описаны без точных числовых характеристик. Дело в том, что соотношение между позиционной оценкой и материалом, а также взаимная значимость различных компонентов приоритета — уязвимости, длины траекторий, вилочности и т. п. — приняты пока приближенно и будут уточнены в процессе дальнейших экспериментов с программой «Пионер».

**БИБЛИОТЕКА ЭНДШПИЛЯ ПРОГРАММЫ «ПИОНЕР»  
(ИСПОЛЬЗОВАНИЕ ОПЫТА ПРОШЛОГО  
ПО СПРАВОЧНОМУ МЕТОДУ И МЕТОДУ СТРЕМЛЕНИЯ)**

**А. Д. Юдин**

**3.1. Введение.** Как отмечалось выше, при составлении программы «Пионер» была поставлена задача сделать ее содержание по возможности близким к содержанию «программы» шахматного мастера. Решению этой задачи служит создание информационной системы «Опыт прошлого».

Важнейшим фактором при решении вопроса о целесообразности создания любой информационно-справочной системы является эффективность использования этой системы. Только в случае, если есть основания к ее успешному использованию, если экспериментально доказано, что вновь созданная система приведет к эффективному решению задач, стоящих перед исследователями, тогда только и необходимо создавать такую информационно-справочную систему.

В нашем случае речь идет о целесообразности создания информационно-справочной системы «Опыт прошлого» шахматной программы для ЭВМ при моделировании мышления шахматиста. Мастер, играя в шахматы, систематически обращается к опыту прошлого в любой стадии партии. Он делает это в тех узлах дерева перебора, где это целесообразно. Отсюда следует, что для повышения эффективности использования опыта прошлого, во-первых, число узлов дерева перебора должно быть невелико и, во-вторых, программа использования опыта прошлого должна быть организована так, чтобы это требовало малого расхода ресурсов ЭВМ. Если удовлетворение второму требованию целиком и полностью находится в руках создателей информационной системы «Опыт прошлого», то первое требование неразрывно связано с принятым алгоритмом шахматной игры. И при создании шахматной программы по алгоритму М. Ботвинника мы имеем дело с небольшим «человеческим» деревом перебора.

Итак, «программу» шахматиста можно условно разделить на две части: 1) программу поиска хода в оригинальной ситуации; 2) библиотеки дебюта, миттельшпиля и эндшпиля и программы пользования этими библиотеками.

Разделение условно потому, что иногда эти две части работают параллельно, в сочетании друг с другом. При переборе в программе поиска хода часто используются библиотечные сведения.

В данном приложении будут рассмотрены вопросы, связанные



с созданием библиотеки эндшпиля шахматной программы «Пионер» и алгоритмов пользования этой библиотекой по справочному методу и методу стремления, а также программная реализация указанных алгоритмов.

Поскольку в программе «Пионер» фигуры передвигаются по траекториям в соответствии с принятой целью игры, то задача пользования библиотекой эндшпиля может быть решена по аналогии с действиями шахматного мастера. Мастер, когда он играет в шахматы, не только ищет совпадение позиций из партии (или из дерева перебора) с позициями из библиотеки, но и стремится получить эти библиотечные позиции (стремление к библиотечной позиции будет рассмотрено в п. 3.10—3.17). Так будет действовать и «Пионер»; после нахождения в библиотеке похожей (близкой) и выгодной позиции, он будет искать траектории для фигур и передвигать фигуры так, чтобы эту выгодную позицию получить. Как только такая позиция будет достигнута, произойдет совпадение позиции из дерева перебора с позицией из библиотеки, оценка позиции станет известной и вариант оборвется.

**3.2. Постановка задачи.** Обычно в справочниках по эндшпилю приводятся позиции и соответствующие им варианты. Но сильный мастер, когда он играет эндшпиль, как правило, не помнит эти варианты. Он помнит опорные, узловые позиции, их оценки и, если это необходимо, трудные первые ходы. Все остальное мастер находит при помощи своего алгоритма поиска хода.

Поэтому представляется целесообразным включить в библиотеку не варианты, а узловые позиции, их оценки и первые ходы (если ходы трудные). Это существенно упрощает методику составления библиотеки и уменьшает объем запасенной информации [4].

Таким образом, перед мастером (или программой) стоит следующая задача. Имеется конкретная эндшпильная позиция, возникшая в партии или в каком-либо варианте перебора. Необходимо при помощи библиотеки оценить (выиграно, ничья, проиграно) данную позицию при очереди хода за той или иной стороной и указать наилучший первый ход (если это необходимо).

**3.3. Конфигурации.** Конфигурацией позиции из  $N$  фигур будем называть набор разностей координат  $\Delta_1, \Delta_2, \dots, \Delta_{N-1}$ , где  $\Delta_i = L_1 - L_{i+1}$  для  $i=1, 2, \dots, N-1$ ;  $L_i$  — линейная координата  $i$ -й фигуры позиции (может принимать значения от 1 до 64).

Возьмем какую-нибудь позицию технического эндшпиля. Далее, зафиксировав относительное расположение фигур (конфигурацию), будем сдвигать позицию относительно вертикальной и горизонтальной осей координат всеми возможными способами, не выходя при этом за пределы доски, а также не нарушая шахматных правил с точки зрения возможного расположения пешек. Получаем множество позиций данной конфигурации. При составлении программы задача состоит в том, чтобы записать данное множество

позиций (их бывает до 40 в множестве) вместе с их решением (оценки, первые ходы) в библиотеку эндшпиля в удобном для работы программы виде.

**3.4. Явление «краевого эффекта». Формулы разбиения.** Возможность компактной записи требуемой информации возникает в связи с явлением, которое получило название «краевой эффект». Было установлено, что указанное множество позиций нетрудно разбить на несколько непересекающихся (с точки зрения оценок и конфигураций первых ходов, решающих позиции) подмножеств:

- позиции с влиянием одного из вертикальных краев доски;
- позиции с влиянием одного из горизонтальных краев доски;
- позиции с совместным влиянием горизонтального и вертикального краев доски (так называемые «угловые позиции»);
- все остальные позиции множества, когда отсутствует влияние краевого эффекта.

Оценки и первые ходы (для данной конфигурации) внутри каждого подмножества позиций остаются неизменными. Более того, поскольку все множество характеризуется неизменной конфигурацией, т. е. взаимным расположением фигур, а функция оценки имеет лишь три значения (выиграно, ничья, проиграно), то для большинства конфигураций некоторые из вышеуказанных подмножеств можно объединить из-за совпадения оценок и первых ходов. Часто также встречаются пустые подмножества, не содержащие ни одной позиции.

Таким образом, чтобы распознать любую позицию множества, достаточно иметь в библиотеке лишь одну позицию, например из углового подмножества (часто, но далеко не всегда, оно «вырождается» в одну позицию), а также формулы разбиения множества, которые характеризуют границы изменения оценок. Подробное решение задачи для одного множества будет показано ниже на примере.

**3.5. Симметрии.** Большое значение в программе имеет также использование симметрий:

— *фланговой симметрии* — отражения позиции относительно вертикальной оси доски;

— *симметрии цвета* — отражения позиции относительно горизонтальной оси доски, сопровождающегося переменой цвета фигур;

— *диагональных симметрий* — отражений позиции относительно диагоналей a1—h8 и h1—a8 (действуют в позициях без пешек).

Каждое поле доски характеризуется своими двумерными координатами  $x$ ,  $y$  или линейной координатой  $L$ , которые связаны между собой:

$$L=8(y-1)+x.$$

Например, поле f5 имеет координаты:  $x=6$ ,  $y=5$  или  $L=38$ .

Опуская несложные преобразования, укажем формулы для

симметрирования. Пусть  $x, y, L$  — координаты поля, на котором стояла фигура до симметрирования. Тогда для симметрии флангов

$$L_{\text{фл}} = L + 9 - 2x;$$

для симметрии цвета

$$L_{\text{цв}} = L + 8(9 - 2y);$$

для диагональных симметрий

$$L_{a1-h8} = 8L - 63y + 56 \text{ и } L_{h1-a8} = 63y + 9 - 8L.$$

Укажем, что для любого поля доски

$$L_{a1-h8} + L_{h1-a8} = 65.$$

Это и некоторые другие полученные соотношения используются для образования симметрированных конфигураций.

Указанные симметрии используются как отдельно, так и совместно, в соответствии с условиями их применения. Так, позиция Белые: Кре3, Лh1; Черные: Крг5, Ch4, помещенная в библиотеку прямым или косвенным (с помощью формул, учитывающих краевой эффект) образом, характеризует еще 15 позиций, связанных с вышеприведенной совместным использованием всех трех симметрий.

**3.6. Структура библиотеки. Классы. Кодирование информации.** Структура библиотеки выглядит следующим образом. Все позиции технического эндшпиля разбиты по материалу на 31 класс. Например, «Король с пешкой против короля»; «Король, конь и пешка против короля и слона» и т. п.

Таким образом, внутри каждого класса соотношение материала в позициях неизменно.

Класс, помещенный в библиотеку, представляет собой прямоугольную матрицу размером  $m \times n$ , где  $m$  — количество позиций в классе;  $n = \text{ENT}[(N+1)/2] + 2$ ;  $N$  — количество фигур в каждой позиции данного класса.

Каждая строка матрицы соответствует одной позиции. Покажем кодирование информации на примере позиции

Белые: Крf2, п.g2; Черные: Крс6, п.h4.

Решение позиции выглядит следующим образом. При ходе белых выигрывает 1.Крf2—g1, при ходе черных — ничья: 1...h4—h3. Пусть эта позиция имеет номер  $i$  в классе «Король с пешкой против короля с пешкой».

Оценка  $r$  может принимать следующие значения:

$$r = \begin{cases} 0 & \text{— выиграно,} \\ 1 & \text{— ничья,} \\ 2 & \text{— проиграно.} \end{cases}$$

Первые  $n-2$  позиции (имеются в виду места в строке матрицы, а не шахматные позиции) в строке образуют попарно объединен-

ные координаты фигур. В двух последних столбцах стоят пятизначные (в общем случае) числа — ходы с приписанными впереди оценками. Таким образом, элементы  $i$ -й строки нашей матрицы соответственно равны:  $M_{i1}=1415$ ;  $M_{i2}=4332$ ;  $M_{i3}=1407$ ;  $M_{i4}=13224$ .

В классах с нечетным  $N$  в  $(n-2)$ -м столбце стоят двузначные координаты  $N$ -й фигуры позиций.

Каждой позиции класса соответствуют формулы разбиения множества, символом конфигурации которого является данная позиция, а также формулы коррекций оценок и первых ходов (см. пример ниже).

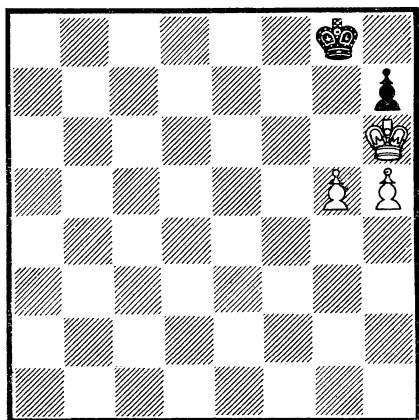


Рис. 37. Пример позиции

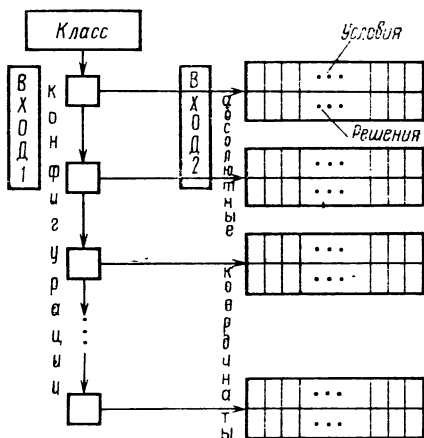


Рис. 38. Схема поиска информации в двумерной таблице с субординацией входов

При указанной структуре пополнение библиотеки новыми позициями (или исключение позиций из библиотеки) означает включение (исключение) строки или строк в (из) соответствующую матрицу-класс.

Рассмотрим поиск по справочному методу и действие краевого эффекта на примере простой конфигурации (см. позицию на рис. 37 и таблицу). Позиция и таблица, кроме последней колонки (приведенной для наглядности), «записаны» в библиотеке и соответствуют позиции на диаграмме в классе «Король и две пешки против короля и пешки».

Использованием симметрии флангов и цвета каждая позиция множества в процессе поиска преобразуется в четыре позиции.

**3.7. Организация информации в виде двумерных таблиц с субординацией входов.** Явление краевого эффекта позволяет расположить всю справочную информацию об эндшпиле в некоторое количество двумерных таблиц с субординацией входов.

*Двумерной таблицей с субординацией входов* назовем таблицу с двумя входами, один из которых является независимым, а другой зависит от первого.

Т а б л и ц а

*Характеристики множества позиций (к диаграмме позиции на рис. 37)*

Номер подмножества	Способ получения подмножества из заданной позиции	Оценка позиции		Первый ход		Количество позиций в подмножестве
		при ходе белых	при ходе черных	белых	черных	
1	Сдвиг вниз на 1—3 поля	ничья		$\infty^1$	$\infty$	3
2	Подмножества нет <sup>2</sup>					0
3	Совпадает с заданной позицией	ничья	выиграно у белых	$\infty$	$\infty$	1
4	Сдвиг влево на 1—6 полей и вниз на 1—3 поля	Выиграно у белых		$l_0^{BK} l_1^{BK}$ , где <sup>3</sup> $l_1^{BK} = l_0^{BK} + 9$	$\infty$	18

<sup>1</sup>  $\infty$  означает, что указание первого хода необязательно; <sup>2</sup> 2-е подмножество для данного признака „пустое“, т. е. не содержит ни одной позиции; <sup>3</sup>  $l_0^{BK}$  — начальное положение (координата) белого короля,  $l_1^{BK}$  — положение белого короля после хода.

В нашем случае каждому классу с характерным соотношением материала соответствует одна двумерная таблица. На рис. 38 представлена структурная схема программы поиска информации в такой таблице. Первым, независимым (или безусловным), входом является поиск по конфигурации, т. е. поиск совпадения относительных координат позиции из партии (перебора) с относительными координатами одной из библиотечных позиций-символов конфигураций. В том и только в том случае, если совпадение обнаружено, управление передается подпрограмме поиска по второму входу таблицы, т. е. будет найдено решение уже в абсолютных координатах. Следует отметить, что эта часть таблицы (большая ее часть) существует не в явном виде, а лишь в виде соответствующих формул разбиения и воспроизводится по этим формулам тогда и только тогда, когда найдено совпадение относительных координат [6].

**3.8. Алгоритм пользования библиотекой эндшпиля (поиск точного совпадения).** Комплекс подпрограмм, реализующий алгоритм

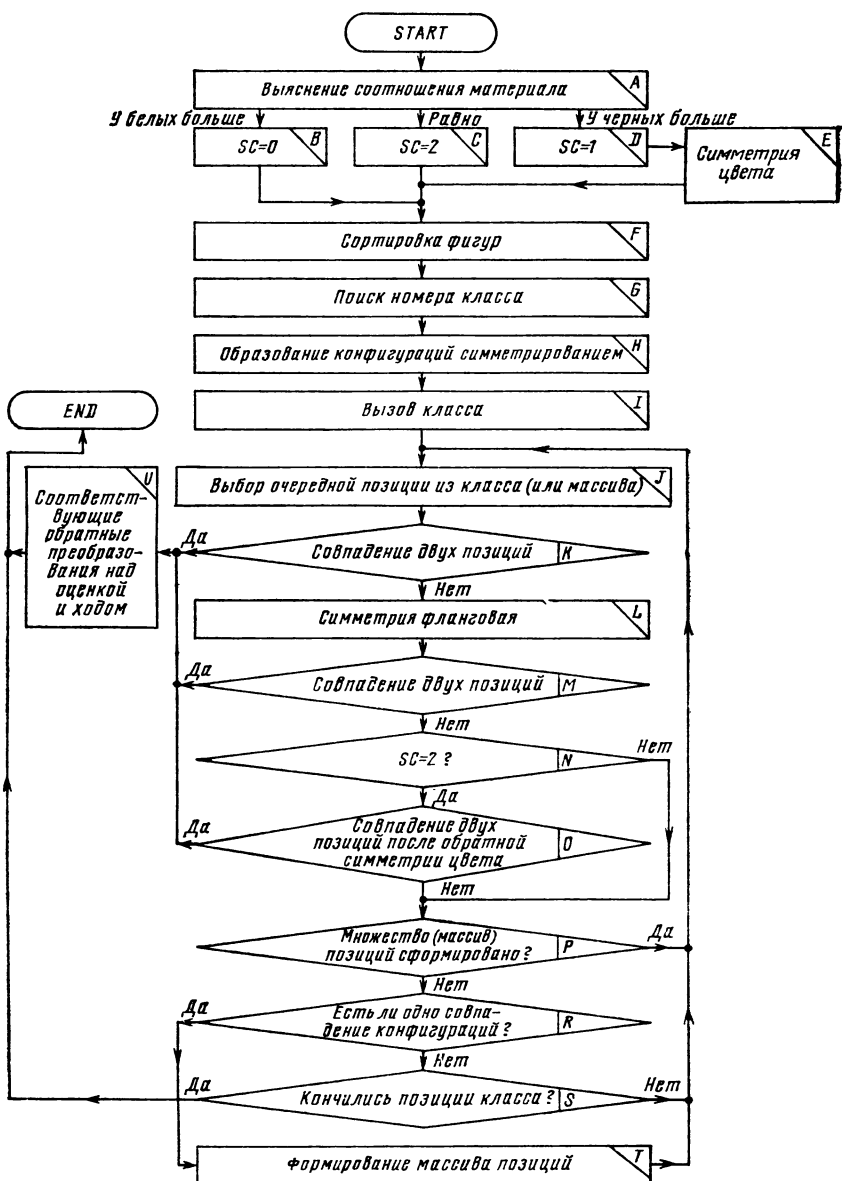


Рис. 39. Блок-схема алгоритма пользования библиотекой эндшпиля (поиск точного совпадения)

пользования, включается тогда, когда в позиции партии или какого-либо варианта перебора впервые встретилось соотношение материала, отвечающее (с точностью до перемены цвета фигур) соотношению материала одного из классов библиотеки.

Рассмотрим блок-схему алгоритма пользования библиотекой эндшпиля (рис. 39). (На схеме для упрощения не указаны процедуры диагональных симметрий.)

Процедура А выясняет соотношение материала сторон, что необходимо для решения вопроса о возможности симметрии цвета. Все дальнейшие процедуры предусматривают материальный перевес белых или равенство; так построена и сама библиотека. Поэтому, если в позиции партии или перебора у черных материальный перевес, то производится симметрия цвета (процедура Е) и находится решение для позиции с переменной цвета фигур, а после нахождения решения процедура У производит обратное преобразование, на необходимость которого укажет присвоенное значение  $SC=1$  (процедура D). Если на доске или в позиции перебора материальное равенство, то можно искать решение как для исходной, так и для симметрированной по цвету позиции, о чем говорит значение  $SC=2$  (процедура С).

Процедура F приводит исходную позицию к виду, удобному для сравнения с библиотечными, т. е. строками матрицы-класса. Здесь происходит сортировка фигур и соответствующее кодирование исходной позиции. После этого вычисляется номер класса (процедура G), который процедура I вызовет в оперативную память.

Рассмотрим процедуру G. Для всех конфигураций данного класса существует некоторая общая характеристика, названная шаблоном, определяющаяся конкретным соотношением фигур.

Шаблом позиции А назовем массив  $U$  из 12 элементов, где  $U(1)=1$ ;  $U(2)$  — количество белых ферзей в позиции А; ...  $U(12)$  — количество черных пешек в позиции А.

Первые шесть элементов этого массива —  $U_w(6)$  — назовем белым шаблоном; последние шесть —  $U_b(6)$  — черным шаблоном. Такие шаблоны запасены для каждого класса и являются его «материальной» характеристикой.

Далее для позиции партии (перебора) «изготавливается» такой же шаблон. Поиск номера класса сравнением шаблонов показан на рис. 40. Этот поиск осуществляется с учетом симметрии цвета, на возможность или необходимость которой указывает значение  $SC$ .

Совпадение шаблона исходной позиции с одним из шаблонов классов автоматически указывает номер класса, который процедура I вызовет в оперативную память машины.

Процедура Н из исходной позиции симметрированием образует всевозможные конфигурации.

Когда в первый раз управление передается процедуре J, то процедуры K, L, M, N и O производят поиск точного, с учетом симметрирования, совпадения исходной позиции с очередной позицией-символом множества в классе (поиск по первому входу в соответствующей двумерной таблице); при успешном завершении поиска управление передается процедуре U, которая производит обратные (с учетом сделанных симметрий) преобразования над найденными оценкой и ходом.

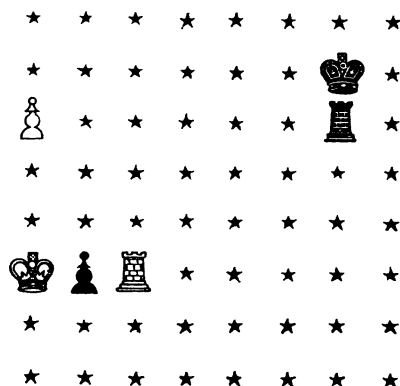
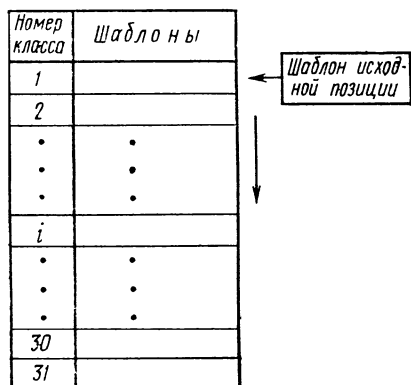


Рис. 40. Схема поиска номера класса сравнением шаблонов

				MASSIV= 32
1	A2	=	B9-B9	= B9-B9
2	B2	=	B9-B9	= B9-B9
3	A3	+-	C3-G3	= G6-A6
4	B3	+-	D3-H3	= H6-B6
5	A4	+-	C4-G4	= G7-A7
6	B4	+-	D4-H4	= H7-B7

Рис. 41. Пример массива позиций, заложенных вместе с их решением в библиотеку

Предположим, а это наиболее вероятный случай, что точного совпадения (с учетом симметрий) не обнаружено. Тогда проверка P, которой управление передается впервые, дает отрицательный ответ и передает управление проверке R; последняя решает вопрос о необходимости восстановления или образования множества по позиции-символу, т. е. по той позиции из библиотеки, с которой мы сейчас имеем дело.

Для того чтобы решение существовало в пределах множества для данной позиции класса, необходимо и достаточно, чтобы конфигурация этой позиции совпала хотя бы с одной из симметрированных конфигураций позиции партии или перебора. Доказатель-



ство как необходимости, так и достаточности этого утверждения элементарно вытекает из определения конфигурации, правил симметрирования и способа образования множества позиций данной конфигурации.

Таким образом, при положительном результате проверки R мы имеем полную гарантию того, что массив или множество позиций зря сформированы не будут (процедура T). Это означает, что после проверки всех (в общем случае) позиций множества, сформированного процедурой T, управление неизбежно будет передано процедуре U вследствие положительного результата одной из проверок K, M или O. Это и есть поиск по второму входу в соответствующей двумерной таблице.

Следует отметить, что при помощи формул разбиения и коррекции оценок и ходов процедура T образует множество позиций уже с их решениями, которые записываются в двух последних столбцах матрицы. После необходимых обратных преобразований над оценкой и ходом (процедура U) задача решена.

Предположим теперь, что проверка R дает отрицательный результат. Тогда управление передается процедуре J, выбирается очередная позиция-символ и т. д.

Если мы перебрали все позиции-символы в классе и ни разу проверка R не дала положительного результата, то по окончании всех позиций-символов в классе проверка S дает положительный ответ, а это означает, что точного решения не существует, т. е. поиск точного совпадения по конфигурации класса успешно не завершился и, следовательно, искомая позиция в библиотеке не найдена. Это происходит достаточно часто, так как количество конфигураций в библиотеке сравнительно небольшое, а число возможных в шахматной партии конфигураций даже при малом количестве фигур технического эндшпиля хотя и ограничено, но весьма велико. В этом случае мы будем иметь дело с аппаратом стремления к библиотечной позиции (см. п. 3.10 и далее).

Приведенная методика позволяет из заложенных в настоящий момент в библиотеку 633 позиций получить до 14000 позиций (без учета симметрий).

**3.9. Примеры работы подпрограммы пользования библиотекой эндшпиля.** Целью экспериментов с библиотекой технического эндшпиля, кроме проверки эффективности работы подпрограммы пользования справочной эндшпиля при обращении к ней из подпрограммы поиска хода, была также аналитическая проверка оценки заложенной информации. При этом проверялось влияние краевого эффекта на оценку позиций, а также использование аппарата симметрий.

После того, как была закончена шахматная часть работы, т. е. составлена сама библиотека, была разработана специальная подпрограмма, которая расшифровывала решение каждой конфигурации, заложенной в справочную технического эндшпиля (при помощи формул разбиения — см. п. 3.4), и печатала для каждой конфигурации весь массив соответствующих позиций (с их решением). Один такой массив показан на рис. 41.

На выданной машине диаграмме — заложенная конфигурация; в первой колонке таблицы — номер по порядку; во второй — положение белого короля; в третьей — оценка при ходе белых («+» — выиграно у белых, «=» — ничья); в четвертой — ход белых (название фигуры, делающей ход, не выдается; код B9—B9 означает, что указание хода, решающего позицию, не обязательно); в пятой — оценка при ходе черных; в шестой — ход черных.

Такие массивы были розданы для проверки десяткам сильных шахматистов, после чего были устранены некоторые ошибки, неизбежные при составлении такой большой библиотеки и ее программировании.

На рис. 42а—в приведены результаты работы подпрограммы пользования библиотекой. Это решения симметризованных позиций, полученных из одной и той же конфигурации. Поясним подписи на машинных распечатках.

NEW POSITION — новая позиция, представленная подпрограмме для анализа;

WHITE TO PLAY — ход белых;

BLACK TO PLAY — ход черных;

DRAW BY — ничья посредством ...;

WHITE WINS BY — белые выигрывают ходом ...;

BLACK WINS BY — черные выигрывают ходом ....

**3.10. Стремление к библиотечной позиции.** Все рассмотренное выше касалось лишь точного совпадения позиций партии (перебора) и библиотечной. Предположим теперь, что позиции, возникшей в партии или в каком-либо варианте перебора, нет в библиотеке эндшпиля, т. е. поиск точного совпадения не привел к положительному результату. Какую же задачу справочного характера ставит перед собой в этом случае шахматный мастер, а следовательно, и программа, моделирующая его мышление?

Х. Р. Капабланка указывал [9], что шахматный мастер не только регистрирует в своих расчетах позиции из библиотеки, чтобы оборвать вариант, но и стремится получить ту или иную выгодную ему библиотечную позицию. Программа должна уметь делать и это.

Пусть конфигурация исходной позиции не найдена среди позиций-символов множеств библиотечных классов. Тогда в библиотеке определяется позиция (или группа позиций), наиболее «близкая» к той, которая получилась в партии или переборе, и организуется *стремление* к этой близкой позиции из исходной. Оно заключается в следующем. В том случае, если рассматриваемая близкая позиция имеет удовлетворяющую нас оценку, в математическое отображение включаются так называемые *планируемые* траектории фигур активной (стремящейся) стороны, т. е. траектории, ходы по которым ведут из исходной позиции к близкой. Активная сторона создает свои зоны. Здесь алгоритм пользования библиотекой действует совместно с алгоритмом поиска хода в оригинальной ситуации. Для того чтобы стремление было организовано, необходимо наличие в математическом отображении траекторий всех несовпадающих фигур пассивной стороны. Важно отметить, что

NEW POSITION

```

* * * * *
* * * * * -K *
* * * * *
* * * * * K *
-P -P -P * * * *
* * * * *
P P P * * * *
* * * * *

```

WHITE TO PLAY.  
\*\*\*\*\*

DRAW BY

1. PB2-B3

```

* * * * *
* * * * * -K *
* * * * *
* * * * * K *
-P -P -P * * * *
* P * * * *
P * P * * *
* * * * *

```

BLACK TO PLAY:  
\*\*\*\*\*

BLACK WINS BY

1. \*\*\* PB4-B3

```

* * * * *
* * * * * -K *
* * * * *
* * * * * K *
-P * -P * * * *
* -P * * * *
P P P * * *
* * * * *

```

*a*

NEW POSITION

```

* * * * *
* -K * * * *
* * * * *
* K * * * *
* * * * * -P -P -P
* * * * *
* * * * * P P P
* * * * *

```

WHITE TO PLAY:  
\*\*\*\*\*

DRAW BY

1. PG2-G3

```

* * * * *
* -K * * * *
* * * * *
* K * * * *
* * * * * -P -P -P
* * * * * P *
* * * * * P * P
* * * * *

```

BLACK TO PLAY:  
\*\*\*\*\*

BLACK WINS BY

1. \*\*\* PG4-G3

```

* * * * *
* -K * * * *
* * * * *
* K * * * *
* * * * * -P * -P
* * * * * -P *
* * * * * P P P
* * * * *

```

*b*

NEW POSITION

```

* * * * *
* -K * * * *
* * * * *
* K * * * *
* * * * * -P -P -P
* * * * *
* * * * * P P P
* * * * *

```

WHITE TO PLAY:  
\*\*\*\*\*

DRAW BY

1. PG2-G3

```

* * * * *
* -K * * * *
* * * * *
* K * * * *
* * * * * -P -P -P
* * * * * P *
* * * * * P * P
* * * * *

```

BLACK TO PLAY:  
\*\*\*\*\*

BLACK WINS BY

1. \*\*\* PG4-G3

```

* * * * *
* -K * * * *
* * * * *
* K * * * *
* * * * * -P * -P
* * * * * -P *
* * * * * P P P
* * * * *

```

*в*

Рис. 42. Образцы результатов работы подпрограммы пользования библиотекой эндшпиля

эти траектории должны быть «антивилочными» в том смысле, что пассивная сторона должна действовать по этим траекториям лишь в крайних случаях. Кроме того, в отличие от основных принципов алгоритма поиска хода в оригинальной ситуации, для препятствия передвижения фигур пассивной стороны по указанным траекториям никакие зоны не создаются.

Покажем теперь, в чем заключается поиск близких позиций.

**3.11. Поиск близких позиций.** Прежде чем приступить к описанию поиска близких позиций, определим следующие понятия.

Позиции  $A$  и  $B$  будем называть позициями *равного материала*, если  $U^A_i = U^B_i$  для любого  $i=1, 2, \dots, 12$ , где  $U^A_i$  и  $U^B_i$  — элементы шаблонов позиций  $A$  и  $B$ .

*Несовпадением*  $\sigma_{AB}$  позиций  $A$  и  $B$  равного материала назовем

$$\sigma_{AB} = \frac{1}{2} \sum_{i=1}^{64} c_i,$$

где

$$c_i = \begin{cases} 1, & \text{если } a_i \neq b_i; \\ 0, & \text{если } a_i = b_i. \end{cases}$$

Здесь  $a_i$  и  $b_i$  — принятые в программе коды фигур, расположенных на  $i$ -м поле позиций  $A$  и  $B$ , т. е.

$a_i = \left\{ \begin{array}{l} 11, \\ 13, \\ 14, \\ 15, \\ 16, \\ 17, \\ 22, \\ 23, \\ 24, \\ 25, \\ 26, \\ 27, \\ 0, \end{array} \right.$	если на	$i$ -м поле позиции $A$ стоит	белая пешка
	13,	то же	белый конь
	14,	"	белый король
	15,	"	белая ладья
	16,	"	белый слон
	17,	"	белый ферзь
	22,	"	черная пешка
	23,	"	черный конь
	24,	"	черный король
	25,	"	черная ладья
	26,	"	черный слон
	27,	"	черный ферзь
	0, если $i$ -е поле позиции $A$ пусто;		

$i$  — линейная координата.

Пусть  $T_1, T_2, \dots, T_{\sigma_{AB}}$  — последовательность траекторий, ходы по которым переводят позицию  $A$  в позицию  $B^*$ .

Разностью  $\Delta_{AB}$  позиций  $A$  и  $B$  равного материала назовем

$$\Delta_{AB} = \sum_{j=1}^{\sigma_{AB}} l_j,$$

где  $l_j$  — длина траектории  $T$  в полуходах для любого  $j=1, 2, \dots$ ,  $\sigma_{AB}$ .

Позиции  $A$  и  $B$  равного материала будем называть *близкими*, если  $\sigma_{AB} \leq \sigma_{\max}$ ;  $\Delta_{AB} \leq \Delta_{\max}$ , где  $\sigma_{\max}$  — максимально допустимое несоответствие (число несоответствующих фигур);  $\Delta_{\max}$  — максимально допустимая разность позиций.

В первом приближении было принято, что поиск близких позиций ведется лишь среди позиций равного материала. Это означает, что однозначно определен класс библиотечных позиций-символов конфигураций, материал в которых (с учетом симметрии цвета) в точности совпадает с материалом в исходной позиции (см. рис. 40). Значения величин  $\sigma_{\max}$  и  $\Delta_{\max}$  являются текущими ограничениями поиска.

Задача состоит в следующем. Имеется исходная позиция  $A$ . Требуется найти группу библиотечных позиций  $B_1, B_2, \dots, B_n$ , каждая из которых является близкой по отношению к  $A$  в указанном выше смысле.

Заметим, что здесь мы не будем анализировать соотношение оценок позиции  $A$  и позиций  $B_1, B_2, \dots, B_n$ ; этот анализ будет проведен ниже. Таким образом, не следует полагать, что к каждой позиции  $B_i$  ( $i=1, 2, \dots, n$ ) впоследствии будет организовано стремление.

Итак, группу позиций  $B_1, B_2, \dots, B_n$  назовем *претендентами на стремление*. Очевидно, что в начале поиска претендентом на стремление является любая позиция из любого множества, описываемого позициями-символами данного класса.

Одной из задач поиска близких позиций является максимальное сокращение числа претендентов на стремление без использования внутрисистемного аппарата нахождения траекторий. Были найдены некоторые ограничения (назовем их *фильтрами*), благодаря которым удастся «забраковать» большую часть множеств библиотечных позиций с точки зрения наличия в этих множествах позиций, близких к исходной, забраковать уже на уровне соответствующих конфигураций. Что же это за фильтры?

**3.12. Фильтр по относительному расположению пешек.** Этот фильтр действует лишь при наличии в исходной (а следовательно,

\* В дальнейшем будем говорить (для простоты), что траектории ведут из позиции в позицию, подразумевая под этим, естественно, ходы по траекториям.

и в библиотечных) позиции двух или более пешек любого цвета.

Имеем исходную позицию  $A$  и позицию  $C$  — символ множества позиций определенной конфигурации. Образует из позиции  $A$  шаблон пешек. Для этого условно «спроектируем» расположение всех пешек позиции на первую горизонталь и присвоим восьми полям первой горизонтали значения

$$SH_i^p = \begin{cases} 0, & \text{если нет пешек на } i\text{-й вертикали;} \\ 10p_i^w + p_i^b, & \text{если на } i\text{-й вертикали есть пешки (пешка),} \end{cases}$$

где  $p_i^w$  — количество белых пешек на  $i$ -й вертикали;  $p_i^b$  — количество черных пешек на  $i$ -й вертикали;  $i=1, 2, \dots, 8$ .

Получили массив из восьми элементов  $SH^p$ . Далее выделим из этого массива ненулевую часть следующим образом. Отбросим слева нули так, чтобы первым элементом был  $SH_i^p \neq 0$ , а справа так, чтобы последним элементом был  $SH_j^p \neq 0$  ( $j \geq i$ ). Полученный массив  $SH^A$  длиной  $l_A = j - i + 1$  назовем шаблоном пешек позиции  $A$ .

Аналогичным способом получим шаблон пешек  $SH^C$  позиции  $C$  длиной  $l_C$ .

Докажем следующее утверждение. Не существует последовательности траекторий, переводящих позицию  $A$  в любую позицию множества, характерного для  $C$ , если а)  $l_A \neq l_C$  или б)  $l_A = l_C$ , но существует хотя бы один номер  $i$  из набора  $i=1, 2, \dots, l_A$ , такой, что  $SH_i^A \neq SH_i^C$ .

**Доказательство.** Пусть  $C_1$  — одна из позиций множества, характеризующегося позицией-символом  $C$ .

Поскольку позиции  $A$  и  $C$  — равного материала, то и позиции  $A$  и  $C_1$  — также равного материала.

Пусть выполнено одно из условий: а) или б).

Предположим противное, т. е. что существует последовательность траекторий, переводящих  $A$  в  $C_1$ . Тогда, раз выполнено хотя бы одно из указанных условий, то в этой последовательности обязательно есть хотя бы одна траектория, меняющая вертикаль хотя бы одной пешки. Это следует из того, что выполнение одного из условий а) или б) означает несовпадение шаблонов пешек. Но, как известно, пешка может сменить вертикаль лишь посредством взятия (взятий). А это противоречит тому, что  $A$  и  $C_1$  — равного материала. И так для любого  $C_1$  из  $C$ . Пришли к противоречию. Следовательно, вышеуказанное утверждение доказано.

Значит, если шаблоны  $SH^A$  и  $SH^C$  не совпадают полностью, то все позиции множества, характерного для  $C$ , исключаются из списка претендентов на стремление.

Следует напомнить, что все проведенные рассуждения справедливы с точностью до симметрий. Это означает, что если выполнено условие б), то необходимо из позиции  $C$  образовать флангово-

симметрированный шаблон пешек  $SH_{\text{фл}}^C$  по формуле  $SH_{i(\text{фл})}^C = SH_{i_{C-i+1}}^C$  ( $i=1, 2, \dots, l_C$ ) и проверить выполнение условия б) для  $SH^A$  и  $SH_{\text{фл}}^C$ .

Если теперь условие б) не выполняется, т. е.  $SH^A = SH_{\text{фл}}^C$  для любого  $i=1, 2, \dots, l_A$ , то  $C$  заносится в список конфигураций-претендентов с одновременным запоминанием информации о неизбежности флангового симметрирования.

Кроме того, если материал белых и черных в данном классе одинаков (в шахматном смысле), то необходимо учесть возможность симметрии цвета для  $SH^C$  по формуле

$$SH_{i(\text{цв})}^C = \begin{cases} 0, & \text{если } SH_i^C = 0; \\ 10p_i^b + p_i^w, & \text{если } SH_i^C = 10p_i^w + p_i^b. \end{cases}$$

Фильтр по относительному расположению пешек позволяет сильно сократить список претендентов на стремление.

**3.13. Фильтр «одноцвет — разноцвет».** Этот фильтр действует лишь при наличии в исходной позиции пары слонов противоположного цвета (по одному у каждой стороны).

Дана исходная позиция  $A$ . Двумерные координаты белого слона  $x_w, y_w$ ; черного —  $x_b, y_b$ .

Известно, что

$$x+y \begin{cases} \text{четно, для черного поля шахматной доски;} \\ \text{нечетно для белого поля шахматной доски,} \end{cases}$$

если  $x$  и  $y$  — двумерные координаты определенного поля доски. Следовательно,

$$Z_A = x_w + y_w + x_b + y_b \begin{cases} \text{четно при одноцветных слонах;} \\ \text{нечетно при разноцветных слонах.} \end{cases}$$

Очевидно, не существует последовательности траекторий, переводящих позицию  $A$  в любую позицию множества, характерного для  $C$ , если  $Z_A$  и  $Z_C$  имеют разную четность, т. е.  $Z = Z_A + Z_C$  — нечетно. Здесь следует отметить, что при работе этого фильтра не требуется учета симметрий, так как любая комбинация симметрированных преобразований над позицией  $C$  не изменит четности величины  $Z_C$ , а следовательно, и четности величины  $Z$ .

Таким образом, сформирован первоначально «отфильтрованный» список конфигураций-претендентов  $C_1, C_2, \dots, C_m$ , т. е. позиции — претенденты на стремление могут находиться только в множествах, характеризующихся этими конфигурациями.

**3.14. Фильтры внутри множества позиций.** Приступим теперь непосредственно к формированию списка позиций — претендентов на стремление.

Имеем исходную позицию  $A$  и очередную конфигурацию-претендент  $C_i$ . Очевидно, не все позиции множества, характеризующегося  $C_i$ , будут претендентами на стремление. Укажем ограничения (фильтры), помогающие уменьшить количество этих позиций.

Во всех нижеследующих рассуждениях мы не будем упоминать аппарат симметрий. На самом же деле работа этого аппарата существенна на всех этапах алгоритма поиска близких позиций.

Итак, очередным сужением круга претендентов на стремление является выбор лишь тех позиций из всего их множества, у которых вертикали всех пешек в точности совпадают с вертикалями пешек исходной позиции  $A$ . Только эти позиции, согласно доказанному в п. 3.12 утверждению, являются претендентами. Но и это еще не все. Необходимо также, чтобы координаты  $y^{A_j}$  всех белых пешек позиции  $A$  удовлетворяли условию

$$y^{A_j} \leq y^{C_j},$$

а для черных пешек — условию

$$y^{A_j} \geq y^{C_j}.$$

Здесь  $y^{C_j}$  — вторые координаты (вертикали) соответствующих пешек в рассматриваемой позиции из  $C_i$ . Иначе говоря, ни одна пешка исходной позиции не должна быть «впереди» соответствующей пешки из библиотечной позиции, так как в противном случае пришлось бы искать траектории, перемещающие пешки назад. Но пешки назад не ходят и, следовательно, таких траекторий не существует.

Далее, если в позиции-претенденте есть слоны, то их белопольность или чернопольность должна соответствовать позиции  $A$ .

Наконец, все фильтры внутри множеств пройдены, и мы получаем первоначальный список позиций-претендентов на стремление.

Необходимо отметить в заключение, что в программе, реализующей данный алгоритм поиска близких позиций, не запоминаются текущие позиции-претенденты, а фиксируются лишь местоположения в программе формул разбиения, воспроизводящих эти позиции по символам соответствующих множеств.

**3.15. Получение группы близких позиций.** Когда список позиций-претендентов стал небольшим, можно приступить к непосредственной проверке «близости» каждой из них к исходной позиции.

Первоначально проверяется, не превышено ли максимально допустимое несовпадение ( $\sigma_{\max}$ ). Список претендентов еще более уменьшается. Теперь, когда количество несовпадающих фигур не превышает установленного лимита, необходимо перейти к использованию аппарата получения траекторий. После нахождения траекторий несовпадающих фигур, т. е. именно тех траекторий, которые переводят близкую позицию в позицию исходную, можно, проверив, не превышена ли максимально допустимая разность позиций



( $\Delta_{\max}$ ), принимать решение о включении претендентов в искомую группу близких позиций. Только теперь получена окончательная последовательность позиций  $B_1, B_2, \dots, B_n$ , близких к исходной позиции  $A$ .

Отметим, что значения  $\sigma_{\max}$  и  $\Delta_{\max}$  могут варьироваться в зависимости от наличия ресурсов времени и памяти. По нашему мнению, точно так же действует и шахматный мастер. В условиях цейтнота или перегруженной памяти большие значения  $\sigma_{\max}$  и  $\Delta_{\max}$  не позволит себе и чемпион мира.

Блок-схема алгоритма поиска близких позиций приведена на рис. 43.

**3.16. Реализация стремления. Антистремление.** Поясним теперь еще раз, что же понимается под стремлением из исходной позиции  $A$  к близкой библиотечной позиции  $B$ . Организовать такое стремление означает включить с соответствующими приоритетами в математическое отображение траектории, ведущие из позиции  $A$  к позиции  $B$ , оценка и решающие ходы которой при различных очередях хода нам известны из библиотеки технического эндшпиля.

Реализация метода, основанного на стремлении к библиотечной позиции, может привести к направленному формированию математического отображения и дерева перебора. Но как только совпадение будет достигнуто, в действие вступает справочный метод, основанный на точном совпадении позиций, оценка становится известной и вариант обрывается.

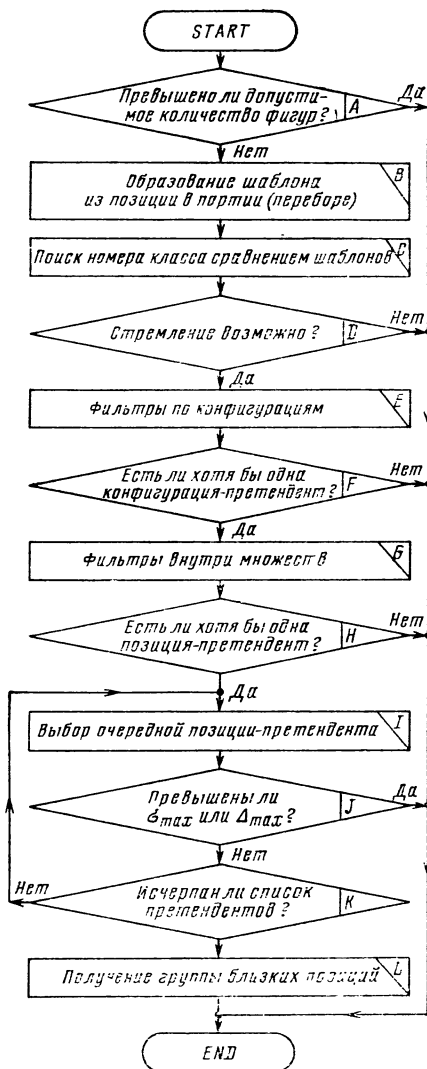


Рис. 43. Блок-схема алгоритма поиска близких позиций для стремления

Введем теперь понятие антистремления.

Предположим, что в исходной позиции  $A$  в белом узле дерева перебора (т. е. ход белых) оценка — выиграно у белых. Среди близких позиций есть позиция  $B$ , в которой оценка при ходе белых — ничья. Пусть также в математическом отображении позиции по каким-либо причинам имеются траектории белых фигур, ведущие из  $A$  в  $B$ . Тогда *антистремлением* назовем искусственное снижение приоритетов ходов по этим траекториям. Таким образом, антистремление позволяет, если можно так выразиться, избежать ловушек.

**3.17. Обращение из подпрограммы поиска хода.** Рассмотрим теперь, как происходит обращение к подпрограмме стремления из подпрограммы поиска хода в оригинальной ситуации. Если в эндшпиле в каждом узле дерева перебора проверять совпадение позиции с позицией из библиотеки, то на это уйдет немного времени — это процедура справочного характера. Но если проверять позиции во всех узлах дерева на стремление, это будет нарушением принятого алгоритма включения зон. Прежде чем стремиться, следует выяснить, что это стремление может дать дополнительно, поскольку всякое стремление связано с включением зон и расширением дерева перебора. Поэтому для стремления принят тот же порядок, что при включении зон, т. е. позиции проверяются на стремление лишь при подъеме по варианту.

Напомним, что если в математическом отображении позиции нет хотя бы одной из планируемых траекторий пассивной стороны, то стремление к такой позиции не организуется и позиция «вычеркивается» из списка претендентов на стремление.

**3.18. Стремление и точное совпадение.** Выскажем теперь некоторые соображения об общности и различии справочного метода и метода стремления.

Из вышеизложенного ясно, что поиск точного совпадения позиций является частным случаем стремления к близкой позиции. Действительно, позицию в библиотеке, в точности совпадающую с исходной, можно считать в общем случае близкой позицией.

Пусть дана исходная позиция  $A$  и некоторая библиотечная позиция  $B$ , являющаяся позицией равного с  $A$  материала. Пусть также установлено, что  $\sigma_{AB}=0$ ;  $\Delta_{AB}=0$ .

Тогда, во-первых, позиция  $B$  в точности совпадает с позицией  $A$  и, во-вторых, согласно определению близких позиций, позиция  $B$  является близкой к  $A$ .

Следовательно, при нулевых несовпадении и разности между двумя близкими позициями эти позиции в точности совпадают. В этом — общность справочного метода и метода стремления.

Различие же между этими двумя методами при использовании опыта прошлого состоит лишь в том, что справочный метод, являясь частным случаем метода стремления, не требует применения

алгоритма поиска хода в оригинальной ситуации, а точнее, во-первых, не требует использования аппарата получения траекторий и, во-вторых, не связан с некоторым расширением дерева перебора вариантов. Именно поэтому использование справочного метода является более простым, т. е. требует меньшего расхода ресурсов ЭВМ. С этим связан и избранный порядок изложения использования этих методов при составлении библиотеки технического эндшпиля и алгоритмов пользования этой библиотекой.

**3.19. Возможность широкого использования библиотечных правил в программе «Пионер».** Укажем в заключение на такую часть библиотеки эндшпиля, использование которой не связано со сравнением (полным или частичным) позиции варианта перебора со встречавшимися ранее. Речь идет о так называемой *библиотеке правил*. В процессе развития теории принятия решения в любой области выработано множество методов, позволяющих оценивать создавшуюся ситуацию без дальнейшего формирования дерева перебора возможностей, а лишь на основании некоторых критериев, правил, имеющих характер строго доказанных теорем. Не являются исключением и шахматы. Чем больше таких правил знает мастер, тем больше вероятность скорейшего обрыва вариантов дерева перебора на основании этих правил.

При составлении любой программы, базирующейся на построении дерева перебора вариантов и минимаксной процедуре на этом дереве, перед составителями всегда возникает проблема, связанная с принудительным обрывом варианта дерева перебора по некоторым критериям до достижения предельной длины вариантов. Эта задача впервые решена в программе «Пионер». Здесь речь идет не об обрывах, связанных с целью игры (этим занимается подпрограмма поиска хода в оригинальной ситуации), а об обрывах вариантов по некоторым библиотечным правилам. Очевидно, что окончательные размеры дерева перебора тем меньше, чем большее число вариантов удалось оборвать таким образом. Естественным поэтому является желание, во-первых, выработать и формализовать достаточно большое количество таких критериев для обрывов и, во-вторых, использовать эти критерии в как можно большем числе узлов дерева перебора.

Однако время, требуемое для решения задачи, может сильно возрастать при использовании указанных критериев в большом количестве узлов дерева. Эффективность решения связана, следовательно, с размерами самого дерева перебора вариантов. Поскольку проверка критериев обрыва сопровождается некоторой, пусть даже незначительной (на узел), затратой времени, то становится ясной бессмысленность такой проверки при больших размерах дерева. Здесь речь идет, конечно, не о формальных отсечениях типа  $\alpha$ — $\beta$ -процедуры, а о критериях обрыва, связанных со специ-

фической задачи, в нашем случае с библиотечными шахматными правилами.

Очевидно, что для большинства существующих шахматных программ, при работе которых возникают (вследствие заложенных принципов полного перебора) громадные деревья вариантов, возможность эффективного использования библиотечных правил сомнительна.

Иная картина наблюдается при составлении программы «Пионер». Здесь мы имеем дело с небольшим, узким (что чрезвычайно важно) деревом перебора — порядка  $10^2$  узлов. При принятом алгоритме имеется возможность широкого использования библиотечных правил. Необходимо отметить и наглядно просматриваемую здесь обратную связь: именно эффективное использование правил (методов игры в различных стадиях шахматной партии) позволяет еще более сократить дерево перебора вариантов, что не только ускорит решение задачи, но и качественно повлияет на эффективность этого решения.

Таким образом, при решении неточных переборных задач по алгоритму М. Ботвинника проявляется взаимосвязь между небольшим деревом перебора и использованием опыта накопленных знаний о предмете исследования, в данном случае о шахматной игре.

**3.20. Критерии обрыва вариантов, основанные на «правиле квадрата».** За многовековую историю своего развития шахматная теория выработала огромное количество методов игры, всевозможных приемов, позволяющих сократить расчет вариантов.

При реализации шахматной программы «Пионер» целесообразным для повышения эффективности решения задачи является, как уже отмечалось, использование наиболее существенных из этих методов, формализованных в критерии для обрыва вариантов.

В качестве примера рассмотрим формализацию и расширение так называемого «правила квадрата» для пешечных окончаний. Нет необходимости разъяснять здесь суть шахматного правила квадрата. Оно подробно изложено в многочисленной специальной литературе.

Поскольку программа может играть любым цветом, то мы не будем говорить конкретно о белых или о черных, а лишь о стороне (+) и стороне (—). Примем следующие обозначения:

$Kp_{(+)}$  — король стороны (+);

$П_{(+)}$  — пешка стороны (+);

$Kp_{(-)}$  — король стороны (—);

$П_{(-)}$  — пешка стороны (—).

*Абсолютный критерий.* М а т е р и а л:  $Kp_{(+)}$ ,  $П_{(+)}$ ;  $Kp_{(-)}$ .

1. Если  $Kp_{(-)}$  не в квадрате \*  $П_{(+)}$ , то сторона (+) выигрывает.

2. Если  $Kp_{(-)}$  в квадрате  $П_{(+)}$  и ближе к  $П_{(+)}$ , чем  $Kp_{(+)}$ , то — ничья.

---

\* Здесь и далее при определении нахождения короля в квадрате пешки учитывается очередь хода. Например, в одной и той же позиции, в зависимости от очереди хода, король может находиться или не находиться в квадрате пешки (это граничный случай — король на границе квадрата).

Здесь и далее понятие «ближе» означает, что

$$|Kp(-)-\Pi(+)| < |Kp(+)-\Pi(+)|,$$

где  $|Kp(-)-\Pi(+)|$  и  $|Kp(+)-\Pi(+)|$  — расстояния в полуходах между соответствующими фигурами. Эти расстояния элементарно вычисляются по координатам фигур. Приведенная формула учитывает очередь хода.

*Первый критерий достаточности.* Материал:  $Kp(+)$ ,  $\Pi(+)$ ;  $Kp(-)$ ,  $\Pi(-)$ .

Пусть оба короля находятся в квадратах неприятельских пешек.

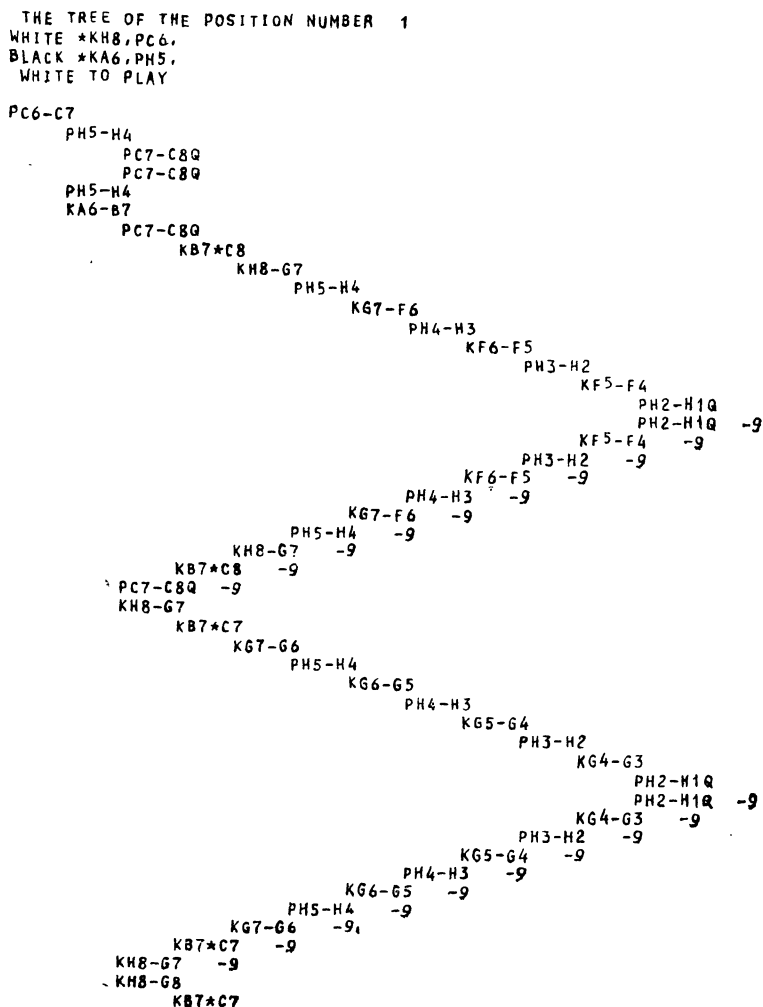


Рис. 44. Распечатка части дерева перебора при решении программой этюда Р. Рети до введения критериев обрыва вариантов, основанных на правиле квадрата

Для того, чтобы одной из сторон была гарантирована ничья, достаточно, чтобы король этой стороны находился ближе к неприятельской пешке, чем его оппонент (к той же пешке). Например, если

$$|Kp_{(+)} - P_{(-)}| < |Kp_{(-)} - P_{(+)}|$$

с учетом очереди хода, то стороне (+) ничья гарантирована.

Докажем это. Пусть выполняется приведенное неравенство. В этом случае, если сторона (+) будет делать ходы только королем в направлении неприятельской пешки так, чтобы расстояние между ними уменьшалось с каждым ходом (что возможно вследствие выполнения правила квадрата), то  $Kp_{(+)}$  беспрепятственно уничтожит пешку  $P_{(-)}$  до или в момент ее превращения. И тогда стороне (+) ничья гарантирована.

*Второй критерий достаточности.* Материал:  $Kp_{(+)}$ ,  $P_{(+)}$ ;  $Kp_{(-)}$ ,  $P_{(-)}$ .

Пусть

—  $Kp_{(-)}$  находится в квадрате  $P_{(+)}$ ;

—  $Kp_{(+)}$  находится не дальше от пешки  $P_{(+)}$  и от горизонтали ее превращения, чем  $Kp_{(-)}$ ;

—  $P_{(-)}$  находится не на вертикалях от  $Kp_{(+)}$  до  $P_{(+)}$  включительно;

—  $Kp_{(+)}$  и  $P_{(+)}$  — не на ладейных и коневых вертикалях.

Тогда: а) если  $P_{(+)}$  проходит в ферзи на один полуход раньше, чем  $P_{(-)}$ , то ничья гарантирована стороне (+); б) если  $P_{(+)}$  проходит в ферзи на один полуход позже, чем  $P_{(-)}$ , стороне (+) гарантирована ничья (если только  $P_{(-)}$  не превращается в ферзя с шахом и если новоявленный  $\Phi_{(+)}$  не уничтожается посредством «вилочного шаха»). Простое доказательство справедливости этого критерия громоздко по объему, поэтому мы его не приводим.

Следует отметить, что понятия «находится в квадрате», «находится не дальше», «находится не на вертикалях», «проходит в ферзи», «раньше», «превращается в ферзя с шахом», «уничтожается посредством вилочного шаха», и т. п. в программе четко формализованы и определяются с учетом очереди хода.

В результате введения в программу «Пионер» критериев (обрыва вариантов), основанных на расширенных правилах квадрата, дерево решения этюда Р. Рети (см. рис. 13) приняло «человеческие» очертания. До введения в программу этих критериев «Пионер» продолжал бессмысленный перебор даже в тех позициях, где шахматист давно оборвал бы вариант (см. рис. 44).

Необходимо отметить, что программа в некоторых случаях обрывает вариант после хода черных, но конечная оценка учитывает предполагаемый ответ белых. Поясним это на примере одной из позиций дерева перебора, построенного «Пионером» при решении этюда Р. Рети.

В позиции на рис. 45 после 3... Krb6: c6 шахматист оборвал бы вариант из-за ответа 4. Kpe5 — f4, программе же даже не надо совершать этот ход, так как абсолютный критерий уже действует. Точно так же, в ответ на 3. ... h4—h3 — ничья по второму критерию достаточности, а шахматист видит 4. Kpe5 — d6.

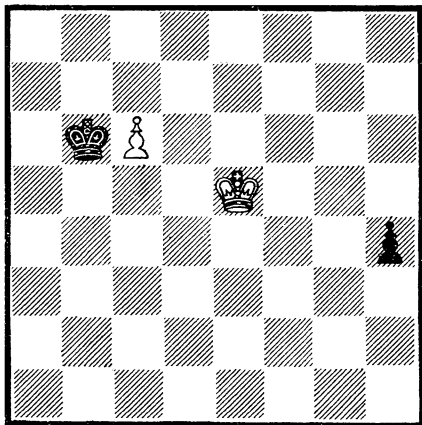


Рис. 45. Позиция из дерева перебора решения программой «Пионер» этюда Р. Рети (Белые : Кр h 8, п. с. 6; Черные : Кра6, п. h5. Ничья)

**3.21. Заключение.** Таким образом, при создании для ЭВМ шахматной программы, моделирующей мышление шахматного мастера, одной из важнейших является задача создания информационно-справочной системы «Опыт прошлого», а также разработка алгоритмов, позволяющих программе эффективно использовать этот опыт.

Есть основания полагать, что использование методов справочного и стремления, а также библиотечного обрыва вариантов аналогично тому, как это делает шахматный мастер (и программа «Пионер»), может найти применение и в других практических областях управления, где, подобно шахматам, возникают неточные задачи переборного типа.

## ПРИЛОЖЕНИЕ 4

### АССОЦИАТИВНАЯ БИБЛИОТЕКА ФРАГМЕНТОВ

**А. И. Резницкий, А. Д. Юдин**

Использование справочного метода, рассмотренного в приложении 3, не может быть эффективным в середине игры, поскольку в этой стадии партии одинаковые позиции возникают крайне редко. Более подходящим для миттельшпиля представляется метод, способствующий принятию решения о выборе хода в создавшейся позиции по аналогии (ассоциации) с ранее встречавшимися.

Метод получения информации, основанный на ассоциации по сходству анализируемой позиции и позиции, сведения о которой занесены в библиотеку, был назван *ассоциативным*, а сама библиотека, хранящая признаки сходства позиции, — *ассоциативной*.

Перспективность использования шахматными алгоритмами опыта прошлого по ассоциативному методу была отмечена еще К. Шенноном [2]. Он приводил в пример шахматного мастера, который «знает сотни, а может быть, и тысячи стандартных позиций, привычных комбинаций и типовых маневров, которые встречаются неоднократно в партиях. Имеются, например, стандартные жертвы коня на f7 или слона на h7, стандартные маты, например мат Филдора, маневры, связанные с вилками, превращениями и т. д. В данной позиции он усматривает много сходства со знакомыми ему случаями, и это направляет его мысль на исследование тех вариантов, в которых вероятность успеха наибольшая».

В попытке шахматиста применить в игре опыт прошлого к настоящему, найдя между ними некоторое сходство, и проявляется ассоциативность его мышления.

Как уже указывалось, в алгоритме М. Ботвинника математическое отображение позиции создается параллельно с построением дерева перебора вариантов. При этом эффективность расходования ресурсов во многом зависит от выбора направления перебора. Если накопленный опыт позволит в позиции, похожей на те, которые уже встречались ранее, получить выгодное расположение фигур, провести стандартную комбинацию или маневр, перебор других вариантов может не потребоваться (они будут отсечены). Таким образом, существующая возможность аналогичных действий в сходных ситуациях может быть использована для сокращения перебора. Эту идею было решено разработать в рамках программы «Пионер» и положить в основу алгоритма, действующего в позициях миттельшпиля и сложного эндшпиля совместно с алгоритмом поиска хода в оригинальной ситуации.

Для формализации понятия сходства шахматных позиций был разработан метод фрагментов. Суть метода заключается в следующем.

Из позиции, в которой в прошлом с выгодой была осуществлена определенная типовая идея (маневр, комбинация), выделяется некоторая ее часть (фрагмент), включающая те фигуры, присутствие которых в анализируемой позиции существенно для реализации данной идеи. Этот фрагмент заносится в библиотеку, и в дальнейшем все позиции (из перебора), его содержащие, считаются сходными с той, из которой был выделен указанный фрагмент. Предполагается, что в каждой такой позиции могут быть условия для реализации той же идеи.

Приведем основные положения метода фрагментов.

Во фрагмент включаются лишь те фигуры, без которых проведение типовой идеи невозможно. Назовем эти фигуры *действующими*.

Действующие фигуры можно разделить на две группы. Фигуры, названные *фиксированными*, должны быть расположены на однозначно определяемых полях, называемых полями фиксации. Для других фигур должны существовать траектории (в пределах установленного горизонта) к другим заданным полям. Назовем эти поля полями привязки, а фигуры — *привязанными*.

Наименование некоторых действующих фигур может быть задано неоднозначно (например, в тех случаях, когда единственная функция этой фигуры — блокада траектории другой действующей фигуры).

Деление действующих фигур на фиксированные и привязанные связано с тем, что лишь одна сторона может быть заинтересована в осуществлении типовой идеи. В процессе подготовки и проведения данной идеи эта активная сторона может перемещать свои фигуры. Поэтому действующие фигуры, принадлежащие активной стороне, считаем привязанными, их расположение характеризуется



достаточно большой степенью свободы. Действующие фигуры противоположной стороны считаем фиксированными, их расположение поддается только проверке.

В зависимости от расстояния привязанных фигур фрагмента до соответствующих полей привязки различаем фрагменты *исходные* и *близкие к исходным*. Исходный фрагмент характеризуется таким расположением привязанных фигур, при котором каждая из них находится на заданном расстоянии от своего поля привязки. Данное расстояние указывается в описании фрагмента для каждой привязанной фигуры в отдельности и называется расстоянием минимального удаления от поля привязки.

Пусть  $n_i$  — минимальное удаление  $i$ -й фигуры от поля привязки (указанное в описании),  $l_i$  — реальное удаление  $i$ -й фигуры от поля привязки в анализируемой позиции (число полуходов, которое требуется для того, чтобы фигура заняла поле привязки),  $k$  — число привязанных фигур фрагмента. Для того чтобы позиция содержала исходный фрагмент, необходимо, чтобы, во-первых, фиксированные фигуры были расположены на своих полях фиксации и, во-вторых, чтобы для каждого  $i$   $l_i = n_i$ ,  $i = 1, 2, \dots, k$ .

Далее, пусть  $H_L$  — предельный горизонт,  $D$  — заданное натуральное число.

Если все фиксированные фигуры — на своих полях фиксации и выполнена система неравенств

$$\begin{cases} n_i \leq l_i \leq H_L, & i = 1, 2, \dots, k, \\ \sum_{i=1}^k (l_i - n_i) \leq D, \end{cases}$$

считаем, что в позиции содержится фрагмент, близкий к исходному.

Сумма  $\sum_{i=1}^k (l_i - n_i)$  характеризует отклонение близкого фрагмента от исходного, т. е. минимальное время (в полуходах), за которое может быть получен исходный фрагмент. Эту сумму назовем *глубиной ассоциации фрагмента* в анализируемой позиции. (Заметим, что для исходного фрагмента значение этой величины равняется нулю.) Величина  $D$ , названная предельной глубиной ассоциации, является ограничением поиска. Значение этой величины можно варьировать в зависимости от наличия ресурсов.

Роль ассоциативного метода в программе «Пионер» заключается в том, что этот метод позволяет определить приоритет перебора вариантов в тех ступенях системы управления, в которых играют фигуры фрагмента, т. е. указать направление перебора. Приоритет включения фрагмента в перебор зависит от стоимости мишени той зоны, в которой ведется игра во фрагменте (эта стои-

мость указывается в описании фрагмента), и от глубины ассоциации фрагмента в анализируемой позиции. Таким образом, фрагменты дополняют систему приоритетов включения зон в перебор, существующую в подпрограмме поиска хода в оригинальной ситуации.

В позициях, содержащих фрагмент, близкий к исходному, активная сторона может стремиться получить исходный фрагмент. С этой целью среди найденных привязанных фигур отмечаются те, которые расположены от своих полей привязки на расстоянии, превышающем минимальное удаление. Для этих фигур вычисляются и записываются в МО траектории, ведущие к полям привязки. Зоны, соответствующие указанным комлевым траекториям, включаются в игру (см. приложение 1). Приоритеты ходов по траекториям включенных зон определяются в соответствии с системой приоритетов, принятой в программе «Пионер».

Предварительное движение фигур по траекториям этих включенных зон или игра в связанных с ними зонах может продолжаться, пока будет достигнут исходный фрагмент либо выяснится, что он не может быть получен по одной из причин:

- нарушилось расположение фиксированных фигур;
- уничтожена какая-либо привязанная фигура;
- траектории некоторой привязанной фигуры, ведущие к ее полю привязки, не могут быть деблокированы.

При выполнении хотя бы одного из перечисленных условий все зоны, включенные для организации стремления к исходному фрагменту, застывают, что эквивалентно застыванию при потере связи с действующей траекторией (см. приложение 1).

Для включения в перебор исходного фрагмента среди привязанных фигур выделяется та, которой предстоит совершить первый ход. Ходу «по фрагменту» дается повышенный приоритет. При включении фрагмента в перебор делается ход выделенной фигурой на поле привязки или по траектории, ведущей к этому полю, если для указанной фигуры расстояние минимального удаления больше единицы.

Итак, во фрагменте заложена следующая информация (речь идет об исходных фрагментах, составляющих библиотеку).

1. Фиксированные фигуры. Для каждой фиксированной фигуры указывается наименование фигуры и поле фиксации.

2. Привязанные фигуры. Каждая привязанная фигура определяется наименованием фигуры, полем привязки и минимальным удалением от поля привязки.

3. Привязанная фигура, совершающая первый ход.

4. Возможные величины сдвига (сдвигом — вертикальным или горизонтальным — называется изменение координат всех полей фиксации и привязки на одно и то же число, называемое величиной сдвига).

## 5. Стоимость мишени.

Создание библиотеки фрагментов оправдано лишь в том случае, когда обращение к ней позволит в среднем сократить время поиска сильного хода в позициях, содержащих библиотечные фрагменты. Поэтому одно из требований к подпрограмме поиска фрагментов состоит в том, чтобы время ее работы было сравнительно невелико по отношению к среднему времени анализа позиции.

Для экономии времени в подпрограмме организован ступенчатый поиск. На каждой ступени проверяются определенные условия, и дальнейший поиск ведется только среди тех фрагментов, которые удовлетворяют этим условиям. Ступени поиска можно представить себе фильтрами, пропускающими только те фрагменты, присутствие которых в анализируемой позиции возможно (с точки зрения данного фильтра).

Фрагменты, прошедшие очередной фильтр, называются претендентами. Их номера в библиотеке образуют список претендентов. На каждом этапе поиска через фильтр пропускаются все оставшиеся претенденты, одна их часть забраковывается, а другая образует новый список претендентов. На первый фильтр поступает вся библиотека, последний пропускает только присутствующие в анализируемой позиции фрагменты. Фильтры располагаются в порядке увеличения времени, необходимого для их работы, с тем чтобы на фильтр с наибольшим временем «фильтрации» поступало минимальное число фрагментов-претендентов. При такой организации подпрограммы возникает возможность отобрать присутствующие в позиции фрагменты за минимальное время.

Проследим работу алгоритма по блок-схеме обращения к ассоциативной библиотеке фрагментов, представленной на рис. 46.

Информация, хранящаяся в библиотеке, соответствует тому случаю, когда присутствие на доске фрагмента выгодно белым. Подпрограмма распознавания действует за определенный цвет и ищет только те фрагменты, которые выгодны этому цвету. Поэтому, если предстоит искать фрагменты за черных, что устанавливается проверкой А, то производится симметрия цвета (процедура В).

В процедуре С анализируемая позиция отражается относительно вертикальной оси доски, заготавливается флангово-симметрированная позиция (отражение). Полученное отражение используется в дальнейшем аналогично анализируемой позиции, т. е. проверяется присутствие библиотечных фрагментов как в самой позиции, так и в ее отражении. Если фрагмент содержится в отражении, то найденные  $\alpha$ -поля траекторий привязанных фигур подвергнутся фланговому симметрированию.

Процедура D — первый фильтр. Здесь проверяется, присутствуют ли на доске все действующие фигуры. Образуется первоначальный список претендентов.

Далее управление передается процедуре Е — второму фильтру. Определяется, расположены ли в анализируемой позиции фиксированные фигуры на заданных полях фиксации. Список претендентов уменьшается.

Процедура F организует поиск привязанных фигур; определяется окончательный список фрагментов-претендентов, т. е. тех фрагментов, которые содержатся в анализируемой позиции. Для каждого фрагмента из этого списка найдены: 1)  $\alpha_0$ - и  $\alpha_k$ -поля траекторий привязанных фигур, находящихся от поля привязки на расстоянии, превышающем минимальное удаление; 2) глубина ассоциации фрагмента в анализируемой позиции.

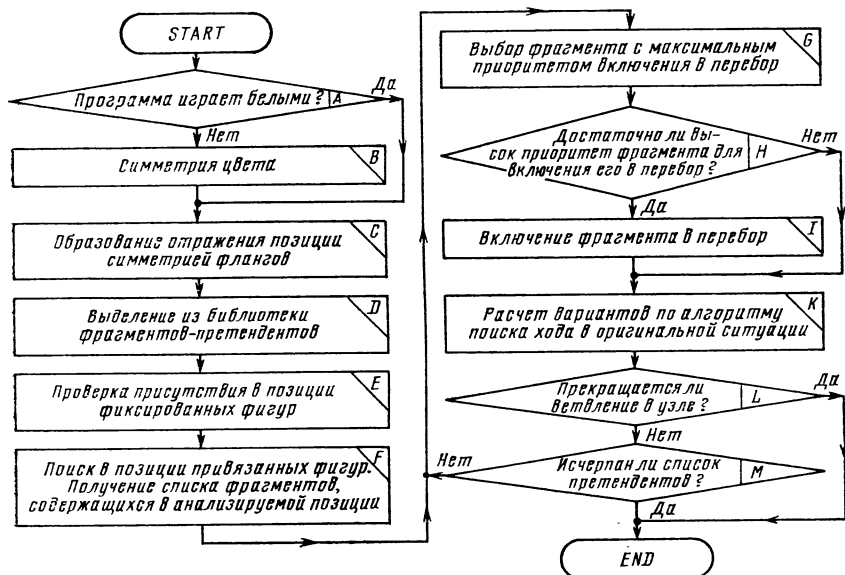


Рис. 46. Блок-схема обращения к ассоциативной библиотеке фрагментов

Затем управление передается процедуре G, которая из окончательного списка претендентов выбирает фрагмент с максимальным приоритетом включения в перебор. Из этого списка выбирается фрагмент, для которого значение функции  $F=(10M-D_a)$  максимально, где  $M$  — стоимость мишени, а  $D_a$  — глубина ассоциации фрагмента в анализируемой позиции.

После того, как фрагмент выбран, решается вопрос о его включении в перебор в соответствии с системой приоритетов, принятой в программе «Пионер» (процедура H). Если игра во фрагменте имеет наивысший приоритет, управление передается процедуре

I, которая осуществляет связь алгоритма распознавания фрагментов с алгоритмом поиска хода в оригинальной ситуации. Здесь сообщается повышенный приоритет ходу «по фрагменту» (для исходного фрагмента) или включаются в игру траектории привязанных фигур, проходящие через найденные  $\alpha$ -поля (для фрагмента, близкого к исходному). Таким образом, процедура I определяет направление перебора.

Далее управление передается подпрограмме поиска хода в оригинальной ситуации (процедура K). Определяется оценка оптимального варианта. Если эта оценка достаточна для прекращения ветвления в том узле дерева перебора, в котором возникла анализируемая позиция (что выясняется проверкой L), библиотека в данном узле больше не нужна — обращение к библиотеке заканчивается. В противном случае выясняется, не исчерпан ли для этого узла список претендентов (проверка M), и если нет, — управление снова передается процедуре G, и в перебор может быть включен очередной фрагмент. Итак, в некотором узле дерева перебора цикл G — M может повторяться либо до принятия решения о прекращении ветвления в этом узле, либо до исчерпания фрагментов, присутствующих в анализируемой позиции.

В качестве иллюстрации рассмотрим типовую комбинацию, известную уже более 300 лет. Комбинация начинается жертвой слона за пешку на поле h7. С помощью этой жертвы атакующая сторона вскрывает положение короля, затем совместными действиями коня и ферзя создает матовые угрозы, отразить которые часто бывает невозможно.

Позиция одной из партий, в которых встретилась эта комбинация, приведена на рис. 47. После 16. e4 — e5 Kf6 — d5 17. Kc3 : d5 e6 : d5 18. Cg5 : e7 Kc6 : e7 были созданы все условия для проведения комбинации: 19. Cb1 : h7 + Kpg8 : h7 20. Kf3 — g5 + Kph7 — g6. (На отступление короля 20 ... Kpg8 последовало бы 21. Фh5 с быстрым выигрышем.) 21. Фd1 — g4. Белые получили выигранную позицию и за несколько ходов довели партию до победы.

Определим исходный фрагмент, соответствующий этой комбинации.

### 1. Фиксированные фигуры:

Наименование фигуры	Поле фиксации
Король	g8
Пешка	g7
Пешка	h7

### 2. Привязанные фигуры:

Наименование фигуры	Поле привязки	Минимальное удаление от поля привязки
Слон	h7	1
Конь	g5	1
Ферзь	h7	2

3. Привязанная фигура, совершающая первый ход, — слон.

4. Возможная величина сдвига — 0 (сдвиг невозможен).

5. Стоимость мишени — 200 (стоимость короля, принятая в программе «Пионер»).

Одно из возможных расположений фигур в исходном фрагменте изображено на рис. 48.

В позиции на рис. 47 присутствует фрагмент, близкий к исходному. Белые могут получить исходный фрагмент, деблокируя  $\beta$ -поле  $e4$  траектории слона и  $\alpha$ -поле  $g5$  траектории коня. Для этого требуется минимум два полухода, следовательно, глубина ассоциации фрагмента в позиции на рис. 47 равняется двум. После ходов  $e4-e5$  и  $Sg5:e7$  белые получают позицию, содержащую исходный фрагмент. В перебор включается первый ход «по фрагменту»  $Sb1:h7+$ , приводящий к успеху.

Так метод фрагментов позволяет определить направление перебора вариантов.

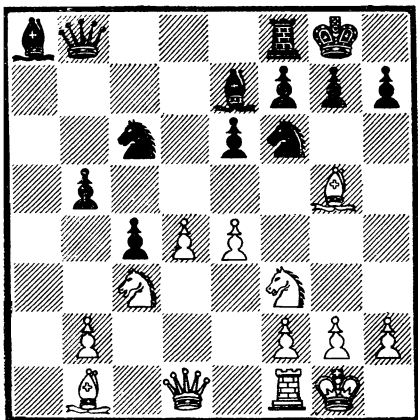


Рис. 47. Позиция из партии Шлехтер-Вольф

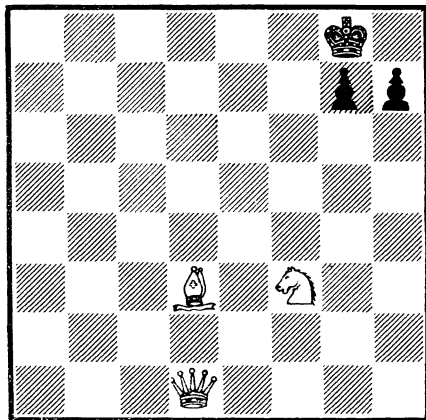


Рис. 48. Одно из возможных расположений фигур в исходном фрагменте

В заключение отметим, что проблема принятия решения в сходных ситуациях актуальна для многих задач управления. Методы, применяемые при моделировании ассоциативного мышления шахматиста, вероятно, могут быть использованы в различных областях народного хозяйства, имеющих практическое значение.

## СПИСОК ЛИТЕРАТУРЫ

1. Ботвинник М. М. Алгоритм игры в шахматы. — М.: Наука, 1968.
2. Шеннон К. Работы по теории информации и кибернетике. Сб. статей. — М.: ИЛ, 1963.
3. Ботвинник М. М. О кибернетической цели игры. — М.: Сов. радио, 1975.
4. Юдин А. Д. Библиотека эндшпиля ЭВМ. — Шахматы в СССР, 1975, № 7, с. 10—11.
5. Штильман Б. М. Машина учится. — Шахматы в СССР, 1976, № 4, с. 20—22.
6. Юдин А. Д. Программа поиска информации в двумерной таблице с субординацией входов. (Библиотека позиций эндшпиля). — Программирование, 1976, № 4, с. 66—72.
7. Stilman B. Zwei Arbeiten zum Botwinnik'schen Schachalgorithmus. — Abt. Informatic Universitat Dortmund, 1976, B29, s. 1—55. См. также Деп. ВИНТИ 3947—76 и 3947а—76/ Б. М. Штильман. О программе формирования зоны игры. Дерево перебора в зоне игры.
8. Адельсон-Вельский Г. М. и др. О программировании игры вычислительных машин в шахматы. — УМН, т. 25, вып. 2 (152), 1970, с. 221—260.
9. Капабланка Х. Р. Учебник шахматной игры. — М.: ФИС, 1975.
10. Ботвинник М. М., Штильман Б. М., Юдин А. Д. Искусственный шахматный мастер. — Вестник АН СССР, 1978, № 4, с. 82—91.
11. Адельсон-Вельский Г. М., Арлазаров В. Л., Донской М. В. — Программирование игр. — М.: Наука, 1978.

## СЛОВАРЬ ТЕРМИНОВ

- Антистремление** — снижение приоритетов ходов по включенным в МО траекториям, ведущим из благоприятной (по оценке) позиции в библиотечную с невыгодной оценкой.
- Ассоциативное мышление (шахматное)** — использование в создавшейся ситуации (позиции) известных типовых идей; проявляется в попытке применить опыт прошлого к настоящему.
- Видение** — факт существования траектории, ведущей с поля, на котором стоит фигура одного цвета, на поле, где стоит фигура другого цвета. В процессе перебора фигуры меняют свои места — появляются новые видения.
- Включение**  
зоны в игру — разрешение на передвижение фигур по траекториям данной зоны в данном поддереве перебора;  
исходного фрагмента в перебор — включение в МО траектории определенной «привязанной» фигуры.
- Глубина**  
ассоциации фрагмента, близкого к исходному — минимальное время (в полуходах), за которое может быть получен исходный фрагмент; величина отклонения близкого фрагмента от исходного;  
включения зоны — глубина узла, ниже которого в поддереве данная зона считается включенной;  
застывания — глубина узла, в котором произошло застывание данной траектории из-за потери связи с действующей траекторией;  
узла в дереве перебора — длина ветви дерева (в полуходах), ведущей из исходного узла к данному.
- Горизонт**  
переменный ( $H_x$ ) — время (в полуходах), отводимое для контроля (блокады) полей комлевой траектории зоны; зависит от количества  $\alpha$ -полей между атакующей фигурой и контролируемым полем;  
предельный ( $H_L$ ) — наибольшее разрешенное время (в полуходах), необходимое данной фигуре для перемещения по траектории при условии, что каждый раз, получая очередь хода, данная сторона перемещает лишь данную фигуру по данной траектории; по существу горизонт задает предельную длину регистрируемых траекторий.
- Движение по рельсам** — процедура обхода ветвей поддерева ниже данного узла и сбора информации для решения вопроса о включении зоны в игру в этом узле.
- Двумерная таблица с субординацией входов** — таблица с двумя входами, один из которых является независимым, а другой зависит от первого.
- Дерево игры** — граф, каждая вершина (узел) которого имеет одну непосредственно предшествующую ей (родительскую) вершину (узел), за исключением выделенной (исходной) вершины, называемой корнем дерева, которая не имеет предшествующих вершин.
- Задача**  
неточная — переборная задача, неразрешимая как точная, которая, однако, может быть приближенно решена формированием усеченного дерева перебора (при неточной цели игры);



точная — переборная задача, которая может быть решена с помощью формирования полного дерева перебора и точной цели игры.

#### Застывание траекторий

при недостаточной величине параметра  $T_x$  — запрет на передвижение фигуры по траектории отрицания зоны из-за нехватки времени на подход к «полю боя»;

при потере связи с действующей траекторией — запрет на передвижение фигуры по траектории, так как траектория, с которой связана эта последняя, удовлетворяет следующим условиям: либо фигура по ней прошла, либо никогда уже не пройдет при продолжении варианта, либо она сама застыла при потере связи с действующей траекторией;

при сходе с  $\alpha_0$ -поля на подъеме по дереву — запрет на передвижение фигуры по траектории, так как траектория, с которой связана эта последняя, удовлетворяет условиям: либо соответствующая фигура сошла с ее  $\alpha_0$ -поля при подъеме по дереву, либо эта траектория сама связана с застывшей в этом же смысле траекторией.

Зона игры — совокупность фигур (белых и черных) и их траекторий, объединенных в поддержке и противодействии атакующей фигуре.

Информационно-справочная система «Опыт прошлого» — комплекс подпрограмм, содержащий библиотеки дебютов, технического эндшпиля, правил, а также ассоциативную библиотеку фрагментов и алгоритмы пользования этими библиотеками; запас шахматных знаний, которыми обладает программа «Пионер».

Класс — часть библиотеки технического эндшпиля, характеризующаяся определенным соотношением материала в позициях.

Конфигурация — характеристика относительного расположения фигур позиции (силуэт позиции).

«Краевой эффект» — явление, позволяющее разбить множество позиций данной конфигурации на несколько непересекающихся подмножеств; оценки и первые ходы, решающие позиции, неизменны внутри каждого такого подмножества.

Массив  $15 \times 15$  для данной фигуры — квадратная таблица размером  $15 \times 15$ , в центральном поле которой записан нуль; на остальных полях записаны числа, равные числу передвижений данной фигуры с центрального поля массива до данного поля по кратчайшему пути.

Математическое отображение позиции (МО) — совокупность зон игры, соответствующих позиции в данный момент игры (перебора).

#### Метод

видения — метод поиска новых траекторий и зон в процессе перебора ходов путем регистрации видений;

граней и оценок ( $\alpha$ — $\beta$ -отсечение) — метод сокращения дерева перебора путем отсечения некоторых ветвей, при котором получается поддерево, эквивалентное несокращенному в том смысле, что в нем сохраняется прежний оптимальный вариант;

фрагментов — метод, используемый при ассоциативном мышлении и основанный на том, что для реализации типовой идеи существенно соответствующее расположение в анализируемой позиции фигур, без которых проведение типовой идеи совершенно невозможно;

Шеннона первый — метод поиска хода (в шахматной позиции) путем формирования дерева перебора всех вариантов на фиксированную глубину, оценки этих вариантов с помощью оценочной функции и выбора оптимального варианта путем минимаксной процедуры;

Шеннона второй — то же, что первый, только в дерево перебора включаются лишь некоторые варианты, имеющие смысл, при этом предельная глубина перебора значительно возрастает.

Мишень —  $\alpha_k$ -фигура зоны нападения

поражаемая — уязвимая мишень, причем комлевая траектория в одно передвижение и очередь хода за данной стороной;  
уязвимая — мишень в зоне нападения, когда все  $\alpha$ -поля комлевой траектории и траекторий отступления  $\alpha_k$ -фигуры находятся под контролем атакующей стороны.

#### Несовпадение

максимально допустимое ( $\sigma_{\max}$ ) — число, ограничивающее количество несовпадающих фигур двух позиций равного материала, разрешающее эти позиции считать близкими;

позиций равного материала — число несовпадающих фигур;

шаблонов пешек — факт, позволяющий исключить все множество позиций данной конфигурации из списка претендентов на стремление.

#### Отсечение ветвей

при ветвлении в пучке траекторий — прекращение дальнейшего рассмотрения в данном узле дерева перебора ходов по траекториям данного пучка;

при «движении вперед» — априорное отбрасывание некоторых ходов в каждой позиции при спуске по дереву — см. метод Шеннона (второй);  
в программе «Пионер» — это ходы не по траекториям;

при «движении назад» — прекращение ветвления в узле в процессе подъема по дереву, если определено, что дальнейшее ветвление в нем не изменит результата минимакса.

Оценочная функция — функция, сопоставляющая каждой позиции оценку — действительное число; оценки заключительных позиций вариантов перебора используются для сравнения вариантов путем минимаксной процедуры.

Параметр  $T_x$  — 13-й параметр следа, управляющий поиском траекторий отрицания и одним из видов застывания; измеряется числом передвижений по траектории и зависит от времени, отводимого данной фигуре в данный момент игры в зоне для перемещения по некоторой траектории отрицания зоны; по сути дела, распределяет время, задаваемое переменным горизонтом  $H_x$  между траекториями отрицания разных степеней.

#### Параметры

следа — 13 элементов списка параметров следа лучка траекторий, содержащего информацию о роли этого пучка в МО;

столбца — 8 элементов списка параметров, содержащего информацию о данном узле дерева перебора.

Передвижение — единица измерения длины траектории.

#### Подсистема

первой ступени — фигура со своей траекторией;

второй ступени — зона игры;

третьей ступени — математическое отображение (полная система).

Позиционная оценка — пропорциональна отношению  $K_w/K_b$ , где  $K_w$  и  $K_b$  — число полей траекторий, контролируемых белыми и черными соответственно

#### Позиция

анализируемая (исходная) — позиция из партии или из варианта перебора, представленная программе для анализа;

библиотечная — позиция, заложенная в библиотеку технического эндшпиля непосредственно или при помощи формул разбиения;

близкая к исходной — позиция равного материала с исходной; при этом не превышены значения максимально допустимого несовпадения ( $\sigma_{\max}$ ) и максимально допустимой разности ( $\Delta_{\max}$ ) между этими позициями;

претендент на стремление — библиотечная позиция, к которой, как предполагается на данном этапе поиска, впоследствии будет организовано стремление из исходной;

равного материала с исходной — позиция (библиотечная), которая имеет шаблон, в точности совпадающий с шаблоном исходной позиции;

символ множества данной конфигурации — позиция, непосредственно заложенная в библиотеку технического эндшпиля, по которой (при помощи

формул разбиения) можно получить любую позицию множества, характеризующегося данной конфигурацией.

#### Поиск решения переборной задачи

- в оригинальной ситуации — поиск решения путем формирования дерева перебора;
- по ассоциации — поиск решения путем направленного формирования дерева перебора в ситуации, похожей на ту, которая давала благоприятный результат в прошлом;
- по методу стремления — поиск решения путем направленного формирования дерева перебора, когда есть надежда получить точно такую же ситуацию, которая давала благоприятный результат в прошлом;
- по справочному методу — поиск решения без формирования дерева перебора, если данная ситуация совпадает с той, которая встречалась в прошлом, и оценка ее известна.

#### Поле

- $\alpha$ -поле траектории, на котором фигура останавливается при своем движении по траектории;
- $\alpha_0$ -начальное поле траектории;
- $\alpha_k$ -конечное поле траектории;
- $\beta$ -поле траектории, которое фигура проходит без остановки;
- привязки — конечное поле траектории привязанной фигуры;
- фиксации — поле, на котором расположена фиксированная фигура.

Полуход — время, затрачиваемое на ход одной стороны; единица измерения шахматного времени при движении по траекториям; единица измерения длины вариантов перебора.

Правило обрыва вариантов — критерий, заложенный в библиотеку правил, позволяющий оборвать вариант перебора с соответствующей точной оценкой.

Предельная длина варианта (глубина усечения дерева) — число, условно ограничивающее длину вариантов в дереве перебора (не смешивать с горизонтом  $H_L$ ).

#### Принцип

- надежды — всякая возможность изучается лишь до тех пор, пока с ней связана надежда на что-то хорошее (на достижение цели);
- наибольшего выигрыша — новая возможность включается в рассмотрение лишь в том случае, если есть надежда получить что-то лучшее по сравнению с уже рассмотренным;
- своевременности — рассматривать надо лишь те возможности, при которых связанные с ними объекты могут успеть принять участие в игре.

Приоритет — очередность включения ходов в дерево перебора; определяется функцией — многочленом первой степени нескольких переменных.

Прогноз — оптимальный вариант; прогнозирование того, насколько хорошо может быть реализована цель неточной игры.

Псевдоперебор — подъем без оценки по ветви дерева (соответствующей текущему варианту перебора), предпринятый только для возможного включения в перебор новых траекторий зоны.

#### Пучок траекторий

- с  $\alpha_0$ -поля на  $\alpha_k$ -поле данной длины и для данной фигуры — совокупность траекторий этой фигуры с  $\alpha_0$ -поля на  $\alpha_k$ -поле, причем таких, что число передвижений фигуры по ним на свободной доске не превышает длины пучка;

отступления или деблокады — совокупность всех траекторий данной фигуры в одно передвижение.

Разворачивание пучков — операция, состоящая в получении полной информации о пучках из их следов.

Размен оптимальный — последовательность взятий без пауз на данном поле, определяемая минимаксной процедурой среди всех таких последовательностей.

Размораживание траекторий — процедура, обратная застыванию; это отмена застывания при возврате в узел дерева перебора, где произошло застывание соответствующего типа.

Разность

максимально допустимая ( $\Delta_{\max}$ ) — число, ограничивающее величину разности двух позиций равного материала, позволяющее эти позиции считать близкими;

позиций равного материала ( $\Delta_{\Delta\alpha}$ ) — число, равное сумме длин траекторий, ходы по которым переводят одну позицию (исходную) в другую (библиотечную).

Регистрация видения — запись информации о траектории, соответствующей видению, т. е. запись следа пучка на  $\alpha_0$ -поле пучка, в частности запись роста зоны.

Росток зоны — след пучка комлевых траекторий зоны на  $\alpha_0$ -поле пучка.

Сдвиг — преобразование позиции, характеризующееся изменением всех координат полей фиксации и привязки действующих фигур на одну и ту же величину, называемую величиной сдвига.

Симметрия

диагональная — преобразование, характеризующееся отражением позиции относительно диагоналей  $a1-h8$  или  $h1-a8$ ; действует в позициях без пешек;

фланговая — преобразование, характеризующееся отражением позиции относительно вертикальной оси доски;

цвета — преобразование, характеризующееся отражением позиции относительно горизонтальной оси доски, сопровождающееся переменной цвета фигур на противоположный.

Система управления — система, выполняющая функции получения информации, ее переработки и исполнения решения.

След пучка на данном поле — ячейка цепного списка, привязанного к данному полю, содержащая информацию о некотором пучке траекторий, проходящих через это поле.

Стирание информации о зонах — стирание следов соответствующих пучков траекторий, т. е. присоединение этих ячеек к цепочке «пустых» ячеек и замыкание цепочки заполненных.

Стремление

к библиотечной позиции — включение в МО планируемых траекторий фигур активной стороны, ходы по которым ведут из исходной позиции в близкую библиотечную с благоприятной оценкой; реализуется методом стремления;

к исходному фрагменту — включение в МО траекторий, ведущих к полям привязки для тех привязанных фигур, которые расположены от своих полей привязки на расстоянии, превышающем минимальное удаление; реализуется при ассоциативном методе.

Типовая идея — маневр, комбинация, реализация которых в прошлом принесла успех.

Типы зон — нападения, блокады, контроля, отступления и деблокады.

Траектория

вилочная — траектория, часть которой является общей для нескольких траекторий той же фигуры;

комлевая — основная траектория зоны, с которой начинается формирование зоны; длина этой траектории ограничена предельным горизонтом  $H_L$ ;

кратчайшая данной фигуры с  $\alpha_0$ -поля на  $\alpha_k$ -поле — траектория, которая имеет наименьшее число передвижений на свободной от фигур доске среди всех траекторий этой фигуры, соединяющих поля  $\alpha_0$  и  $\alpha_k$ ;

отрицания — некомлевая траектория зоны; длина ее ограничена из-за переменного горизонта  $H_x$ ;

отрицания 1-й степени — связана с комлевой траекторией зоны;  
отрицания  $n$ -й степени — траектория отрицания, связанная с траекторией отрицания  $(n-1)$ -й степени;  
планируемая — траектория, ходы по которой ведут из исходной позиции к близкой библиотечной;  
простая — кратчайшая траектория для всех фигур, кроме ферзя; для ферзя любая траектория в два передвижения на свободной доске;  
стыкованная — траектория, составленная из двух простых.

**Удаление фигуры от поля привязки**

минимальное — длина траектории привязанной фигуры до соответствующего поля привязки в исходном фрагменте;

реальное — то же в анализируемой позиции.

**Улучшение результатов перебора** — принятие решения о возможном расширении МО после анализа построенного поддерева ниже данного узла, если есть надежда на получение в данном узле нового текущего оптимального варианта (ТОВ) с более высокой оценкой.

**Фигура**

действующая — фигура, наличие и соответствующее расположение которой в анализируемой позиции необходимо для реализации типовой идеи;

комлевая — фигура, которой принадлежит комлевая траектория зоны;

отрицающая — фигура, которой принадлежит траектория отрицания;

привязанная — вид действующей фигуры; должна иметь траекторию заданной длины (но не более предельного горизонта) к соответствующему полю привязки;

фиксированная — вид действующей фигуры; должна быть расположена на своем поле фиксации.

(+) — фигура (в зоне) одного цвета с комлевой фигурой зоны;

(—) — фигура (в зоне) другого цвета, нежели фигуры (+);

$\alpha_0$ -фигура, расположенная на  $\alpha_0$ -поле своей траектории;

$\alpha_k$ -фигура, расположенная на  $\alpha_k$ -поле комлевой траектории зоны нападения.

**Формулы разбиения** — формулы, позволяющие воспроизвести любую позицию множества данной конфигурации по соответствующей позиции-символу.

**Фрагмент** — совокупность действующих фигур, содержащаяся в анализируемой позиции, необходимая для реализации типовой идеи;

близкий к исходному — фрагмент такой, что за некоторое число ходов, не превышающее предельную глубину ассоциации, может быть получен исходный фрагмент;

исходный — фрагмент, характеризующийся расположением каждой фиксированной фигуры на своем поле фиксации и каждой привязанной — на заданном расстоянии (по траектории) от поля привязки.

**Цель игры**

точная в шахматах — для каждой из сторон дать мат неприятельскому королю;

неточная в модели — выигрыш материала (стоимости фигур в модели:  $p-1$ ,  $Q-3$ ,  $S-3$ ,  $L-5$ ,  $\Phi-9$ ,  $Kp-200$ ).

**Цепной список** — список, в котором отдельные члены списка располагаются произвольно в машинной памяти и связываются между собой так называемыми адресами связи; адрес связи помещается совместно с данным членом списка и указывает положение следующего члена списка.

**Шаблон**

пешек — массив из восьми (в общем случае) элементов; характеристика взаимного расположения пешек в позиции;

позиции — массив из двенадцати элементов; материальная характеристика позиции.

## СОДЕРЖАНИЕ

Предисловие . . . . .	3
1. ОБЩИЕ ПОЛОЖЕНИЯ . . . . .	4
Определение неточной задачи . . . . .	4
Неточные задачи и системы управления . . . . .	5
Два метода решения неточных задач . . . . .	5
Цель игры и оценочная функция . . . . .	9
Цель и прогноз (оптимальный вариант) . . . . .	11
Многоступенчатые системы управления . . . . .	12
Типы многоступенчатых систем . . . . .	13
Преимущества общей цели . . . . .	15
О методе сближения оптимальных вариантов ступеней при управлении типов С и Е . . . . .	16
Программы ЭВМ и человека . . . . .	17
Проблема распространения искусственного интеллекта . . . . .	18
2. МЕТОДЫ ОГРАНИЧЕНИЯ ДЕРЕВА ПЕРЕБОРА . . . . .	19
Усечение дерева перебора . . . . .	19
Цель неточной игры . . . . .	20
Оценочная функция . . . . .	20
Обрыв варианта . . . . .	21
Отсечение ветвей . . . . .	21
Горизонт . . . . .	21
Два дерева. Математическое отображение (МО) . . . . .	22
Многоступенчатость системы . . . . .	23
Три общих принципа ограничения . . . . .	24
Улучшение результатов перебора . . . . .	25
3. ПОИСК РЕШЕНИЯ И ОПЫТ ПРОШЛОГО . . . . .	26
Оригинальная ситуация . . . . .	26
Поиск решения по ассоциации . . . . .	27
Поиск решения по справочному методу . . . . .	27
4. ПРИМЕР РЕШЕНИЯ НЕТОЧНОЙ ЗАДАЧИ (ИГРА В ШАХМАТЫ) . . . . .	28
А. Поиск хода в оригинальной позиции . . . . .	29
Усеченное дерево перебора ходов . . . . .	29
Цель неточной игры в шахматах . . . . .	30
Оценочная функция. Две составляющие . . . . .	31
Обрыв варианта и прекращение игры . . . . .	32
Горизонт . . . . .	33
Шахматы — трехступенчатая система . . . . .	33
Первая ступень — фигура с траекторией . . . . .	34
Вторая ступень — зона игры . . . . .	36
Третья ступень — совокупность зон (МО) . . . . .	37

Дерево перебора и минимаксная процедура . . . . .	37
Техника определения траекторий . . . . .	38
Техника формирования зоны . . . . .	39
Позиционная составляющая оценочной функции . . . . .	41
Приоритет включения в перебор включенных в игру фигур и зон . . . . .	43
Включение зон в игру . . . . .	45
О повторении хода в узле . . . . .	47
Глубина перебора и включение зон в перебор . . . . .	48
Отсечение ветвей . . . . .	50
Три состояния зоны (действующее МО) . . . . .	52
Запоминание МО в конечном узле . . . . .	53
Технический вопрос . . . . .	53
Б. Использование опыта прошлого . . . . .	54
Библиотека дебютов . . . . .	55
Библиотека миттельшпиля . . . . .	55
Библиотека эндшпиля . . . . .	55
5. ТРИ ЭТЮДА (ЭКСПЕРИМЕНТ) . . . . .	58
6. ВТОРОЙ ЧЕМПИОНАТ МИРА . . . . .	64
Заклучение . . . . .	69
Приложение 1. Б. М. Штильман. Зоны игры . . . . .	70
Приложение 2. М. А. Цфасман, Б. М. Штильман. Позиционная оценка и приоритеты . . . . .	104
Приложение 3. А. Д. Юдин. Библиотека эндшпиля программы «Пионер» (использование опыта прошлого по справочному методу и методу стремления) . . . . .	110
Приложение 4. А. И. Резницкий, А. Д. Юдин. Ассоциативная библиотека фрагментов . . . . .	133
Список литературы . . . . .	141
Словарь терминов . . . . .	142

ИБ № 538

МИХАИЛ МОИСЕЕВИЧ БОТВИННИК

**О решении неточных переборных задач**

Редактор **Н. Г. Давыдова**

Художественный редактор **Н. С. Шенин**

Технический редактор **В. А. Позднякова**

Корректор **Н. М. Давыдова**

Сдано в набор 13.11.78. Подписано в печать 02.01.79 Т-03102  
Формат 60×84/16 Бумага типографская № 1 Гарнитура литерат.  
Печать высокая. Объем 8,83 усл. печ. л. 10,06 уч.-изд. л.  
Тираж 15 000 экз. Зак. 901 Цена 55 к.

Издательство «Советское радио», Москва, Главпочтамт, а/я 693

Московская типография № 10 «Союзполиграфпрома»

Государственного Комитета СССР

по делам издательств, полиграфии и книжной торговли.

Москва, М-114, Шлюзовая наб., 10



**Ботвинник М. М.**

**Б 86    О** решении неточных переборных задач.— М.: Сов. радио, 1979.— 152 с., ил.— (Кибернетика).

55 к.

В книге изложена общая теория решения неточных задач, аналогичных шахматной игре, приводится алгоритм игры шахматного мастера и частично дано описание программы «Пионер» с результатами проверки этой программы на шахматных этюдах.

Предполагается, что методы программы «Пионер» найдут применение в различных практических областях, и книга с интересом будет встречена специалистами, связанными с задачами управления.

**Б**  $\frac{30501-013}{046(01)-79}$  БЗ-97-8-78. 1502000000

**ББК 32.81**

**6Ф0.1**



55 к.



В серии «КИБЕРНЕТИКА»  
выйдут в свет следующие книги:

- |                  |  |
|------------------|--|
| Альтшуллер Г. С. | <b>Творчество как точная наука</b>                                 |
| Манин Ю. И.      | <b>Доказуемое и недоказуемое</b>                                   |
| Яглом И. М.      | <b>Математические структуры<br/>и математическое моделирование</b> |
| Яглом И. М.      | <b>Булева структура и ее модели</b>                                |

Предварительный заказ на указанные книги Вы можете  
оформить в магазинах, распространяющих литературу по данной  
тематике.

