

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 α_k -поле этой траектории. Пусть при этом необходимо уничтожить неприятельскую фигуру, стоящую на этом поле. Подсистема второй ступени 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 α_0 и α_k шахматной доски стоят фигуры разного цвета (+) и (-), называемые α_0 -фигурой и α_k -фигурой. Утверждение, что α_0 -фигура напала на α_k -фигуру, означает, что существует траектория α_0 -фигуры, ведущая с поля α_0 на поле α_k , причем такая, что время (в полуходах), затрачиваемое на передвижение α_0 -фигуры по этой траектории, не превышает некоторого числа N_L , называемого предельным горизонтом нападения. Рассмотренная траектория фигуры стороны (+) называется комлевой траекторией зоны, а α_0 -фигура — комлевой фигурой.

Рассмотрим совокупность полей комлевой траектории: это будут поля остановки (α -поля) и поля, проходимые без остановки (β -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 комлевой 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

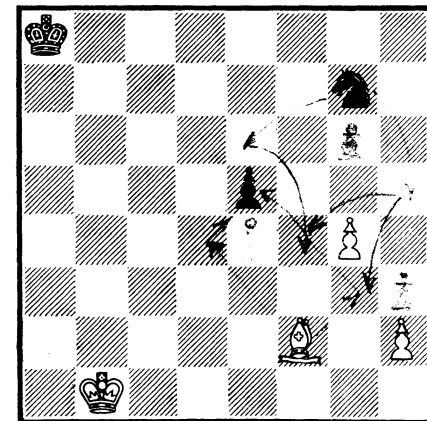


Рис. 19. Искусственная позиция

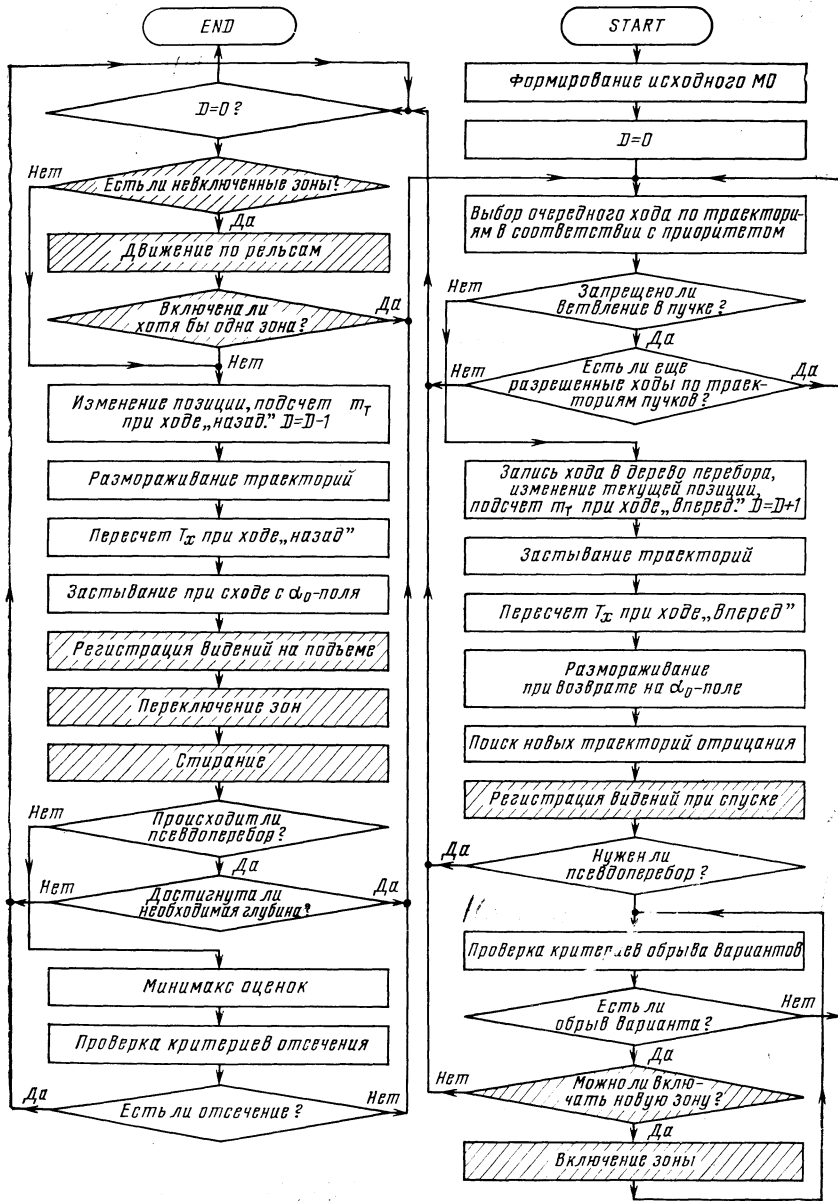


Рис. 20. Блок-схема программы поиска хода в оригинальной позиции

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.Cf2—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 наиболее выгодные 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 слонем). Программа выясняет: поднимаемая оценка 3 настолько плоха для черных, что предыдущий свой ход (1...Kg7—f5) они все равно отбросят по минимаксу. Поэтому исследовать другие ходы белых в данной позиции не имеет смысла, и

```

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

*****
BF2-G3
BF2-G3 500
***** 500
NG7-F5
PG4*F5
PG4*F5 3
NG7-F5 3
NG7-H5
PG4*H5
PG4*H5 3
NG7-H5 3
*****
BF2-G3
*****
BG3*E5
BG3*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
BG3*F4
PE5*F4
PE5*F4 0
BG3*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, P — 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.

Хотя, 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. И так, в исходной позиции все ходы черных, противодействующие нападению слона на пешку, получили оценка 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, проходимым 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

лишь 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.

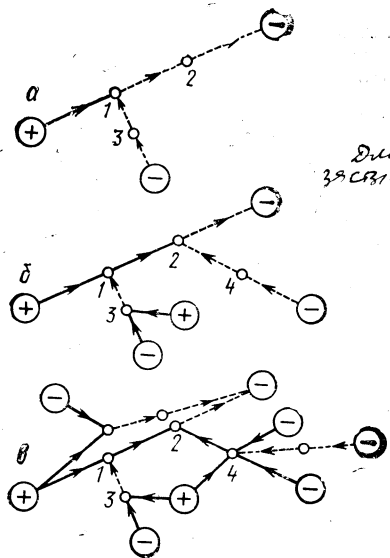
1.4. Формирование зоны. Построив в исходной позиции лишь комлевую траекторию, будем формировать зону в процессе перебора ходов и в дальнейшем осуществлять сам перебор, передвигая фигуры по построенным к данному моменту траекториям зоны. Принято, что траектория произвольной фигуры от поля a_0 до поля 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.



1	Номер фигуры
2	α_k -поле пучка
3	Длина кратчайшей траектории пучка
4	Блокада или контроль
5	Номер атакованной фигуры
6	Глубина застывания
7	α_i
8	α_{i+1}
9	α_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. 22. Некоторые этапы формирования зоны игры

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

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 α_0 -полю. По первым четырем параметрам из перечисленного списка пучок однозначно восстанавливается с помощью подпрограммы получения пучка траекторий [3].

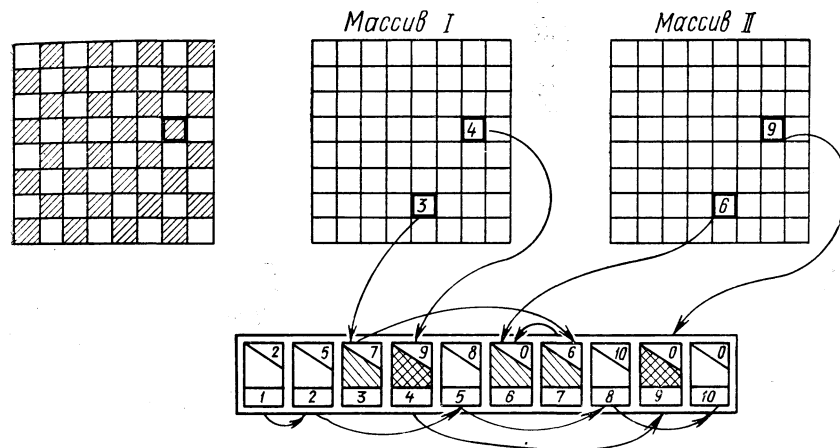


Рис. 24. Размещение множества пучков в оперативной памяти ЭВМ

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, α_k -поле пучка, длину кратчайшей траектории пучка, тип пучка (блокады, контроля), пара-

Сведения о пучках можно было бы хранить в виде многомерного 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 8×8 , нумерация элементов в которых соответствует нумерации полей на шахматной доске.

Информация обо всех пучках, проходящих через α_0 -поле, привязывается к α_0 -полям этих массивов 8×8 . Это делается следующим образом: на α_0 -поле массива I записывается адрес ячейки, со-

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

держашей информацию о первом пучке, проходящем через α -поле, затем все ячейки с информацией о пучках, относящихся к α -полю, связываются в цепочку, первым звеном которой является упомянутая выше ячейка. Адрес последнего звена цепочки записывается на α -поле массива Π . 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 α -поле, то информация обо всех пучках этой фигуры, проходящих через α -поле, будет привязана к данному полю. Если при продолжении перебора фигура снова появится на α -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 α -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 α -полями этой траектории, т. е. с полями шахматной доски (массивом 8×8), на которых в процессе перебора успевает побывать эта фигура (см. п. 1.5). Таким образом, с каждым полем массива 8×8 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 8×8 , 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 β -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 α_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, отводимая 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 ω -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 α - или β -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 $X — Y$. Определим траектории всех фигур, стоящих на доске в позиции после хода $X — Y$, 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 8×8 . 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 $X — Y$. 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 β -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 имей 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 $X - Y$ по траектории некоторой зоны. Тогда можно определить глубину включения этой зоны в дереве 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 $X_1 - Y_1$ 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 $X - Y$. 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 β -поля атакующей фигуры, и тогда последняя уже прошла мимо этого поля, либо траекторией контроля α -поля, причем атакующая фигура уже стоит на этом α -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 дальше всего простояла на своем α_0 -поле в текущем варианте. Казалось бы, наивысший узел в дереве, где эта фигура еще стоит на α_0 -поле (прослеживая 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,

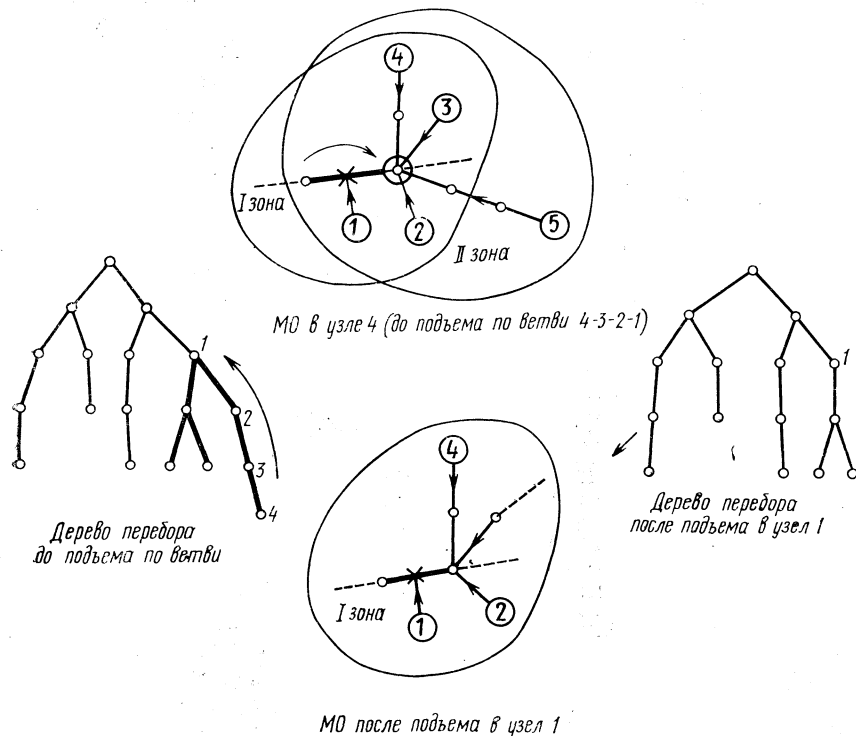


Рис. 25. Процедура псевдоперебора

сравнивая два числа: глубину подъема, на которой фигура еще стоит на α_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 α_i траектории отрицания зоны, причем кратчайший путь с поля α_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 $\alpha_i - \alpha_{i+1}$, 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 комлевого 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 α -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	2 ₁	3 ₂	4 ₁	5 ₃
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

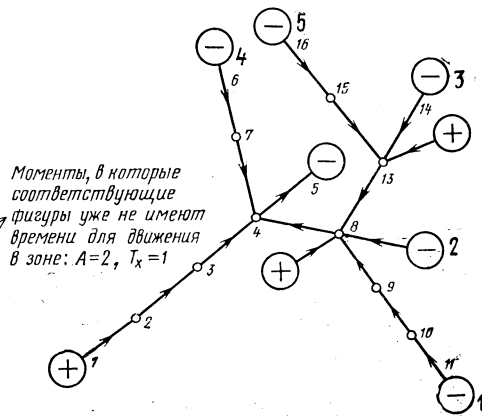


Рис. 26. Синхронный с перебором подсчет параметра T_x

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 участками, должны быть выключены из игры. Ту же операцию надо проделать с траекториями более высоких степеней отрицания, связанных с выключенными. Назовем эту операцию застыванием при потере связи с действующей траекторией.

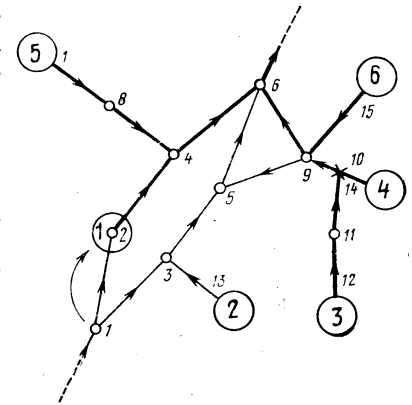


Рис. 27. Процедура застывания при потере связи с действующей траекторией

На рис. 27 иллюстрируется процедура застывания при ходе 1—2 фигуры 1. Застывают следующие участки и траектории: 1—2, 1—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 α -поле, принадлежащее нескольким пучкам траекторий данной фигуры. Назовем α -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 α -поля застывает лишь

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

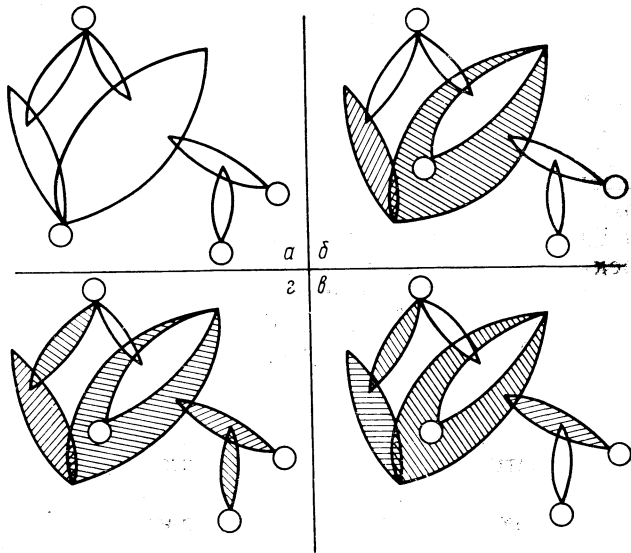


Рис. 28. Последовательность операций при застывании пучков траекторий

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

В момент застывания α -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 8×8 помечаются поля, принадлежащие траекториям, по которым сделан ход, причем рассматриваются только те α - и β -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 8×8 , 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 α -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 8×8 , 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 α -поле V данного пучка застывает, а просмотр массива 8×8 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 8×8 , 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 α_0 -поля на подъеме по дереву. Процедура застывания при сходе с α_0 -поля при подъеме по дереву соответствует процедуре размораживания при возврате на α_0 -поле, которая выполняется при спуске по дереву (рис. 29)

В процессе перебора при подъеме по дереву фигура может, вообще говоря, сойти с α_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 α_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 α_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 $Y-X$ просматривается весь массив 8×8 с целью поиска пучков, имеющих Y своим α_0 -полем. Это возможно, так как информация об α_0 -поле пучка содержится в соответствующей ячейке цепного списка (см. рис. 23). 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 α_0 -поле пучка, застывшего в смысле «99», то он и все связанные с ним пучки должны быть разморожены. Соответствующая процедура, аналогичная процедуре застывания при спуске с α_0 -поля, производит поиск пучков, застывших в смысле «99», имеющих данное поле своим α_0 -полем. Размораживаются эти пучки, а затем и пучки более высоких степеней отрицания.

Размораживание траекторий из-за прихода фигуры на α_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 α_0 -поле траектории, по значению параметра T_x , приписанного к этому полю, можно решить вопрос о включении траекторий в перебор.

1.18. Траектории 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-

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 спуске фигуры с α_0 -поля траектории при подъеме по дереву.

Организуя и воздействуя 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 α_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 α_k -фигуру. Для траектории контроля (отрицающей или комлевой траектории зоны контроля) цель — уничтожить контролируемую фигуру, поджидая ее в засаде на α_{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, воздействуя 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.

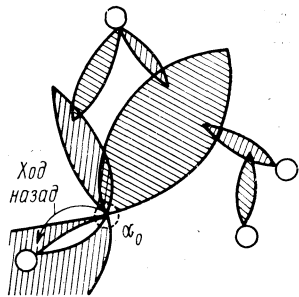


Рис. 29. Застывание пучков при спуске фигуры с α_0 -поля в процессе подъема по ветви дерева перебора (застывшие пучки траекторий заштрихованы)

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 минимуму 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. Критерий обрыва вариантов перебора. Целью игры в зоне является взятие α_k -фигуры, т. е. фигуры, которая атакована комлевой фигурой по комлевой траектории. Стоимость α_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 α_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) α_0 -фигура уничтожена или заблокирована так, что снятие блокады невозможно;

2) α_k -фигура уничтожена или отступила с α_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 α_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 α_0 - и α_k -фигур, причем они могут быть разного цвета. Чтобы сформулировать критерий обрыва варианта в МО, надо вычислить суммы стоимостей белых и черных α_k -фигур, входящих в совокупность зон: Σm_w и Σm_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 Σm_w и Σm_b из условия 4). Рассмотрим некоторую комлеву 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 α - и β -поля, обнаружим, что на α -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 — m_3+m_6 , для 1—2—3—5—6 — 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 8×8 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

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 8×8 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;

Рис. 30. Комлевые траектории совокупности зон

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 M более не формируются.

Это соображение лежит в основе метода «граней и оценок». Этот метод, называемый иногда α — β -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 $n=5$ полуходами. 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 оценок 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 α_k -фигур, т. е. Σm_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 минимакс 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 Σm_w и Σm_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 8×8 на соответствующий массив 15×15 [3] и определения полей в массиве 8×8 , 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 наивыгоднейшее 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 отсека 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 атакованной фигуры запоминается в соответствующих следах, привязанных к α -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 α -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 атакованной, а также о проходимости этой траектории. Говоря о проходимости, мы имеем в виду следующее. Рассмотрим все α -поля некоторой траектории, определим фигуру, находящуюся на расстоянии одного передвижения от данного α -поля. Для каждого α -поля определим результат оптимального размена на нем для этих фигур с учетом того, что фигура, двигаясь по траектории, пришла на это поле.

α -поля, на которых размен оказался в пользу стороны, которой принадлежит данная траектория, называются проходимыми. Зона с комлевой траекторией, у которой все α -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, строятся все ее α - и β -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-

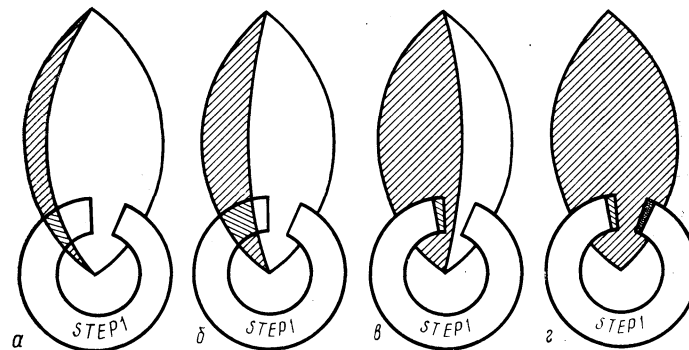


Рис. 31. Анализ траекторий пучка по признаку их длины

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 α -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 α -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 α_1 -поля и отмечаем α_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 α_1 -поле в массиве STEP1 не отмечается, если — меньше, происходит рассмотренная выше очистка массива STEP1

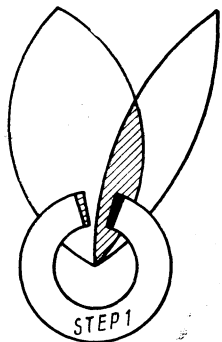


Рис. 32. Анализ траекторий пучка по признаку вилочности

и изменение текущего значения A . В результате за один просмотр пучка в массиве STEP1 окажутся отмеченными α_1 -поля всех кратчайших траекторий пучка, имеющих одинаковую длину (31, г). 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, α_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 данной фигуры имеет α_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 α_1 -поля основного пучка. Остается просмотреть все элементы массива STEP1 и найти наибольший. Его номер и будет координатой самого вилочного α_1 -поля кратчайших траекторий основного пучка. Следовательно, в переборе будет сделан ход α_0 — α_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 α_0 , а конечные поля — все поля доски, куда данная фигура может по пра-

вилам попасть за один ход с поля α_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 α_0 привязывается к этому полю. Во второй позиции следа вместо α_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, ближайшее к заблокированному
8	0	Поля, запрещенные для деблокады

Рис. 33. След пучка отступления (деблокады)

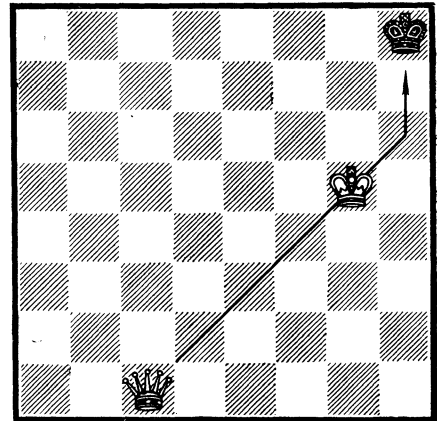


Рис. 34. Пример заблокированной траектории

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 заблокированному полю α -поля, блокированной траектории, т. е. α -поле, попадая на которое, фигура как бы упирается в блокаду.

Например, на рис. 34 траектория $\Phi c1$ — $h6$ — $h8$ 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 $Kpg5$ — $f4$ и $Kpg5$ — $h6$ в соответствующем пучке деблокады. Для этого в восьмой позиции следа записываются координаты по-

лей f4 и h6 — соседних с заблокированным $\alpha(\beta)$ -полем траекторий. 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 трактуется соответствующие лучи как заставленные своими фигурами, а ходы в этих направлениях — как запрещенные правилами игры (на рис. 34 это лучи g5—h6 и g5—c1). 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 заблокированная траектория анализируется для выяснения возможности ее деблокады. В этом анализе также участвует процедура хода не по МО. Если выясняется, что деблокада невозможна, заблокированная 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 заблокированным полем, произойдет по окончании анализа траекторий 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 α_k -поля пучка, либо сошла с данной траектории (уйдя в другую зону), либо же прошла 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 α_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 отсечения 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 двигаться по траекториям выключенной зоны, вилочным с какими-то траекториями включенных зон. В этом случае считается, что движение 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 ноль.

* 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 выполнении 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 изображена 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 $X-Y$. 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 $X-Y$). Записываем этот ход в специальный массив. В том же столбце находим адрес «брата» (в примере на рис. 35,а это узел 5). Переносим 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

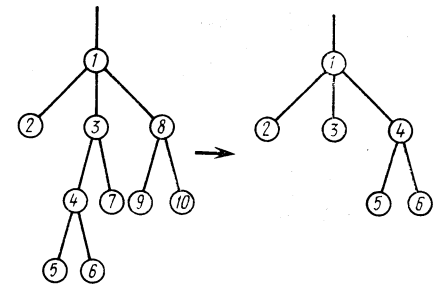
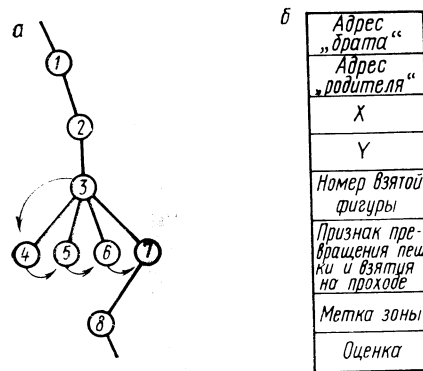


Рис. 35. Структура дерева перебора Рис. 36. Перестройка дерева перебора

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 отсекающими 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 α -полей незастывших траекторий включенных в игру зон, контролируемых белыми и черными соответственно. Далее уточняется, как эти α -поля выбираются.

Вычисление позиционной оценки связано, в основном, с решением двух задач. Прежде всего надо определить незастывшие траектории фигур и отметить их α -поля. Затем для каждой траектории и для каждого ее α -поля определить список фигур, находящихся на расстоянии одного передвижения от этого α -поля, и вычислить результат оптимального размена этих фигур на данном поле. Если данное α -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 α -полей траекторий производится до первого непроходимого α -поля.

Поскольку в программе «Пионер» информация о траекториях хранится в упакованном виде, точнее, в виде следов пучков траекторий (см. приложение 1), для определения α -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-